Influence Maximization in Complex Networks Through Supervised Machine Learning

Owais A. Hussain[1](✉) and Faraz Zaidi[2]

[1] Karachi Institute of Economics and Technology, Karachi, Pakistan
[2] Institute of Business Administration, Karachi, Pakistan
fzaidi@iba.edu.pk

Abstract. Identifying influential nodes in complex networks is a well studied problem in network science. Finding an optimal set of influential nodes is an NP-Hard problem and thus requires the use of heuristics to find the minimal set of nodes capable of maximizing influence in a network. Once identified, these influencer nodes can been applied in various applications such as controlling disease outbreaks, identifying infectious nodes in computer networks, and finding super spreaders for viral marketing in social networks. This paper proposes a novel approach to solve this problem by modeling it as a supervised machine learning problem. Several synthetic and real world networks with nodal and network level attributes are used to train supervised learning models. Model performance is tested against real world networks emanating from a variety of different domains. Results show that the trained models are highly accurate in identifying influential nodes in networks previously not used for training and outperform commonly used techniques in the literature.

Keywords: Influence maximization · Complex networks · Machine learning · Gradient boosting · Resilience

1 Introduction

Influence is defined as the tendency of an individual to perform an action, and trigger another individual to perform the same action. Influence maximization is the problem of identifying optimal subset of such individuals capable of maximizing influence in a social network [27]. This field has wide applications in social and other networks. Real world applications of influence maximization can be found in controlling diseases outbreaks spreading information as word-of-mouth and viral marketing processes [14] and identifying critical nodes in infrastructures such as transportation networks and power systems.

Researchers have used various approaches to solve this NP-hard problem [22]. Kleinberg and colleagues [22] suggested a greedy strategy based on submodular functions that can obtain a solution that is provably 63% within the optimal one

R. M. Benito et al. (Eds.): COMPLEX NETWORKS 2021, SCI 1073, pp. 217–228, 2022.
https://doi.org/10.1007/978-3-030-93413-2_19

but does not guarantee the best solution [27]. Furthermore, greedy approaches are not scalable to moderate size networks with hundreds of thousands of nodes.

Another approach to identify influential nodes is to use heuristic methods. Widely used methods include the high-degree centrality (HD), high-degree adaptive (HDA) [18], PageRank [28], k-cores [2], betweenness centrality, closeness and eigenvector centrality [23], equal-graph-partitioning [10] and collective influence [27]. These methods have often performed indifferently on networks with varying structural properties [19,20] and there is no single heuristic that is capable of identifying the smallest set of influential nodes.

This paper models the problem as a supervised machine learning problem where influential nodes are a function of nodal and network level attributes combined together. Using several networks, varying in size, domain and other structural properties, a set of influential nodes is identified and then machine learning models are trained to classify influential nodes in test networks. To the best of our knowledge, this is the first attempt to use supervised machine learning algorithms to identify influential nodes in complex networks.

2 Related Work

2.1 Influence Mining

Influence mining is a diverse field and many researchers have studied it in a variety of domains to: develop models for information diffusion; enhance these models; learn influence probabilities; maximize influence; observe influence propagation; apply direct data mining approaches; explore applications in real life.

[14] was the first to study the problem of identifying influential users and influence propagation in terms of data mining. They modeled a social network as Markov Random field and proposed heuristics for selection of most influential users. Extending Domingos's work, Kempe et al. [22] took greedy approximation approach to guarantee influence maximization. They focused on two basic and very frequently used time step-based diffusion models: Linear Threshold model (LT) and Independent Cascade model (IC).

Since influence maximization under both LT and IC models are NP-Complete [22], Leskovek et al. targeted the propagation problem using a different approach, calling it Outbreak Detection [24]. The outbreak detection problem is to detect the minimal set of nodes that could spread a virus in a network in a minimal time. They developed an efficient algorithm using an approach called lazy-forward optimization to select new seeds; the algorithm is 700 times more efficient than simple Greedy solution. Wei Chen *et al.* [9] extended IC model to incorporate propagation of negative opinions by introducing a parameter, quality factor to depict natural behaviour of people and, while maintaining Greedy method's sub-modularity property, designed a non-trivial algorithm for k seed selection that computes influence in tree-structures and further build a heuristic for influence maximization. Kazumi Saito et al. [29] discussed how to learn influence probabilities for IC model from historical influence propagation traces. They propose a method, Expectation Maximization (EM) to solve the maximization problem by using likelihood maximization. The EM method is a heuristic, which assumes that most influential users are the ones who propagate their actions to most of their neighbours. Their work has limitations when applying to real-World problems, most significant of which is that according to their model, nodes with higher number of connections can influence more nodes and result in being influential nodes rather than being able to generate a cascading behaviour.

Goyal et al. [16] studied the same problem under an alternate model, the General Threshold Model. They used this model and made the probabilities decline with time. The assumption behind this is that if a user u performs an action, then his linked user v will either repeat the action shortly after, or never. This decline of probability is observed to be exponential. They compared two genres of models; one assumes the influence probability remains static with time; 3 models, Bernoulli distribution, Jaccard index and Partial Credits and their combinations were discussed. The contrary assumes the influence probability to be a continuous function of time. The results showed that time-aware model was better, not only in terms of accuracy but being able to roughly predict future action times.

Lu et al. [25] comprehensively summarized the area of influence mining on different types of networks, discussing centrality methods to state-of-art heuristics to Greedy algorithm and its variations, the authors conclude on the applications of influence mining in social networks, financial area, scientific influence.

2.2 Machine Learning

Surprising as it may seem, there is scarcity of studies which utilize the power of continuously evolving machine learning techniques in order to identify influential nodes in a network in a generic fashion. Among those that exist, the prominent publications are often specific to social networks.

[17] demonstrate how predictive analytics can be used to predict diffusion cascades in a social network. Their proposed model, T-BaSIC does not assume a fixed diffusion probability, but a time-dependent function. This temporal diffusion model is used to create time series to describe how a topic originated and spread in a Twitter network in a closed set of users. The authors trained multiple classifiers to learn parameters of diffusion function, and opted Bayesian classifier to define the diffusion function. The results show that in the information diffusion process, initially the volume of tweets is high against a topic, and then lowers with time following a wave pattern. The T-BaSIC method reduces the overall diffusion prediction error by 32.75% in comparison with 1-time lag model.

Zanin *et al.* [33] discusses the application of data mining techniques in complex networks. The authors describe how data mining techniques can make use of structural patterns of individual nodes as well as whole network. Excerpts from this article are widely reflected in experiments we propose in this manuscript.

Albeit these attempts, to the best of our knowledge there is lack of any published work which solves the influence mining problem using supervised machine learning on a variety of complex networks.

3 Design and Methodology

The experiments discussed in this paper consist of openly available real graphs as well as synthetically generated graphs. Moreover, unlike majority of studies, which use network data, user attributes, action logs and other information, we only use network's structural properties that can be calculated by $G = (V, E)$, where V is a set of vertices or nodes in the graph G, while E is the set of edges or links connecting these vertices. Source code of all experiments conducted in the study are shared on GitHub[1].

3.1 Training Data Sets

We used networks from various domains with different sizes and structural properties. A total of 390 graphs were used to train and validate the model. The networks used in this study come from three sources:

Synthetic Networks are generated networks, which exhibit similar properties as real networks. We drew 6 random samples for each of the following types of complex networks using Igraph library ranging from size 10 - 2000 nodes [12]:

1. Scale free: almost all complex networks are scale free [3]. They are defined by a prominent property that their degree distribution obeys power-law[2]. The power law coefficient in a scale free network is usually $2 < \gamma < 3$.
2. Small-world: these networks have low APL and high CC [31].
3. Small-world and scale free: most of the real world networks, especially social networks exhibit both scale free and small-world properties [18]

Cited Networks are frequently cited networks from various domains and sizes that are available online:

1. Zachary karate: members of a university karate club by Wayne Zachary.
2. World trade: data about manufacturers of metal among 80 countries [30].
3. Nematode: Neural network of the nematode [32].
4. Political blog: hyperlinks between weblogs on US politics [1].
5. Yeast protein: protein-protein interaction network of yeast [21].

Extracted Networks additionally, we extracted two social networks to ensure that the training set not only contains clean and ready-to-use networks but also replicate real-world situations where networks are in raw state without any preprocessing. These networks are described as follows:

[1] https://github.com/seekme94/influence-mining/tree/influential-node-prediction/Experiments.

[2] States that a change in one quantity X results in a proportional relative change in another quantity Y.

1. Influence citations: network of paper citations on topic of influence mining till 2016. The data set was constructed by collecting a corpus of research articles on various topics of Influence mining. Each node in this network represents a unique article, while the edges represent citations between these articles.
2. Twitter: starting with a reference user account (ID: seekme_94) in the Twitter network, a subset network was extracted, such that each node is either a direct connection of reference account, or a connection at a distance of 1 degree. The reference node was finally removed from the network.

3.2 Test Data Sets

To test the final model, following complex networks from various domains were used. Again, variation in their nodal as well as structural properties can be observed in Table 1. All these are also frequently cited networks in the literature. These networks can be downloaded from KONECT: http://www.konect.cc.

– Author NetScience: coauthorship network of network theory researchers.
– ITA 2000: air transport network representing connections between airports through direct flights.
– AS-CAIDA: network of autonomous systems of the Internet connected with each other from the CAIDA project.
– JDK Dependencies: a network of class dependencies in JDK v1.6.

Table 1. Networks used for Training. Basic statistics for all the real networks used for training the model. APL is Average Path Length, γ is Power-law coefficient, CC is clustering coefficient and AvDg is Average Degree. The variation in size of the networks, APL, γ, CC and AvDg demonstrates how we have captured different network level structural properties for our training dataset.

Network	Nodes	Edges	APL	γ	CC	AvDg
Training data						
Karate club	34	78	2.4	2.12	0.256	4.6
World trade	80	1000	1.7	2.56	0.460	25
Nematode	297	2359	2.5	3.29	0.181	15.9
Political blog	1222	19089	2.7	3.89	0.173	4.6
Yeast protein	1458	3941	6.1	3.24	0.226	31.2
Influence citation	1254	1993	3.8	2.35	0.057	3.2
Twitter	894	2060	4.9	4.60	0.052	5.4
Test data						
Author	379	914	6.0	3.36	0.431	4.8
ITA 2000	3304	67195	4.0	1.69	0.249	40.7
AS-CAIDA	26475	53381	3.9	2.09	0.007	4.0
JDK	6434	150985	2.1	2.01	0.011	46.9

3.3 Influential Node Identification

The first step to train supervised machine learning models is to identify influential nodes in training dataset.

Influence Maximization Algorithms. In order to guarantee that the nodes we label as influential, in fact are influential, we used the optimal solution (NP-hard) for networks of size less than 50 nodes, and baseline Greedy algorithm for larger networks [22]. Opting for greedy algorithm was a compulsion because finding the optimal solution for larger graphs is practically impossible in polynomial time.

Influence Test. In order to quantify the influence of a node, multiple methods exist, including diffusion test under linear threshold model, diffusion test under independent cascade model [22] and resilience test [11] which quantifies the network breakdown after a set of nodes is removed. Since the resilience test does not require any data other than network information, therefore this study uses resilience test as many other similar studies [18,27].

Conventionally, the resilience test is a temporal function, i.e. in discrete time step, one node is removed from the network and we compute the largest connected component remaining in the network. This iteration is repeated until network completely decomposes. The quicker a method decomposes the network, the better its performance. In this article, instead of single node, a batch of k nodes is removed in exactly one iteration and the size of the biggest connected component is measured. The smaller the size of the biggest connected component, the better the identification of influential nodes.

Budget. Selecting the right budget is a widely debated topic. Pareto principle [26] suggests that the budget should be 20%, while [22] argue that this number is much smaller. Leskovec et al. [24] showed that the growth in gain with respect to budget is logarithmic. Nevertheless, this is a debatable topic and out of scope of this study. In the experiments, this budget was fixed to 10% for networks with nodes less than 50 and, 2.5% for all other networks.

3.4 Structural and Nodal Attributes

For a machine learning model to perform optimally, feature selection and data transformation are the key steps. For this experiment, we carefully chose a number of nodal and network level traits which are described in Table 2.

Table 2. Network traits, and heuristics used in the experiments.

Name	Description
Structural attributes	
Nodes	Number of nodes in the network
Edges	Number of edges in the network
Maximum degree	Maximum number of connections of a single node
Average degree	Average number of edges for each node
APL	Average of lengths of shortest path from all pairs of nodes
CC	Connectedness of nodes in the network. Value of 1 means the network is a clique
Diameter	Shortest distance between the two farthest nodes
Density	Ratio between existing network edges and all possible edges among nodes
Average distance	Average distance when connecting two nodes
Triads	Number of triangles, i.e. 3 interconnected nodes
Assortativity	Pearson correlation coefficient of nodes at any side of an edge. Ranging between -1 and 1, negative value means that the high degree nodes tend to connect with low degree nodes, positive value means that the nodes connect with respect to degree similarity
Nodal attributes	
Degree	Number of direct connections of a node to other nodes
Betweenness	The number of times a node falls along the shortest path between two other nodes
Closeness	Reciprocal of the sum of length of shortest paths between a node with all other nodes
Eigenvector	Ranks node u based on not only the degree, but also the degrees of the v nodes that are direct neighbours of u
Eccentricity	The reciprocal of the longest shortest path between a node u and all other nodes in the network v the shortest path between the node v and all other nodes in the network
Heuristics	
Coreness	Defines which core does a node lie in. A k-core of is a maximal subgraph in which each node has at least k degree. If a node belongs to k-core, but not (k+1) core, then its coreness is k. This heuristic is chosen due to its performance as claimed by [2]. [4] computes coreness in $O(m)$ time.
Pagerank	Eigenvector centrality based Webpage ranking algorithm developed by Google Inc. [28]. Pagerank of a node is high if the sum of ranks of its neighbours is also high. This algorithm has proven to be effective on scale
Collective influence	A state-of-art influence maximization algorithm by Morone et al. [27] (see 2). This algorithm identifies weak nodes, which act as bridges between strong influences, and is also scalable to large networks

4 Experiments and Results

4.1 Machine Learning Models

Fernandez et al. [15] argued that the real world classification problems do not require too many machine learning models after comparing 179 classifiers on 121 data sets. Analyzing the results of the best and worst performing models from different families, we used the following models for our experiment:

1. Logistic regression: Generalized Linear Model (**GLM**) by Dobson [13].
2. Decision Trees and Rule-Based: Recursive partitioning (**Rpart**) by [6], and **C5.0**, (an extended version of original C4.5 model).
3. Support Vector Machines: (**SVM**) model using Gaussian kernel from popular LibSVM library [7].
4. Boosting: Random Forests (**RForest**) by [5], and eXtreme Gradient model by [8] (**XGBoost**).

From the dataset, the target variable tells whether the node is identified as an influential node by the Optimal or Greedy algorithm.

The models are evaluated using measure of accuracy, a standard way to compare the performance of machine learning models. We denote true positives as the number of nodes which were correctly classified as influential; true negatives is the number of nodes which wore correctly classified as non-influential; false positive is the number of nodes which were incorrectly classified as influential; finally, false negative is the number of nodes which were incorrectly classified as non-influential. We test the accuracy of the models on various sizes of networks and average out the accuracy of each model on the validation set.

Figure 1 illustrate the accuracy of classification models by size, where overall accuracy is given in Fig. 2. Size of the networks are on horizontal axis, while the accuracy of each model on these sizes are on vertical axis. It is evident that XGBoost has the highest accuracy overall on varying sizes of networks. We also observe that the performance of all models in small graphs is variable. However, XGBoost still outperformed other models on small networks. Therefore, we selected XGBoost based on its performance in comparison with other supervised machine learning models. However, it is to note that the variance in performances of these models is not high. This observation also implies that the application of machine learning classification is not subject to a specific model, but with a small percentage trade-off a broad range of models can be applied to the underlying problem.

4.2 Results

The key objective of this study is to find out if a machine learning algorithm can outperform heuristics in discovering influential nodes in networks of various sizes and structural properties. Previous section explained the methodology followed in order to select and train an appropriate model, i.e. XGBoost model. In order to test our hypothesis, we pass all the test networks discussed in Table 1 to the

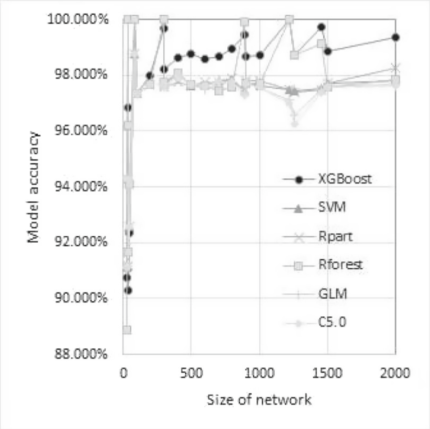

Fig. 1. Accuracy of classification models on various sizes of networks in validation set. The figure demonstrates that the models perform consistently on a variety of sizes.

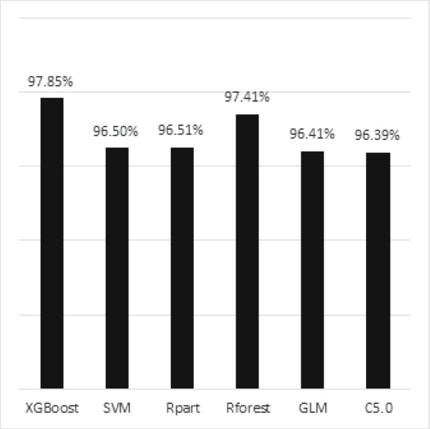

Fig. 2. Average accuracy of classification models on all networks in validation set. The results from previous figure are averaged out for all sizes.

model to identify k most influential nodes. The output obtained from the model is a probability against each node, which quantifies how strongly the model classified that particular node as influential. We pick the set of k nodes of the highest probabilities and perform resilience test on the test networks. Resilience test is destructive, i.e. it is used to measure how much a network decomposed when a set of nodes is removed. After removing k influential nodes identified by the model, we calculate the size of the largest connected component in the remaining network. Therefore the method, which identifies k nodes that results in higher decomposition of the network qualifies as a better performer.

The results shown in Fig. 3 illustrate that the XGBoost model performed very well when compared with commonly used heuristics. In this figure, “size” on x-axis represents the number of nodes in the original test network, while the rest of the bars represent how large the network remained after removing k influential nodes. Best results are highlighted with bold-italic text in the figure.

1. **Author**: this is the smallest of all test networks. XGBoost model collapsed the network to 254 after removing k influential nodes.
2. **ITA 2000**: XGBoost showed best results with network size dropping to 3014, but only marginally better than Pagerank and Betweenness centrality.
3. **AS-CAIDA**: a relatively large network, where XGBoost (score 16264) dominated other methods with significant margin. Eccentricity (score 22756), which is the next best is still had 6492 nodes.
4. **JDK**: Similar results with XGBoost (score 4696). Interestingly, Eigenvector centrality and Eccentricity are only slightly better than random method, indicating that centrality based methods may not universally perform better.

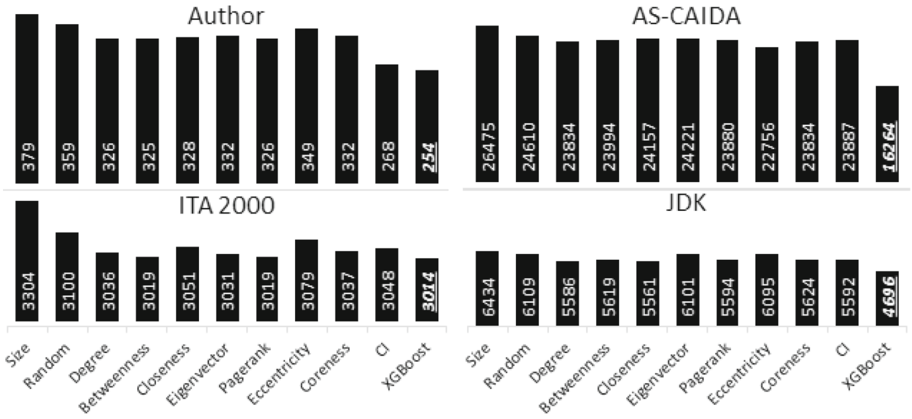

Fig. 3. Comparison of influence mining performance. On x-axis are the influence mining methods under comparison. On y-axis is the number of nodes in largest component after removing k nodes from the network

Table 3. Comparison of heuristics with ML model. XGBoost outperformed all other heuristics, while in the other results, there is no clear second best method.

Network	Author	ITA 2000	AS-CAIDA	JDK
Random	10	10	10	10
Degree	4.5	5	3.5	3
Betweenness	3	**2.5**	7	6
Closeness	6	8	8	2
Eigenvector	7.5	4	9	9
Pagerank	4.5	**2.5**	5	5
Eccentricity	9	9	**2**	8
Coreness	7.5	6	3.5	7
CI	**2**	7	6	4
XGBoost	1	1	1	1

Analyzing the results, although XGBoost model stands atop, analysis on next best also reveals that there is no clear runner up method. Table 3 lists the influence methods and ranks using average ranking method. As observable, CI has 2nd rank in Author network, but fails on other networks; Betweenness and Pagerank perform equally well on ITA 2000, but do not hold this position for other networks; likewise, Eccentricity is impressive on AS-CAIDA network, but performs almost similar to random method on other networks. This observation also validates that heuristics perform differently, based on structure of the network. Therefore, no heuristic can be considered as a generic algorithm to accurately identify influential nodes in a variety of complex networks.

Another observation is that not all networks decompose with the same rate. For example, ITA 2000 network proved highly resilient by sustaining a size of 3014 (91%) after removing influential nodes. On the other hand, AS-CAIDA reduced to 16264 (61%). It is also evident that random methods perform poorly, as indicated by various authors before.

5 Conclusion and Future Work

This study proposed use of supervised machine learning to generalize the problem of influence mining in complex networks of different structural properties. The experiments and results not only validate the proposal by outperforming various conventional and state-of-art methods, but take us a step ahead in terms of prerequisites. The model which was trained for this study was based on synthetic networks, combined with some openly available network data sets. Moreover, it did not require large amount of data and high performance computing machines. The classification outperformed the competing methods in all tested real world networks. While machine learning can certainly help in various applications, it is important to note that training the model is computationally much more expensive than any other heuristic in comparison. It is assumed that anyone applying this solution aims to reuse the model on various networks, and multiple times. For simple one-time applications, the solution is still valid in terms of performance, but has a high computation overhead. Therefore, one of the key areas of optimization is selecting the right set of features which reduces the overall complexity while maintaining the high performance achieved. This study opens up many potential research avenues to explore, and help develop robust methods to identify influential nodes in large complex networks.

References

1. Adamic, L.A., Glance, N.: The political blogosphere and the 2004 u.s. election: divided they blog. In: LinkKDD '05 Proceedings, pp. 36–43. ACM Press (2005)
2. Bae, J., Kim, S.: Identifying and ranking influential spreaders in complex networks by neighborhood coreness. Physica A: Stat. Mechanics **395**, 549–559 (2014)
3. Barabási, A.L., Albert, R.: Emergence of Scaling in Random Networks. Science, pp. 1–11 (1999)
4. Batagelj, V., Zaversnik, M.: An O(m) Algorithm for Cores Decomposition of Networks. arXiv 1(49), 1–10 (2003)
5. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
6. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. CRC Press (1984)
7. Chang, C.C., Lin, C.J., Chang, C.-C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM (TIST) **2**(3), 27 (2011)
8. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference, pp. 785–794 (2016)
9. Chen, W., Collins, A.: Influence maximization in social networks when negative opinions may emerge and propagate. In: SDM 2011 (2011)

10. Chen, Y., Paul, G., Havlin, S., Liljeros, F., Stanley, H.E.: Finding a better immunization strategy. Phys. Rev. Lett. **101**(5), 058701 (2008)
11. Cohen, R., Erez, K., Ben-Avraham, D., Havlin, S.: Resilience of the internet to random breakdowns. Phys. Rev. Lett. **85**(21), 4626 (2000)
12. Csardi, G., Nepusz, T.: The igraph software package for complex network research. Int. J. Complex Syst. **1695** (2006). http://igraph.sf.net
13. Dobson, A.J.: An intro. to generalized linear models. CRC Press (2018)
14. Domingos, P., Richardson, M.: Mining the Network Value of Customers. In: Proceedings of the Seventh ACM SIGKDD International Conference, pp. 57–66 (2001)
15. Fernández-Delgado, M.: Do we need hundreds of classifiers to solve real world classification problems? J. ML Res. **15**(1), 3133–3181 (2014)
16. Goyal, A., Bonchi, F., Lakshmanan, L.: Learning influence probabilities in social networks. In: 3rd ACM Conference - WSDM 2010, p. 241. ACM Press (2010)
17. Guille, A., Hacid, H., Favre, C.: Predicting the temporal dynamics of information diffusion in social networks. In: Proceedings of the 21st International Conference on WWW (2013)
18. Holme, P., Kim, B.J., Yoon, C.N., Han, S.K.: Attack vulnerability of complex networks. Phys. Rev. E **65**(5), 056109 (2002)
19. Hussain, O., Zaidi, F., Rozenblat, C.: Analyzing diversity, strength and centrality of cities using networks of multinational firms. Netw. Spat. Econ. **19**(3), 791–817 (2019)
20. Hussain, O.A., Zaidi, F.: Empirical analysis of seed selection criterion for different classes of networks. In: IEEE - SCA 2013, pp. 348–353 (2013)
21. Jeong, H., Mason, S.P., Barabási, A.L., Oltvai, Z.N.: Lethality and centrality in protein networks. Nature **411**(6833), 41–42 (2001)
22. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: 9th ACM SIGKDD International Conference, p. 137. ACM Press (2003)
23. Landherr, A., Friedl, B., Heidemann, J.: A critical review of centrality measures in social networks. Bus. Inf. Syst. Eng. **2**(6), 371–385 (2010)
24. Leskovec, J., Krause, A., Guestrin, C.: Cost-effective outbreak detection in networks. In: 13th ACM SIGKDD International Conference, pp. 420–429. ACM Press (2007)
25. Lü, L., Chen, D., Ren, X.L., Zhang, Q.M., Zhang, Y.C., Zhou, T.: Vital nodes identification in complex networks. Phys. Rep. **650**, 1–63 (2016)
26. Mornati, F.: Manuale di economia politica, vol. 40. Studio Tesi (1971)
27. Morone, F., Makse, H.a.: Influence maximization in complex networks through optimal percolation: supplementary information. Current Sci. **93**(1), 17–19 (2015)
28. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Tech. rep, Stanford InfoLab (1999)
29. Saito, K., Nakano, R., Kimura, M.: Prediction of information diffusion probabilities for independent cascade model. Lecture Notes in CS **519**, 67–75 (2008)
30. Smith, D.A., White, D.R.: Structure and dynamics of the global economy: network analysis of international trade 1965–1980. Soc. Forces **70**(4), 857–893 (1992)
31. Watts, D.J., Strogatz, S.H.: Collective dynamics of a small-world' networks. Nature **393**(June), 440–442 (1998)
32. White, J.G., et al. E.S.: The structure of the nervous system of the nematode c. elegans. Phil. Trans. R. Soc. London, B: Bio. Sci. **314**, 1–340 (1986)
33. Zanin, M., Papo, D.: Combining complex networks and data mining: Why and how. Phys. Rep. **635**, 1–44 (2016)