Stanford's Web graph Analysis.

Gayathri Pendyala
gayathripendyala@vt.edu
Virginia Polytechnic Institute and
State university.
Blacksburg, Virginia, USA

Pranav Vishwanatham
pranav0909@vt.edu
Virginia Polytechnic Institute and
State university.
Blacksburg, Virginia, USA

Saikiran Reddy Ramacharla
kiran1906@vt.edu
Virginia Polytechnic Institute and
State university.
Blacksburg, Virginia, USA

ABSTRACT

Web Graphs hold the structure of complex network of websites that function as a digital representation of human knowledge and interaction. In order to understand the behaviour and structure of the web, we have utilized the Stanford's Web Graph Data from 2002. Due to the dataset's size, which consists of 281903 nodes and 2312497 edges and limited amount of resources, performing computational tasks on it would be costly. To prevent this, we have conducted degree-based sampling to extract a smaller subset of the data, containing 13142 nodes and 359114 edges. At first, we used the Gephi software and NetworkX to look at the graph and gain some initial understanding of the network. The graph shows a high level of connectivity, with 61 communities at the highest level and a significant number of large cliques. Next, we evaluated the Pagerank algorithm as a baseline by comparing it with the Hot Kernel Pagerank and Classic Pagerank algorithms. The Mean Squared Error was used as the metric for comparison between these algorithms. In addition, we utilized the Networkx and Pytorch library to conduct Link Prediction. We then compared the effectiveness of the Adamic-Adar and Preferential Attachment algorithms with a Graph Convolutional Network. In this study, we provide a comprehensive framework to analyze web graphs while comparing various algorithms.

KEYWORDS

Web Graphs, Network Analysis, Graph Sampling, Gephi, NetworkX, Community Detection, Pagerank Algorithm, Link Prediction, Graph Convolutional Network.

1 INTRODUCTION

The internet is an extensive and complex network of websites that functions as a digital representation of human knowledge and interaction. In this network, each website is considered a node, and the hyperlinks that connect them are seen as edges, creating an intricate web of relationships. This study specifically examines a portion of online information from Stanford University, consisting of a network diagram with 13,142 individual points and 359,114 connections. While this subset only accounts for a small portion of the extensive dataset consisting of 281,903 websites and 2,312,497 links, it offers a practical yet comprehensive setting for thorough analysis.

The main aim of this study is to classify websites into separate groups and determine the most influential nodes within each group. Through this approach, we can reveal the underlying organization and actions of the internet, providing valuable information about which websites have the greatest impact on user involvement and interconnectivity.

The motivation for this research is inspired from the practical requirements and intellectual curiosity arising from the vast scale and intricate nature of the internet. Search engines necessitate a comprehensive comprehension of web architecture in order to optimize their algorithms and user interfaces, thereby enhancing the accuracy and efficiency of search results. Academic researchers and the industry is interested in understanding the internet's topology as it helps enhance comprehension of digital information systems and their development. Moreover, businesses can gain advantages from this analysis by pinpointing crucial websites for focused advertising, enhancing their marketing endeavors to effectively reach the most involved audiences[1].

The purpose of this study is to contribute to the progress of technology in navigating and utilizing the internet more efficiently, while also enhancing the academic discussion on the dynamics of network ecosystems.

2 BACKGROUND

This section of our study offers a fundamental overview of the terminology and concepts that are crucial for understanding graph theory in relation to web graphs. We analyze various types of graphs, including directed and undirected graphs, and examine important metrics such as node degree, graph diameter, and shortest path lengths that are essential for our analysis. The section also examines community detection, providing a detailed analysis of techniques used to identify clusters within networks. These clusters can unveil the hidden structures within the web. In addition, we explore the PageRank algorithm, providing an explanation of its historical evolution and its importance in determining the ranking of nodes in a network. Finally, we analyze link prediction methods that forecast future connections in the network, a crucial element for improving the precision and significance of web searches and network investigations.

2.1 Graph Terminology

In this section, we break down some key terms that form the backbone of our exploration into web graphs. At its simplest, a graph is made up of points called nodes (or vertices) that represent items like websites, and lines called edges that connect them, representing links between sites. These connections can be one-way if we're looking at a directed graph—imagine a one-way street where information flows from one point to another—or two-way in an undirected graph, similar to a regular two-way street.

Each node has what we call a degree, which is just a count of how many connections it has. In a directed graph, we get more specific: the 'in-degree' counts incoming connections, and the 'out-degree' counts the outgoing ones[2]. Imagine checking your inbox and outbox to see how many emails you've received and sent—that's similar to in-degree and out-degree.

We also talk about the diameter of a graph, which is the longest of all the shortest paths between any two nodes. It's a bit like measuring the furthest distance you'd need to travel between two points in a network of roads. Then, there's the average shortest path length, which gives us an idea of how interconnected the graph is, by averaging the shortest paths between all pairs of nodes[3].

Lastly, we discuss connected components. A strongly connected component means that there is a way to get from any node to any other node in that component, considering the direction of edges—think of it as being able to circle around a roundabout to reach any exit. A weakly connected component is where the direction can be ignored—more like a free-for-all space where you can walk directly to any point from any other.

2.2 Link Prediction

In this Link Prediction section, we look at methods to predict future connections between pairs of websites through examining patterns observed in the current network.

We discuss various methodologies for making these predictions. A quick and easy method is to consider common neighbors. If two websites have many connections in common, it is more probable that they will also link to each other. Consider it as a situation where two individuals who share numerous common connections are more inclined to encounter each other. Another method is the Adamic-Adar index which enhances the concept by assigning greater significance to connections that have lower overall connectivity, suggesting that these rarer links may carry more importance. In addition, the concept of preferential attachment proposes that new links are more likely to be established with nodes that already have a substantial number of connections. This is similar to well-known websites or celebrities on social media who are more likely to gain new followers simply due to their visibility and existing popularity.

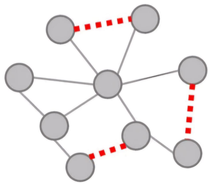

Figure 1: Image showing Link prediction between nodes

By applying these principles, we can create models which help in predicting future connections in the web graph, which improves our knowledge of the internet's structural evolution. It is importance for tasks such as enhancing search engine rankings and understanding the distribution of information on the internet.

2.3 Community Detection

The Community Detection section looks into the process of finding groupings or clusters in the web graph where the nodes (websites) are more closely connected to one another than to the other nodes in the network.

We consider multiple approaches for identifying these communities: from straightforward methods based on graph theory to more complex ways including statistical models or machine learning. For example, modularity-based approaches enable us to identify the distinct communities inside a network, similar to identifying the groups at a party that engage more with each other than with others.Community detection is quite helpful since it enables us to observe the division of information on the internet. This can enhance the way results are arranged and displayed for search engines. Studying these communities can provide marketers and researchers with insights on online social dynamics and consumer behavior[4].

We may obtain important insights into the information flow inside these communities and enhance our understanding of the web's structure by plotting them out. These insights can be critical for a variety of purposes, including academic research and digital marketing.

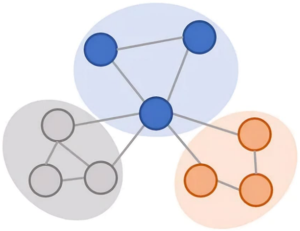

Figure 2: Image showing Community detection between nodes

2.4 Page Rank Algorithm

In the section on the 'Page Rank Algorithm', we explore in determining the relative significance of websites. Consider a scenario of a popularity contest where the weightage of votes varies, with a vote from the king or queen carrying more significance than a vote from an ordinary individual. That is essentially the mechanism behind PageRank. Originally created by Google's founders, it brought about a revolutionary change in the way websites are ranked in search results.

PageRank algorithm is based on a fundamental concept: a website is deemed important if it is linked to by other significant websites. Every hyperlink to a webpage serves as an endorsement. Nevertheless, not all votes carry the same weight. A hyperlink originating from a site with a strong reputation holds greater value compared to a hyperlink originating from a site with a lower level of recognition. In addition, PageRank takes into account the probability of randomly transitioning from one page to another. The issue at hand involves not just tallying the number of explicit votes, but also considering the likelihood of randomly accessing a webpage. This guarantees that the rankings accurately represent both direct endorsements and general accessibility.

By using the PageRank algorithm on our web network, we may begin to identify the nodes (websites) that are genuinely central and significant. Understanding online mechanics and search engine optimization are not just academic pursuits, but also essential for various purposes.

3 APPROACH

We have utilized Stanford's web graph data taken from the website to study the web as a graph. First, we started sampling the data to 40k nodes since the data that we originally had was expensive to compute. But even after sampling the data to 40k, it took us much longer to perform computations on it. Later we downsized the data to 26k and later to 13,142 which worked best for us to compute the entire data. Our code is based on the degree of its nodes and from a given directed graph G it extracts a subgraph. For this top N nodes from the data are chosen. A sum of in-degree and out-degree which gives us the total degree of nodes is calculated. Our code then determines if the subgraph has a minimum level of connectedness or not to make sure that the subgraph is strongly connected and there Is a directed path between every pair of nodes of the subgraph is determined. From the NetworkX package, methods for in-degree and out-degree are used.

The performance of two algorithms of the PageRank is then compared to the actual PageRank scores generated by the networkx's pagerank function. The two implemented versions of PageRank are Heat Kernel PageRank and Classic PageRank. The traditional PageRank algorithm, referred to as "classic PageRank," evaluates the importance of individual nodes according to the number and quality of links that point to them. It replicates a directed network, similar to the web. Heat Kernel PageRank is a version of PageRank that utilizes the heat kernel, a fundamental concept in the field of graph theory, to determine the degree of similarity between nodes in a graph. In order to enhance the computation of PageRank scores, a function is defined to convert the subgraph into a sparse adjacency matrix[5]. The PageRank scores are calculated utilizing the sparse adjacency matrix in the implementation of the Classic PageRank algorithm by the classic Pagerank function. The self-defined heat kernel pagerank function is utilized to implement the Heat Kernel PageRank algorithm, which computes PageRank scores utilizing the heat kernel of the subgraph. The actual PageRank scores are eventually computed utilizing the NetworkX built-in PageRank function in addition to the previously defined functions for the Classic and Heat Kernel PageRank scores. To assess the relative efficiency of the two algorithms, the mean absolute error and root mean squared error are computed between the ground-truth, Classic, and Heat Kernel PageRank scores[4].

Next, we generate a roster of nonexistent edges that symbolize potential future links within the graph. We compute the Adamic-Adar index, and Preferential attachment score for every edge. These metrics are commonly used for link prediction tasks. The Adamic-Adar index is calculated using the Link-prediction.adamic-adar-index function, which produces a dictionary of scores. Preferential attachment score are computed using the nx.preferential-attachment functions, respectively. Once we have calculated these metrics for all possible edges, we arrange the scores in a descending order and determine the top 10 predictions for each method. These top predictions serve to emphasize the potential new connections that the graph may form. During this procedure, mathematical formulas are utilized to comprehend the implementation of link prediction algorithms in NetworkX.Further, Graph neural networks (GNNs), particularly graph convolutional networks (GCNs), can be essential for link prediction tasks in complex networks such as web graphs. We leveraged Pytorch framework to build a Graph convolutional network for link prediction.

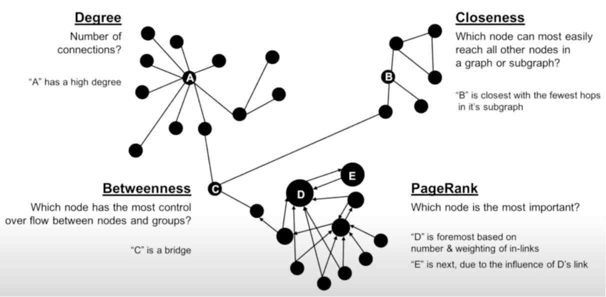
Degree
Number of connections?
"A" has a high degree
A
Closeness
Which node can most easily reach all other nodes in a graph or subgraph?
"B" is closest with the fewest hops in it's subgraph
B
Betweenness
Which node has the most control over flow between nodes and groups?
"C" is a bridge
C
D
E
PageRank
Which node is the most important?
"D" is foremost based on number & weighting of in-links
"E" is next, due to the influence of D's link

Figure 3: Types of centralities.

4 EXPERIMENT

In this section, we will analyze the outcomes and their significance in accordance with the methodologies we have previously discussed. This section has been divided into four components:

4.1 Network Analysis

Our network research and visualization are conducted using two amazing tools: Gephi, an open-source software, and Networkx, a robust Python library. Based on our initial research, here is our learning: The network exhibits an average degree of 27.326, indicating that, on average, each node is connected to around 27 additional nodes. The network width is 23, which means that the greatest distance between the furthest nodes is 23 steps[6].

The graph density is extremely low, measuring at 0.002. This indicates that the network, despite its size, is highly sparse, with a limited number of connections in comparison to the total number of possible connections. The network's modularity is 0.796, which signifies a high degree of modularity and the existence of distinct communities or clusters within the network. The high modularity of our network indicates that it can be efficiently separated into groups that have stronger internal connections compared to their connections with the rest of the network[7].

In addition, the average clustering coefficient is 0.507, indicating that nodes have a tendency to form cohesive groupings by clustering together. This coefficient quantifies the probability that two adjacent nodes in a network are connected, hence indicating the network's propensity to create localized clusters.

Finally, the average path length is much extended, reaching 8.514. This implies that the network is easily traversable, yet it covers a broad expanse, potentially reflecting a multitude of connections across various contexts or subjects. The extensive path length also emphasizes the magnitude and complexity of the network structure we are studying. Further investigation will investigate these features in greater depth in order to gain a comprehensive understanding of the complex connections and structural complexities of our web

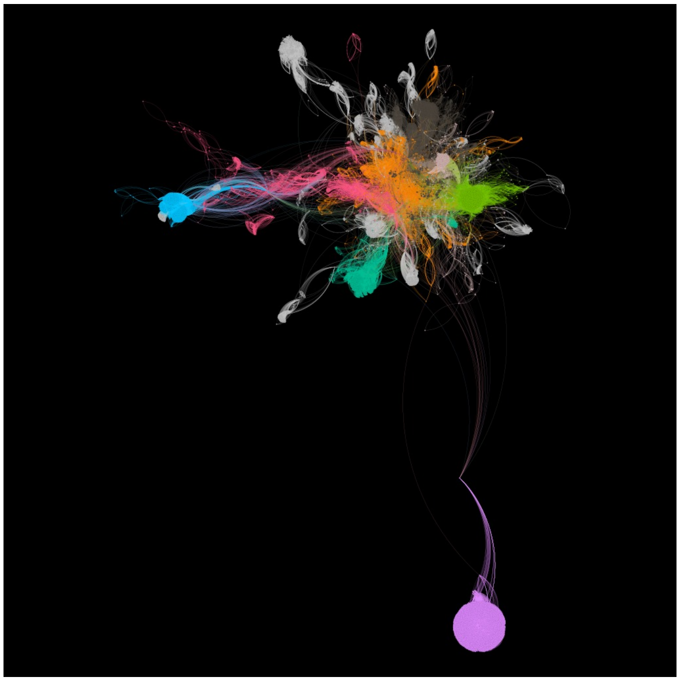

Figure 4: Network structure visualization using Gephi

graph. To get a visual representation of these findings, please refer to the graphical outputs in Gephi displayed in figure 4.

4.1.1 Graph Visualization. For our study, we utilized Pyvis, a robust tool that allows us to visualize networks directly within Jupyter notebooks or as separate HTML pages. This tool works with NetworkX, which is ideal for digging into network analysis, while Pyvis enhances the visual appeal. By integrating Pyvis into our workflow, we successfully generated attractive and interactive graphs that effectively highlighted the complex relationships and patterns inside our web graph. These engaging representations were innovative, simplifying the understanding of the complex connections we were investigating for both people with technological skills and those who are not. This methodology not only enhanced our ability to analyze data, but also simplified the spread of our findings, making them more accessible and effortless to share with others. You can see the visualization in figure 5.

4.2 Page Rank Analysis

When assessing the effectiveness of our prediction models, we applied measures such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to measure the degree of similarity between the predicts and the actual values. The Classic PageRank method achieves a very low Mean Absolute Error (MAE) of roughly 0.000093 and a similarly low Root Mean Square Error (RMSE) of about 0.000492. The Heat Kernel PageRank algorithm has remarkable accuracy, achieving a Mean Absolute Error (MAE) of approximately 0.000093 and a Root Mean Square Error (RMSE) of roughly 0.000492 [8].

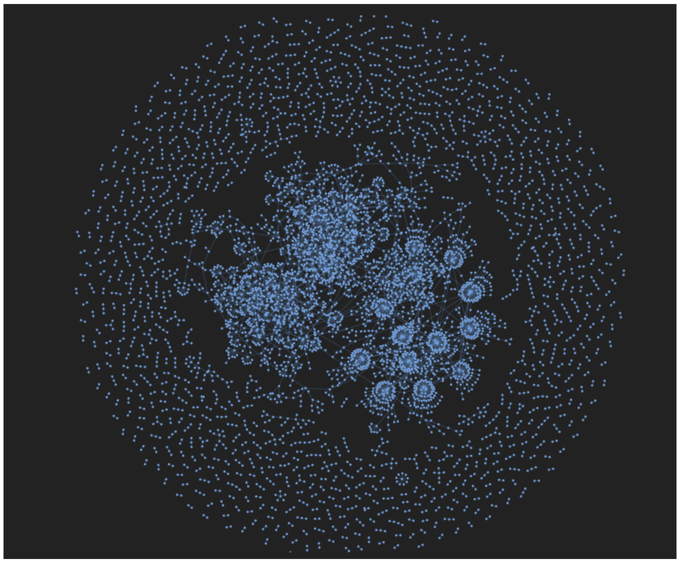

Figure 5: Image showing Visualization of web pages

Metric	Value
Average Degree	27.326
Avg. Weighted Degree	27.326
Network Diameter	23
Graph Density	0.002
Connected Components	1
Modularity	0.796
Statistical Inference	1206174.258
Avg. Clustering Coefficient	0.507
Avg. Path Length	8.514

Table 1: Network Overview Metrics from Gephi

The remarkably low error figures indicate that both the Classic and Heat Kernel PageRank algorithms possess a high level of accuracy in assessing the importance of nodes in the web graph. The similarity in the error metrics indicates that both methods yield accurate rankings for the nodes. It is crucial to acknowledge that certain forms of web graphs, particularly those with dense clusters or numerous irrelevant links, may necessitate modifications to these algorithms in order maintain this level of accuracy.

Algorithm	MAE	RMSE
Classic PageRank	0.000093	0.000492
Heat Kernel PageRank	0.000093	0.000492

Table 2: MAE and RMSE of PageRank Algorithms

4.3 Link Prediction

4.3.1 Adamic-Adar Index. The Adamic-Adar index estimates the probability of future connections between nodes in a graph by considering the number of shared neighbors they have. The concept is that nodes with a greater number of common neighbors are more likely to establish a connection. The Adamic-Adar index has been used to predict linkages, and our analysis has identified the top 10 predictions. These predictions involve node pairings that have obtained high scores, indicating a high likelihood of future links.

The node pair (84906, 90543) has the highest score of around 372.83, suggesting a high probability of close association and potential future linking between these nodes. Two further significant pairs are (82476, 151707) and (3164, 82476), which have scores of 342.70 and 342.67, respectively. These scores indicate that these nodes also have a sufficient number of shared connections, which may lead to the establishment of interconnections in the near future.

The consistent high scores observed in pairs such as (204315, 82476) and (235570, 204315), both scoring approximately 327.32, provide additional evidence for the effectiveness of the Adamic-Adar index in predicting probable future connections within the web graph.

Pair of Nodes	Score
(84906, 90543)	372.83
(82476, 151707)	342.70
(3164, 82476)	342.67
(82476, 90543)	342.57
(3164, 204315)	327.74
(204315, 82476)	327.58
(235570, 204315)	327.32
(204315, 151707)	327.32
(204315, 259439)	327.32
(204315, 90543)	327.19

Table 3: Top 10 Link Predictions Using Adamic-Adar Index

4.3.2 Preferential Attachement. Preferential attachment is a concept that suggests nodes in a network have a tendency to form connections with nodes that have a high degree of connections and are well-established. The method produces the top 10 link predictions that involve pairs of nodes, where at least one member of each pair has a significant amount of connections. This observation provides evidence for the idea that nodes with a higher number of connections are more appealing for the establishment of new interconnections[9].

For instance, all the important pairs, such (137632, 226411), (177991, 226411), and (176790, 226411), show the node 226411, which seems to possess an exceptionally high degree. 226411 appears to function as a key node in the network, facilitating extensive connections with many nodes. Like node 226411, other nodes such as 17781, 181701, and 183004 also occur numerous times in the top predictions, suggesting their strong connection and central positions in the network[10].

The findings highlight the basic idea of preferential attachment, which states that nodes with a higher number of current connections are more likely to acquire new connections. Although this set of predictions primarily considers a small number of nodes with strong connections, using other link prediction techniques could produce a more general and thorough collection of potential future links. This would improve the overall accuracy and robustness of our network study.

Node Pair	Score
(137632, 226411)	11249070
(77999, 226411)	11249070
(176790, 226411)	11249070
(17781, 226411)	11249070
(181701, 226411)	11249070
(183004, 226411)	11249070
(120708, 226411)	11249070
(247241, 226411)	11249070
(221087, 226411)	11249070
(62478, 226411)	11249070

Table 4: Top 10 Link Predictions Using Preferential Attachment

4.3.3 Graph Neural Networks. Graph neural networks (GNNs), particularly graph convolutional networks (GCNs), are essential for link prediction tasks in complex networks such as web graphs. Graph neural networks are good at collecting the structural information and relational relationships that are characteristic of graph data. This makes them well-suited for tasks such as detecting missing edges or linkages between nodes[6]. The PyTorch architecture we created for link prediction combines GATConv layers for graph attention, BatchNorm for normalization, and dropout for regularization. These components have been included to improve the model's capacity to generalize and prevent overfitting. The forward pass of the model consists of applying the Rectified Linear Unit (ReLU) activation function after each layer, and then applying dropout. Dropout helps in the learning of strong representations of nodes and their connections. The decode function utilizes the acquired embeddings to predict edges connections between nodes. The optimizer employs learning rate scheduling and weight decay to optimize the model's parameters during training. The BCEWithLogitsLoss function is employed to train the model specifically for binary classification tasks, such as edge prediction[4].

Table 5: Comparison of Learning Rates and Performance Metrics

Learning Rate (lr)	AUC-ROC Score	Accuracy	F1-Score
0.1	0.9845	0.9840	0.9837
0.01	**0.9995**	**0.9989**	**0.9989**
0.001	0.9992	0.5000	0.6667

The results of our comparison of learning rates for link prediction using graph neural networks demonstrate diverse performances depending on the selected learning rate. With a learning rate of 0.1, the model demonstrates a Test AUC-ROC score of 0.9844, signifying its robust ability to differentiate between positive and negative samples. The Test Accuracy and F1-Score exhibit strong values of 0.984 and 0.983. Nevertheless, when the learning rate is reduced to 0.01, there is a notable enhancement in performance. The Test AUC-ROC score rises to 0.9994, accompanied by nearly full Test Accuracy and F1-Score values. Conversely, when the learning rate decreases to 0.001, the Test AUC-ROC score stays high while the Test Accuracy decreases to 0.5, suggesting inadequate performance in accurately categorizing positive and negative data. The decrease in accuracy indicates that a learning rate of 0.001 might be insufficient for successful training, resulting in unsatisfactory performance of the model. Ideally, these values underscore the model's capacity to accurately predict connections between nodes in the graph. However, we feel that there's a chance of model overfitting and we would like to revisit the existing architecture and analyze it to make the model robust and improve its performance. It is crucial to find a compromise between various parameters like learning efficiency and model performance.

4.4 Community Detection

We performed community detection and dendrogram analysis to provide a thorough understanding of the structure of the webgraph. Within the network, there are many nodes that have strong connections, creating resilient community clusters. The dendrogram analysis revealed a distinct hierarchy of communities at various levels, each exhibiting a high modularity value.

At every level, the highest-ranking communities exhibit unique measures, emphasizing differences in their characteristics. Communities at level 0 have diameter that range from 1 to 2 and densities between 0.503 and 1.0. At the first level, communities have a diameter ranging from 1 to 6 and display densities between 0.218 and 1.0. At level 2, communities exhibit diameters ranging from 1 to 5, with densities ranging from 0.273 to 1. The graph exhibits a propensity for nodes to form tight clusters, as evidenced by the elevated average clustering coefficient. This pattern suggests that the webgraph may reflect a network of interconnected websites[8].

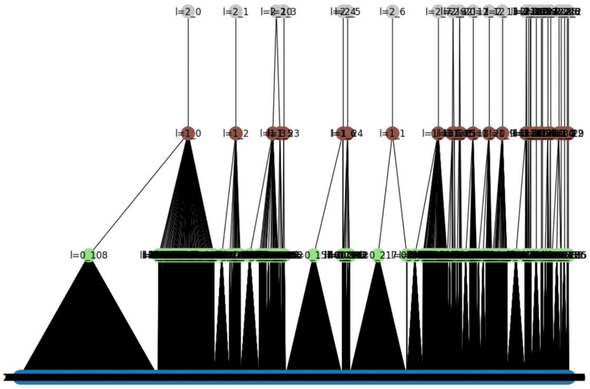

Figure 6: Image showing community formation

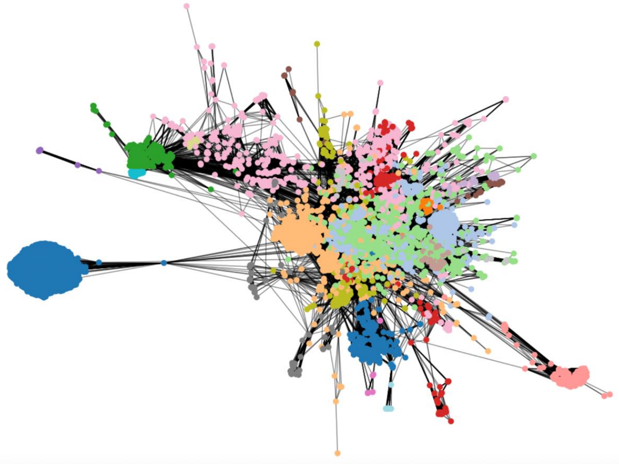

Figure 7: NetworX Graph

Within the largest community consisting of 13142 nodes, there are multiple cliques, indicating a highly integrated set of websites. These findings suggest that the webgraph is a densely interconnected network with a strong community structure. The high modularity values and different qualities observed in the top communities at each level indicate that there is a large variety of characteristics among the nodes and links in the graph.

5 CONCLUSION

To summarize, our analysis with Gephi and the Python NetworkX module indicates that we are dealing with a web graph that is extensively interconnected and exhibits a strong community structure. The subgraph we sampled closely resembles the wider web graph, as indicated by its diameter, average degree, and network density, along with high values in modularity and clustering coefficients. The community detection methods we used revealed a distinct hierarchy of communities at different levels, each characterized by substantial modularity. This demonstrates the diverse nature of these communities.

To predict future connections between nodes, we utilized metrics derived from both the preferential attachment and the Adamic-Adar index. Both individuals offered valuable perspectives, but their predictions did not coincide, indicating that a combination of different research approaches could produce more thorough and accurate results. In addition, the accuracy of our predictions is enhanced by the low values of mean absolute error and root mean squared error obtained by our Classic PageRank and Heat Kernel PageRank algorithms, compared to the baseline provided by NetworkX's built-in PageRank.

However, in the case of web graphs with a significant concentration of spam connections or considerable clustering, it may be essential to make modifications to the PageRank algorithms in order to ensure confidence. Our study of the webgraph's structure and its connection to the larger web graph using different metrics and algorithms has not only confirmed the structure of the web graph, but also highlighted the possibility of predicting future links. These findings allow additional research and the development of improved methods for understanding and going through complex web graphs.

6 FUTURE WORK

Utilizing advanced machine learning models has the potential to enhance the accuracy and dependability of link prediction. Graph Neural Networks can be optimized to reduce overfitting and enhance generalization across different network architectures. Advanced statistical techniques have the potential to reveal intricate patterns of community dynamics and their changes over time in community detection analyses. It is possible to develop tools that can handle larger datasets by focusing on navigation, visualization, and computational efficiency. This comprehensive approach will enhance understanding of network structures and dynamics, benefiting both theoretical research and practical applications.

REFERENCES

[1] Aric Hagberg, Pieter Swart, and Daniel S Chult. "Exploring network structure, dynamics, and function using NetworkX". In: 2008.

[2] Roberto Marmo. "Social media mining". In: *Encyclopedia of organizational knowledge, administration, and technology*. IGI Global, 2021, pp. 2153–2165.

[3] Bernhard Rieder. "What is in PageRank? A historical and conceptual investigation of a recursive status index". In: 2. 2012.

[4] Dmitri Goldenberg. "Social network analysis: From graph theory to applications with python". In: 2021.

[5] Xing Su et al. "A comprehensive survey on community detection with deep learning". In: IEEE, 2022.

[6] Riju Bhattacharya, Naresh Kumar Nagwani, and Sarsij Tripathi. "CommunityGCN: community detection using node classification with graph convolution network". In: vol. 57. 4. Emerald, 2023, pp. 580–604.

[7] Ravi Kumar et al. "The Web as a graph". In: (2000), pp. 1–10.

[8] Linyuan Lü and Tao Zhou. *Link prediction in weighted networks: The role of weak ties*. Vol. 89. 1. IOP Publishing, 2010, p. 18001.

[9] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. "Gephi: an open source software for exploring and manipulating networks". In: *Proceedings of the international AAAI conference on web and social media*. Vol. 3. 1. 2009, pp. 361–362.

[10] Petr Chunaev. "Community detection in node-attributed social networks: a survey". In: vol. 37. Elsevier, 2020, p. 100286.