Analysis of Network Embedding Tools

Chanaka Bulathsinghalage
Computer Science Department
North dakota State University
Fargo, ND, 58105
Email: chanaka.cooray@ndsu.edu

Tanvir Ahmed
Computer Science Department
North dakota State University
Fargo, ND, 58105
Email: t.ahmed@ndsu.edu

***Abstract*—We examine the performance of 'DeepWalk'[1], 'Line'[2] and 'Node2vec'[3] three network embedding tools on a biological data set.[5][6] Traditional dimension reduction algorithms are not efficient for large scale network data. For this reason, we use network embedding tools. We applied two major classification algorithms for the classification to justify the results. We measure the Macro-F1 and Micro-F1 score after the classification to measure the performance. Finally, we compare the prediction accuracy of three network embedding tools. We use a protein protein biological data set for this analysis. We pre-processed the data sets before applying network embedding tools and classification algorithms for the classification. Our result provides a new dimension to illustrate the accuracy of those three networks embedding tools.**

1. INTRODUCTION

Various network embedding tools are being popular now-a-days. To be more specific 'DeepWalk' [1], 'Node2vec' [3], 'Line' [2], those three are the most popular and revolutionary network embedding tool as their main purpose is to learn low-dimensional vector representation for nodes in a network. In their work, those three tools showed that their algorithm produce excellent output and efficient for dimension reduction for network. Most of them have used social media dataset like- Twitter, YouTube, Flicker etc. Usually social dataset contains a large amount of data but they consider only portion of that data for their experiments. Another important point is that dataset itself is sparse. We try to apply their tools on biological dataset which is also sparse and significantly large dataset. The sparsity of a network representation has its own strength and weakness. Sparsity empowers the structure of productive discrete calculations, however, can make it harder to sum up in measurable learning. AI applications in systems, for example, arrange grouping, content suggestion, inconsistency discovery, and missing connection expectation must almost certainly manage this sparsity to endure. To overcome this problem in network field traditional dimension reduction algorithms like- PCA(Principal component analysis), MDS(Multidimensional Scaling) are not suitable. Because of that reason, network embedding tools are being popular more and more in the fields of data mining and machine learning.

Our contributions are as follows:

- We used a dataset which was not used before by those three networks embedding tools for this experiment. We make sure that dataset itself is sparse and very large.
- We preprocessed dataset based on their true and false values in the labels. Dataset contains more than twenty thousand labels. So, we extensively evaluate our comparison on multi label classification task on this biological protein-protein network by applying three network embedding tools And randomly select labels from features since we can not consider all of the features as the space is large.
- After applying two classification algorithms for the classification we calculate the micro-F1 and macro-F1 score for the accuracy observation of three network embedding tools.

The reminder of the report is organized as follows. In Section 2 we present state of the art and in section 3 we presents our methodology for data preprocessing and our implementation for the model. We illustrate our results and analyses in Section 4. We close with our conclusions in 5 and future works in 6.

2. RELATED WORK

Feature engineering has been broadly contemplated by the machine learning network under different headings. In networks, the conventional paradigm for generating node features is based on techniques for extracting features that typically involve some hand-crafted seed features based on network properties. On the other hand, our goal is to automate the entire process by casting extraction of features as a problem of representation learning, in which case we do not need any hand-crafted features.

2.1. Deepwalk

As a tool for analyzing graphs, they introduce deep learning to build robust representations suitable for statistical modeling. Within short random walks, DeepWalk learns structural regularities. On multiple social networks, they extensively evaluate their representations on multilabel classification tasks. In the presence of label sparsity, they show significantly increased classification performance, getting 5 percent-10 percent improvements on the sparse dataset problems they consider. In some cases, the representations of DeepWalk can outperform their competitors even if the training data is 60 percent lower. By building web-scale graph representations (such as YouTube) using a parallel implementation, they demonstrate the scalability of their algorithm. In addition, they describe the minimal changes needed to build their approach's streaming version.

2.2. Line

They are proposing a novel network embedding model called the LINE, which matches arbitrary types of information networks and easily scales into millions of nodes. It has a carefully designed objective function that preserves proximity to both the first and second order. To optimize the goal, they propose an edge-sampling algorithm. The algorithm addresses the limitation of decent classical stochastic gradient and improves inference efficiency and effectiveness. On real-world information networks, they conduct extensive experiments. Experimental results prove that the proposed LINE model is effective and efficient.

2.3. Node2vec

They are proposing node2vec, an efficient scalable algorithm for network feature learning that efficiently optimizes a new network-aware neighborhood that uses SGD to preserve objective. They show how node2vec complies with established network science principles, providing flexibility in discovering representations that conform to different equivalences. They extend node2vec and other feature learning methods from nodes to pairs of nodes for edge-based prediction tasks based on neighborhood-preserving goals. For multi-label classification, they empirically evaluate node2vec and link prediction to multiple real-world datasets.

3. METHODOLOGY

3.1. Overview

We take 'A Human protein-protein Interaction Dataset'. We get a feature dataset [5] which contains 18,159 nodes and 11,759,455 edges. We also get a labels dataset [6] which holds 21,979 labels. Our preprocessing step can be seperated into three sections.

- Preprocess Links: We first removed the nodes which has no connection with labels. Then build the network (adjacency matrix).
- Preprocess Labels: We built a label matrix from label data [6].
- Create three datasets in conditions:
  - Balanced dataset
  - Medium-Balanced dataset
  - Unbalanced dataset

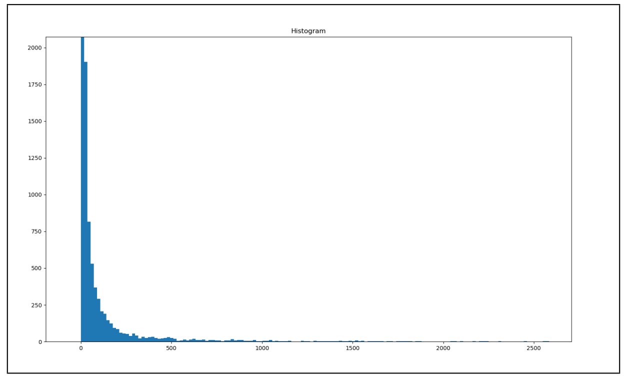

Figure 1. Histogram of the labels

We generate a histogram for better understanding and to visualize why we make three different datasets based on their conditions. We see in the Figure 1, X-Axis contains number of true labels and Y-Axis contains number of labels. In the 5%-10% range, we see most of the labels are there as their frequency is very large. However, they contain most of their label values are false and we cannot make any good prediction with this data. So, for this reason we called this portion of data is an unbalanced dataset. Also, we measure around 25%-30% portion in the histogram which contains true and false values almost similar and that's why we represent that portion of the dataset is a medium-balance dataset. Rest of the portion which is around 35%-65% are contains significant amount of true values, so we represent them as balance dataset.

After that we generate the outputs for different dimensions using the tools and applied the classification algorithms for the prediction. Then we built a model to calculate the macro-F1 and micro-F1 scores which helps to evaluate the performance of the results.

3.2. Classification Algorithms used in our methodology

3.2.1. Support Vector Machines. Support-vector machines (SVMs, also support-vector networks) are supervised learning models in machine learning with associated learning algorithms that analyze data used for classification and regression analysis. It is mostly used in classification issues, though. In this algorithm, each data item is plotted as a point in n-dimensional space (where n is the number of features we have) with the value of each feature being the value of a specific coordinate. Then, by finding the hyper-plane that differentiates the two classes very well.

3.2.2. Logistic Regression. Logistic regression is a classification algorithm used by a discrete set of classes to assign observations. Logistic regression transforms its output using the logistic sigmoid function to return a probability value that can then be mapped to two or more discrete classes, unlike linear regression that outputs continuous number values. There are three types of logistic regression: binary, multi and ordinal.

3.3. Implementation of the model to analyze performance

We used micro-F1 and macro-F1 scores to analyze the performance of the tools. Equations for those two as follows for a multi class classifications problem. In this equation sample, we only calculated the scores for a classification problem with two labels.

$$Micro-average-precision = \frac{TP1+TP2}{TP1+TP2+FP1+FP2}$$

$$Micro-average-recall = \frac{TP1+TP2}{TP1+TP2+FN1+FN2}$$

TP, FP, TN and FN are the values those are resided in the confusion matrix and stands for True positive, False positive, True negative and False negative. Micro-F1 value can be derived by getting the harmonic mean of those two values.

Macro-F1 values are generated by taking the harmonic mean of the average of the precision and recall of the results.[4]

Next, we explain how we generate the confusion matrix from the results we have so far.

We have two types of data sets. First one contains the features we extracted from the tools and second one contains the labels we are predicting and use for the classification. To explain this, let's take one feature set extracted from 'deepwalk' tool which has only 4 features as follows.

TABLE 1. SAMPLE FEATURE SET EXTRACTED FROM DEEPWALK TOOL FOR FOUR DIMENSIONS

Node Id	Feature 1	Feature 2	Feature 3	Feature 4
0	0.15	0.07	-0.678	0.177
1	0.0467	0.118	-0.607	0.255
2	0.0715	0.0339	-0.696	0.1223
:	:	:	:	:
:	:	:	:	:

So we have four features(We can think those features as attributes of each node) for each node in the network. In our dataset, we have features for all 18159 nodes like above.

Then we have the predicting labels as follows. Labels are the significant attributes associated with the nodes those are already identified previously from various researches.

So we have labels for each node ID and in our data set we have 30 labels for each of the unbalanced, balanced and medium-balanced data sets.

Then we implement our model to include both features and labels as the inputs and then we shuffle the rows n times and for each shuffle we consider n% of data as the training set and (100-n)% data for testing set. We predict the labels for the testing data using two classification algorithms those are support vector machines and logistic regression. We have used two different classifications algorithm to make the results to be consistent. Then using the predicting values, we built the confusion matrix and from that we calculate the micro-F1 and macro-F1 values according to the equations we defined earlier.

TABLE 2. LABELS FOR UNBALANCED SET

Node Id	Label 1	Label 2	Label 3	...
0	0	1	0	...
1	1	0	1	...
2	1	1	0	...
:	:	:	:	...
:	:	:	:	...

4. RESULTS AND ANALYSIS

We list down all the results separately for the two classifications algorithm we used and also by the macro-F1 and micro-F1 values. Macro-average method can be used when we want to know how the system performs overall across the sets of data and micro-average method used when we analyze the results more deeply.

From each tool, we extracted 10 sets of features(feature sets consist of number of features varies as 2,4,16,32,64,128,256,512 and 1024). And each feature set used with three categories of labels(Unbalance, balance and medium-balanced) to analyze the performance. So for 3 tools, we have 30 (3x10) output feature sets in total and when comparing each feature set with three categories, we have 90 (30x3) result sets. Since we are using two classification algorithms, we evaluated each feature and each label two times. So we have 180 (90x2) results in total. Due to the higher time complexity of the classification algorithm and the limited time frame, some of the results could not be generated for the higher number of feature sets(such as 256,512,1024).

We present the tables of the results as well as the graph figures for better analysis.

TABLE 3. MACRO-F1 VALUES FOR UNBALANCE DATASET USING SVM

log2(N)[1]	Deepwalk	LINE	Node2vec
1	5.201234448	4.745412731	5.066096716
2	7.851855167	8.215361288	7.524849911
3	10.47666459	11.97088324	9.527082619
4	12.29433382	13.53741295	9.966670795
5	13.40054011	14.26337705	9.899701113
6	13.96101052	14.17278914	9.784935415
7	13.70854096	14.28905118	9.480504903
8	14.51227989	14.30794799	8.463898648

1. *N* represents number of features extracted from each tool. Here we represent the logarithmic values in base 2 in the table. So the actual values will be 2,4,8...,2^n.

TABLE 4. MICRO-F1 VALUES FOR UNBALANCE DATASET FOR SVM

log2(N)[1]	Deepwalk	LINE	Node2vec
1	6.212770522	5.907342479	6.232173554
2	8.674915379	9.140684442	8.309886387
3	10.99848259	12.44466251	10.14835619
4	12.71628705	13.777782	10.50829924
5	13.75698543	14.48300007	10.45378594
6	14.29485512	14.39347755	10.41809428
7	13.99081087	14.520903	10.17470183
8	14.80535832	14.54539905	9.220703175

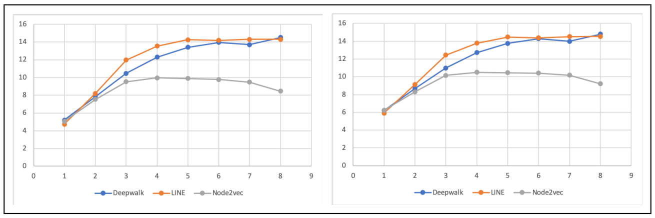

Figure 2. Macro-F1 and Micro-F1 scores for unbalanced data using SVM[2]

TABLE 5. MACRO-F1 VALUES FOR MEDIUM-BALANCE DATASET FOR SVM

log2(N)[1]	Deepwalk	LINE	Node2vec
1	48.85701482	44.66509374	46.14794502
2	56.80100632	57.40884395	54.62623917
3	61.13330124	63.81585018	58.37778746
4	63.33834476	65.59813927	58.19865396
5	64.84438274	66.51345008	58.0294962
6	65.18469152	66.14380992	58.82566211
7	65.73599673	65.2904951	58.80073256

TABLE 6. MICRO-F1 VALUES FOR MEDIUM-BALANCE DATASET FOR SVM

log2(N)[1]	Deepwalk	LINE	Node2vec
1	49.27109005	45.3000912	46.12511827
2	56.62198881	57.09498356	54.44562184
3	60.98591733	63.5805026	58.26084614
4	63.18074654	65.44071474	58.10496095
5	64.67889501	66.35487405	57.94873673
6	65.03769133	66.00293364	58.73490804
7	65.56975068	65.14395905	58.67011597

TABLE 7. MACRO-F1 VALUES FOR BALANCE DATASET FOR SVM

log2(N)[1]	Deepwalk	LINE	Node2vec
1	70.22127135	70.02298825	69.12149786
2	76.01688592	76.40673692	74.29947504
3	78.27083852	79.8453322	76.14203695
4	79.66497463	80.81972808	76.00271898
5	80.55033986	81.04394582	75.82617763
6	80.69904375	80.84418155	75.77684405

2. Left graph shows the Macro-F1 values and right graph shows the micro-F1 values. X axis denotes the number of features extracted from the tools where y axis denotes the scores.

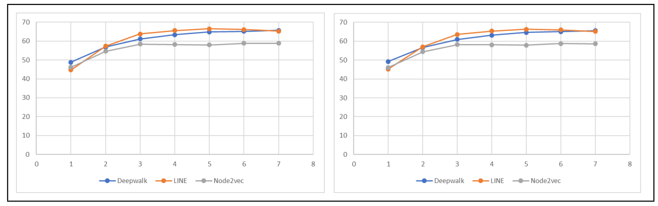

Figure 3. Macro-F1 and Micro-F1 scores for medium-balanced data using SVM[2]

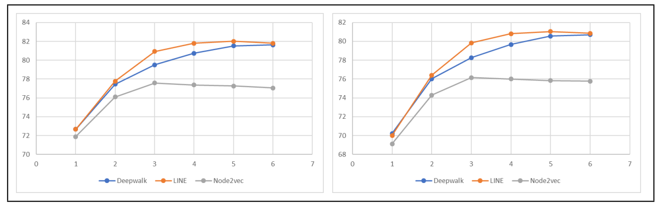

Figure 4. Macro-F1 and Micro-F1 scores for balanced data using SVM[2]

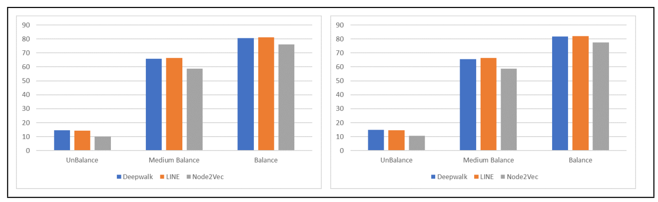

Figure 5. Macro-F1 and Micro-F1 scores for three categories of data using SVM

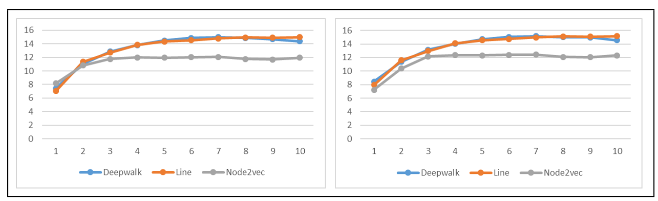

Figure 6. Macro-F1 and Micro-F1 scores for unbalanced data using Logistic Regression[2]

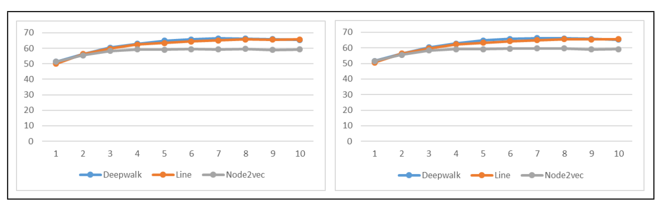

Figure 7. Macro-F1 and Micro-F1 scores for medium-balanced data using Logistic Regression[2]

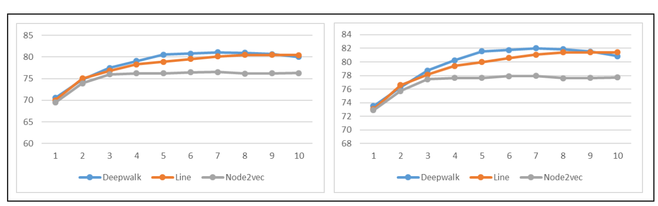

Figure 8. Macro-F1 and Micro-F1 scores for balanced data using Logistic Regression[2]

TABLE 8. Micro-F1 values for balance dataset for SVM

log2(N)[1]	Deepwalk	LINE	Node2vec
1	72.67806297	72.66978132	71.89555066
2	77.46236416	77.78064339	76.10702053
3	79.5058748	80.93093451	77.57526884
4	80.748149	81.78838514	77.36840545
5	81.51525121	82.00549618	77.25949123
6	81.6315073	81.80009606	77.05583058

TABLE 9. Macro-F1 values for the comparison between categories for SVM

Categories	Deepwalk	LINE	Node2vec
Unbalance	14.51227989	14.30794799	9.966670795
Medium Balance	65.73599673	66.51345008	58.82566211
Balance	80.69904375	81.04394582	76.14203695

TABLE 10. Micro-F1 values for the comparison between categories for SVM

Categories	Deepwalk	LINE	Node2vec
Unbalance	14.80535832	14.54539905	10.50829924
Medium Balance	65.56975068	66.35487405	58.73490804
Balance	81.6315073	82.00549618	77.57526884

TABLE 11. Macro-F1 values for unbalance dataset for Logistic Regreesion

log2(N)[1]	Deepwalk	LINE	Node2vec
1	7.439415136	7.038283219	8.164622446
2	11.00536791	11.33379731	10.83022557
3	12.85586423	12.73059734	11.78187833
4	13.80497166	13.86109423	12.00745114
5	14.50369594	14.3487228	11.96546346
6	14.85923858	14.53967014	12.04892227
7	14.97617304	14.7887705	12.06508669
8	14.8687155	14.94973426	11.76922222
9	14.68331543	14.89545151	11.7112323
10	14.39094068	14.98757652	11.95964921

TABLE 12. Micro-F1 values for unbalance dataset for Logistic Regreesion

log2(N)[1]	Deepwalk	LINE	Node2vec
1	8.452530978	7.957178839	7.222837531
2	11.35503212	11.58914563	10.37181088
3	13.14900185	12.97473589	12.17753979
4	14.03898167	14.09214551	12.34337816
5	14.710545	14.54399434	12.33163242
6	15.04305216	14.7242239	12.40176407
7	15.14596467	14.96132481	12.41898214
8	15.02200574	15.11994543	12.10907674
9	14.98437627	15.07049803	12.05334411
10	14.53509714	15.16756421	12.32256064

TABLE 13. Macro-F1 values for medium-balance dataset for Logistic Regreesion

log2(N)[1]	Deepwalk	LINE	Node2vec
1	51.26368878	50.04423522	51.27621003
2	56.12625519	56.26776347	55.4561548
3	60.45748395	59.71667954	58.28278989
4	62.98699851	62.46610015	59.20892441
5	64.86624012	63.37584057	59.1571695
6	65.82153938	64.35424828	59.49801604
7	66.40637763	65.12825603	59.29363411
8	66.30676396	65.68106001	59.65540856
9	65.85517191	65.63441315	59.00239659
10	65.44664619	65.6747875	59.26775546

TABLE 14. Micro-F1 values for medium-balance dataset for Logistic Regreesion

log2(N)[1]	Deepwalk	LINE	Node2vec
1	51.65705011	50.53457397	51.67431112
2	56.26105465	56.36491844	55.59479107
3	60.30886397	59.70628053	58.30390287
4	62.85143478	62.3694847	59.17338887
5	64.74327875	63.24861997	59.15211523
6	65.69596419	64.23942947	59.44580797
7	66.26559338	65.00964194	59.63158569
8	66.16883676	65.56852347	59.60817647
9	65.70708903	65.50769032	58.94383657
10	65.30311377	65.56441236	59.19901067

TABLE 15. Macro-F1 values for balance dataset for Logistic Regreesion

log2(N)[1]	Deepwalk	LINE	Node2vec
1	70.54100701	69.97970474	69.49230269
2	74.73930612	75.06800722	73.90539745
3	77.4495772	76.82835671	75.96419707
4	79.08844608	78.30069359	76.25489086
5	80.52272979	78.8832317	76.21653526
6	80.79641327	79.52952374	76.49222419
7	81.05509343	80.12147565	76.50778414
8	80.97039905	80.48239974	76.18382274
9	80.66758858	80.4742072	76.21467549
10	80.01583956	80.45952241	76.30022167

TABLE 16. Micro-F1 values for balance dataset for Logistic Regreesion

log2(N)[1]	Deepwalk	LINE	Node2vec
1	73.49775107	73.0377183	72.88313456
2	76.30305644	76.58976272	75.73286006
3	78.72601548	78.13989635	77.46705633
4	80.22633357	79.4169942	77.64059646
5	81.54803019	79.97145824	77.64983962
6	81.76325452	80.56757203	77.90260413
7	82.00550982	81.06935189	77.938965
8	81.85322306	81.4105985	77.60952737
9	81.51925843	81.41268092	77.64063677
10	80.85496527	81.40590836	77.71612502

As the analysis, we are providing a general analysis here according to the results we derived from the classification algorithms, support vector machines and logistic regression.

For the unbalanced data set, we can clearly observe that node2vec is not performing well where deepwalk and LINE performs better according to Figure 2 and Figure 6. For the lower dimensions, LINE tool is performing better, but when we consider higher dimensions, deepwalk starts to perform

TABLE 17. MACRO-F1 VALUES FOR THE COMPARISON BETWEEN CATEGORIES FOR LOGISTIC REGRESSION

Categories	Deepwalk	LINE	Node2vec
Unbalance	14.97617304	14.98757652	12.06508669
Medium Balance	66.40637763	65.68106001	59.65540856
Balance	81.05509343	80.48239974	76.50778414

TABLE 18. MICRO-F1 VALUES FOR THE COMPARISON BETWEEN CATEGORIES FOR LOGISTIC REGRESSION

Categories	Deepwalk	LINE	Node2vec
Unbalance	15.14596467	15.16756421	12.41898214
Medium Balance	66.26559338	65.56852347	59.63158569
Balance	82.00550982	81.41268092	77.938965

better and better.

When considering the medium balanced dataset, node2vec also has poor performance, but the LINE and Deepwalk tools are performing almost equally as in the Figure 3 and Figure 7.

Considering balanced dataset, node2vec still has very poor performance, and deepwalk and LINE perform almost equally. Also we can notice one thing from Figure 4 that, when we extracting less number of features LINE performing good and when we increase the size of the feature set, deepwalk starts to perform better.

Also here we represent a comparison of three categories as well in Figure 5 and Figure 9. In there we get the highest performed one in each category for each tool and represent in the graph. In there we can observe that, in unbalanced dataset every tool not performing good but when we keep increasing the balance of the data, performance of the tools increases as well. Also node2vec does not perform well in all three types of categories, but deepwalk and LINE performs almost equally in all three categories.

5. DISCUSSION AND CONCLUSION

We try to measure the performance of the most renowned network embedding tools. We applied the tools for a large biological data which itself is a very sparse dataset. We measure the F1 scores after applying two different classification algorithms and find out their prediction accuracy. From the results we observed several noticeable things. Node2vec does not perform well in this dataset comparing with deepwalk and LINE. Deepwalk performs better when we extract large number of features and LINE performs better when we extract less number of features.

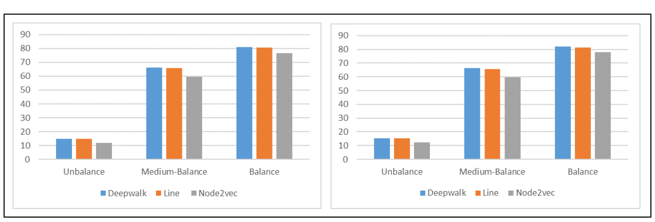

Figure 9. Macro-F1 and Micro-F1 scores for three categories of data using Logistic Regression

Node2vec tool's performance in this dataset is quite interesting since in their original work, they have proven that node2vec is perfoming better than all other tools[3]. But those experiments are conducted for small or medium-scaled networks and the dataset we use in this experiment is a large network. So the question remains that does the node2vec only performs better in small or medium scaled network and does not perform well in large networks. To clearly justify that we need to do multiple experiments using large networks.

Also we have found out that, in unbalance dataset the prediction accuracy of all the tools are very poor but when we keep increasing the balance of the prediction data, the tools start to perform better. It can be justified that since when we have a dataset which has prediction data for only one side, the prediction accuracy becomes low as we do not have significant training data for balance prediction. We expected that pattern to be there and the results have proven that the pattern exists.

In addition, we noticed that logistic regression has a better accuracy as a classification algorithm than the support vector machine algorithm.

6. FUTURE WORK

The confusion still remains regarding the performance of the node2vec tool. So as the future work, we have to test on different types of datasets(Small, medium-scaled and large networks) and represent the work as we have done for this experiment.

ACKNOWLEDGMENTS

We are thankful to Dr. Lu Liu who gave us the idea for this project, for giving us the background knowledge and providing us the dataset as well.

References

[1] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk," Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 14, 2014.

[2] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line," Proceedings of the 24th International Conference on World Wide Web - WWW 15, 2015.

[3] A. Grover and J. Leskovec, "node2vec," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16, 2016.

[4] Shams, R., Shams, R. and profile, V. (2011). Micro- and Macro-average of Precision, Recall and F-Score. [online] Rushdishams.blogspot.com. Available at: http://rushdishams.blogspot.com/2011/08/micro-and-macro-average-of-precision.html [Accessed 5 May 2019].

[5] String-db.org. (2019). Downloads/STRING: functional protein association networks. [online] Available at: https://string-db.org/cgi/download.pl?species_text=Homo+sapiens [Accessed 7 May 2019].

[6] String-db.org. (2019). Index of /mapping_files/geneontology. [online] Available at: https://string-db.org/mapping_files/geneontology/ [Accessed 7 May 2019].