# Predictive Modeling and Analysis of Processes on Network Graphs in R

Daniel Last-Name, Logan Flaherty

*Department of Computing and Technology*

*East Tennessee State University*

Johnson City, Tennessee

email@etsu.edu

email@etsu.edu

***Abstract*— This report explores techniques for modeling and predicting processes on network graphs using R. It applies nearest-neighbor methods, Markov Random Fields (MRF), and kernel regression to a protein-protein interaction dataset (ppi.CC). The study also simulates dynamic processes such as the Susceptible Infected Recovered (SIR) model on various network topologies. The results highlight significant improvements in prediction accuracy with advanced methods and demonstrate the important influence of topology on dynamic processes. Graphs generated via R are presented for clarity.**

***Keywords*— Network Analysis, Markov Random Fields, Kernel Regression, Epidemic Modeling, Protein-Protein Interactions, SIR model (Susceptible, Infected, Recovered).**

## I. Introduction

Network graphs are important in representing interactions between data in various systems. Beyond the structural representation of networks, attributes associated with vertices are often of interest, as these attributes can influence or be influenced by network interactions. For instance, social behaviors, protein functions, and disease spread are examples of vertex-associated processes. These processes are classified as static or dynamic. Static processes are attributes that are inherently independent of time, remain constant, or represent a snapshot of a dynamic process at a given time. Dynamic processes are time-varying attributes that evolve based on network conditions. Modeling these complex processes involves statistical inference, parameter estimation, and prediction. These methods enable a better understanding of the relationship between network structure and vertex properties and show how they can address physical issues in the world.

## II. Rationale of the Project

This project focuses on the yeast protein-to-protein interactions (ppi.CC) dataset, a network of 134 proteins with 241 interactions. This represents an ideal "testbed" because of its biological relevance and manageable size. Each protein is marked with attributes such as intracellular signaling cascade involvement. This project aims to predict the intracellular signaling cascade (ICSC) status of proteins and simulate epidemic processes to study the influence of network topology on disease spread. The techniques we discuss/employ include probabilistic Markov Random Fields models, kernel regressions, and the SIR model for dynamic processes. Through these methods, this report seeks to demonstrate how network-based modeling can address real-world problems.

## III. R/Coding Set Up Details

For the project, we used the R Studio software version 4.3.3 and downloaded the R code to create the simulation at a GitHub repository here https://github.com/kolaczyk/sand/blob/master/sand/inst/code/chapter8.R. Furthermore, three lines in the code must be changed a small amount to account for a deprecation in a function and warnings. On line 34 in the chunk 5 region the function nei(x) was deprecated and must be replaced with .nei(x). If this change is not implemented, then the code will stop running and the simulation will not progress. Last quick changes, on lines 56 and 61 'force = TRUE' must be added to the install call to force install the packages. Note: we initially thought just running the code line by line instead of all at once fixed this however, it does process the data correctly resulting in wrong and missing graphs. Now the code can be run so that the necessary packages/dependencies will be installed. R Studio handles this seamlessly along

with continuing code execution if the code is just run without any setup. However, in the case of setting up the packages beforehand or to verify our setup, presented below is an overview of our set of dependencies with their purpose:

- batchmeans: consistent batch means estimation of monte carlo standard error.
- BiocManager: access the Bioconductor Project Package Repository.
- cli: helpers for developing command line interfaces.
- cpp11: a C++ 11 interface for R's C interface.
- glue: interpreted string literals.
- GO.db: a set of annotation maps describing the entire gene Ontology.
- GOstats: Tool for manipulating GO and microarrays.
- igraph: network analysis and visualization.
- igraphdata: a collection of network data sets for the 'igraph' package.
- kernlab: kernel-based machine learning lab.
- lifecycle: manage the life cycle of your package functions.
- magrittr: a forward-pipe operator for R.
- ngspatial: fitting the centered autologistic and sparse spatial generalized linear mixed models for areal data.
- Org.Sc.sgd.db: Genome wide annotation for Yeast.
- pkgconfig: private configuration for 'R' packages.
- Rcpp: seamless R and C++ integration.
- RcppArmadillo: 'Rcpp' integration for the 'Armadillo' templated linear algebra library.
- rlang: functions for base types and core R and 'Tidyverse' features.
- sand: statistical analysis of network data with R, 2nd edition.
- vctrs: vector helpers.

## IV. R/Coding Running Details

After ensuring the proper setup has been completed, the running of this simulation is quite simple by hitting run all within the 'Code' menu under the 'Run Region' tab or the command 'Alt + Ctrl + R' can quickly be used. Going forward will be a discussion on what the function of each chunk(s) in the code is.

Chunk 1 sets up the environment for reproducible results by setting the seed, loading the sand library, and loading the dataset containing a protein-to-protein interaction (PPI) network, with proteins represented as nodes and their interactions as edges. Chunk 2 outputs basic statistics, including the number of vertices (134) and edges (241), and a list of vertex attributes. These attributes describe protein properties or annotations. Chunk 3 displays the ICSC attribute for the first 10 vertices in the graph. Chunk 4 visualizes the network, coloring vertices based on the ICSC attribute. Chunk 5 computes the average ICSC values for the neighbors of each vertex. Chunk 6 creates two histograms showing the proportion of neighbors with ICSC for vertices with ICSC == 1 and ICSC == 0. Chunk 7 predicts ICSC for each vertex based on whether the neighbor average exceeds 0.5, and then calculates the error rate of predictions. Chunks 8-9 install necessary Bioconductor packages for analyzing gene ontology data. Chunks 10-14 extract updated annotations for proteins involved in intracellular signaling transduction. Identifies proteins with newly discovered functions and compares nearest-neighbor averages for these proteins. Chunks 15-19 use the ngspatial package to fit an auto-logistic model. Chunks 20-27 analyze the auto-logistic model by displaying coefficients for the network effect and then evaluating prediction accuracy. Then finally a second model with additional gene motif information and compares results. Chunks 28-30 simulate new vertex attribute configurations using the fitted auto-logistic models and calculate assortativity to assess the quality of the model's fit. Chunks 31-32 perform Laplacian eigen-decomposition which is a matrix represented as a graph for purposes of telling how well a graph is connected. Chunks 33-42 implement kernel regression using the kernlab package for the purposes of constructing Laplacian and motif-based kernels as well as training Support Vector Machines (SVMs) for prediction. Finally chunks 43-48 simulate epidemic spread on different random networks using sir() to model disease transmission and then plot the results, showing how network structure affects epidemic dynamics.

## V. Provision That Could Be Added

Three provisions could be added to the project to further improve the model's ability to solve more complex and realistic problems. They are as follows:

1) Bayesian Markov Random Fields

One of the limitations of the auto-logistic Markov Random Fields models used in this project is their inability to quantify uncertainty in predictions. Bayesian Markov Random Fields extends traditional Markov Random Fields by incorporating probabilistic priors into the model parameters. This allows for more "robust" predictions, especially in cases where the dataset may have missing values or "noisy" observations (errors in the true value and observed value). By introducing prior distributions over parameters, Bayesian Markov Random Fields can better capture the variability and uncertainty that come with real-world networks

For example, in the context of protein interaction networks, uncertainty may come from incomplete data or errors in experimental results. A Bayesian approach could provide confidence intervals for the predicted attributes, allowing researchers to prioritize high-confidence predictions for further experimental validation. This framework could also accommodate hierarchical priors, enabling the integration/addition of more biological information, such as pathway enrichment or protein domain information, to further improve prediction accuracy.

Implementing Bayesian Markov Random Fields would require more computational resources because of the need for sampling techniques like Markov Chain Monte Carlo (MCMC) or Variational Interference. However, advancements in computing power and efficient algorithms have made this increasingly feasible. The ability to model uncertainty would make the predictions more reliable and interpretable.

2) Dynamic Graph Models

The networks analyzed in this project are static, meaning the structure and attributes remain constant throughout the analysis. However, many actual real-world networks are dynamic, with constantly changing connections and attributes over time. Dynamic graph models account for this temporal evolution by incorporating time-dependent features, enabling the study of processes like the spread of information, diseases, or influence in social networks as they unfold.

In biological terms, dynamic graphs could model protein interactions that vary under different cellular conditions (like stress or disease). For example, some protein-protein interactions may only occur during specific phases of a cellular process (like mitosis). Adding temporal data would allow for a more accurate representation of the biological system and better predictions of protein functions under varying conditions.

Dynamic graph models could also involve "time-varying adjacency matrices" or sequential snapshots of the network. Techniques like temporal random walks or dynamic extensions of graph convolutional networks (GCNs) could also be applied. These models could then be integrated with dynamic versions of the SIR model to better simulate disease progression in networks where connections change, such as the patterns that humans travel in during an outbreak.

With dynamic graph capabilities, the project could handle a larger range of real-world scenarios and provide a more comprehensive understanding of the interaction between network structure and time-dependent processes.

3) Multi-Class Prediction

The models used in this project are limited to predicting binary attributes. However, many actual real-world problems involved multi-class attributes, where nodes can belong to more than two categories. For example, proteins could be classified into multiple functional categories, or individuals in a social network could have various roles (influencers, followers, or passive participants).

Expanding to a multi-class prediction would significantly enhance the applicability of the models. Techniques like multinomial logistic regression, multi-class support vector machines (SVMs), or deep learning-based graph neural networks (GNNs) could be employed to handle multiple categories. For

instance, in the context of kernel regression, combining multiple kernels tailored to different classes could improve the classification accuracy.

Also, hierarchical multi-class prediction could be implemented where classes are organized in a tree or graph structure. With biological networks, this could align with hierarchical "gene ontology" terms, enabling predictions that account for relationships between function categories. This would both improve accuracy and provide richer insights into the biological roles of proteins.

Implementing this multi-class prediction would require adjustments to both the training process and evaluation metrics. Measures like an F1-score, precision-recall for each class, or macro and micro-averaging across classes would be necessary to determine model performance. Also, techniques such as "one-vs-all" classification or ensemble methods could be assessed to handle any class imbalances that are present in real-world datasets.

4) GPU Transformation

When having to do consistent repetitive calculations like in this simulation, making use of the parallel computation power of a GPU would be significantly beneficial in ensuring the simulation was scalable without the degradation of performance. As it stands the simulation runs smoothly because the dataset is relatively small however, once the dataset grows to thousands of nodes the performance would suffer greatly. Transforming this simulation to use GPU computing would solve this scale problem with ease as GPUs are able to perform millions of calculations every second. To achieve this while still using the R software, utilizing a library such as 'gpur' would allow general-use GPU integration. We could even still utilize the machine learning aspect of the simulation on the GPU by installing the 'tensorflow-gpu' library.

## VI. Your Analysis

Next, we are going to discuss our analysis of the simulation with respect to the graphs generated. The leading two figures below are the visualized results of the Nearest-Neighbor method. The moderate accuracy in predicting intracellular signaling cascade status reflects the simplicity of the nearest-neighbor method and its reliance on strong homophily (tendency to be attracted to like attributes).

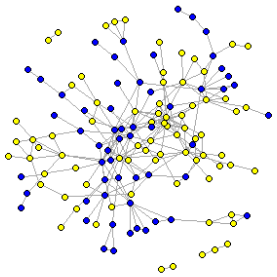

Fig. 1 Network of interactions among proteins known to be responsible for cell communication in yeast. Yellow vertices denote proteins that are known to be involved in intracellular signaling cascades, a specific form of communication in the cell. The remaining proteins are indicated in blue.

In Figure 1, the network visualization shows clusters of yellow vertices (proteins involved in intracellular signaling cascade) surrounded predominantly by other yellow vertices. This supports the assumption of homophily (things seek out others that are similar to themselves), where nodes with similar attributes are likely to cluster.

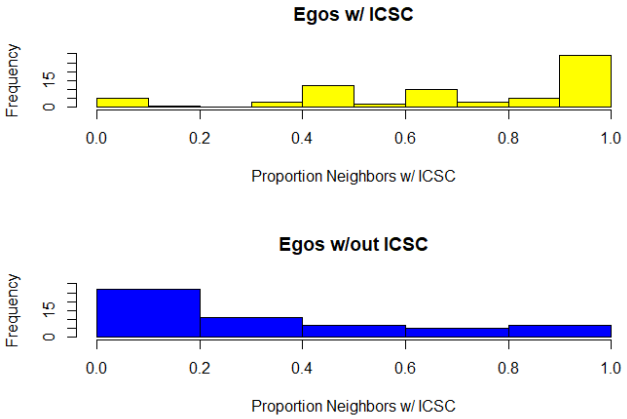


Fig. 2 Histograms of nearest-neighbor averages for the network shown in Fig. 1, separated according to the status of the vertex defining each neighborhood (i.e., 'ego').

However, Figure 2 reveals the limitations of this method. The histograms of nearest-neighbor averages show that while nodes with ICSC=1 tend to

have higher averages, the overlap between the two histograms suggests problems (misclassifications) in areas where the network lacks proper clustering. These problems add to the method's error rate of approximately 25.89%. While computationally simple, the nearest-neighbor approach does not capture the nuanced dependencies present in the network.

With Markov Random Fields, the inclusion of exogenous data significantly enhances its performance. In Figure 3, the weights of the Laplacian kernel are shown to demonstrate the complex dependencies in the network. The addition of gene motifs as the exogenous data reduces the error rate from 20.47% to 18.89% (network-only features). This improvement highlights the value of integrating multi-modal analysis.

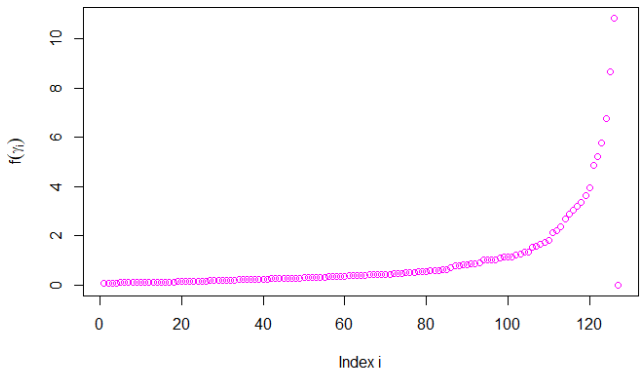


Fig. 3 Plot of the weights f (γi) defining the Laplacian kernel K = L− i.

Also, Markov Random Fields captures probabilistic relationships between neighboring nodes, which are overlooked by the nearest-neighbor methods. This improvement highlights the advantage of combining the network topology with "domain-specific" node attributes. By considering both the endogenous (network structure) and exogenous (gene motifs) factors, Markov Random Fields provides a more adaptable and accurate prediction framework.

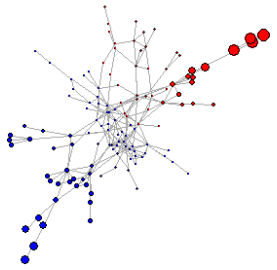

Fig. 4 Visual representation of the eigenvectors φi for the protein interaction network ppi.CC.gc. Negative values are shown in blue, and positive values, in red, with the area of each vertex proportional to the magnitude of its entry in the corresponding eigenvector.

Kernel regression methods outperform both nearest-neighbor and Markov Random Fields models, which show their ability to capture complex dependencies in the network. The spectral methods use the "eigenvectors" and "eigenvalues" of the Laplacian matrix, as shown in Figure 4, where eigenvectors are represented with nodes colored according to their positive or negative values. This representation allows kernel regression to use global structural information from the network.

This combined kernel approach, which integrates the Laplacian kernel with exogenous features (like gene "motifs": a short recurring pattern in DNA or protein), achieves the lowest error rate of 6.29%. This shows the power of spectral methods to generalize better than traditional approaches (work best in networks with diverse structures). Kernel regression's flexibility in non-linear relationships further strengthens its applicability in complex (biological) systems.

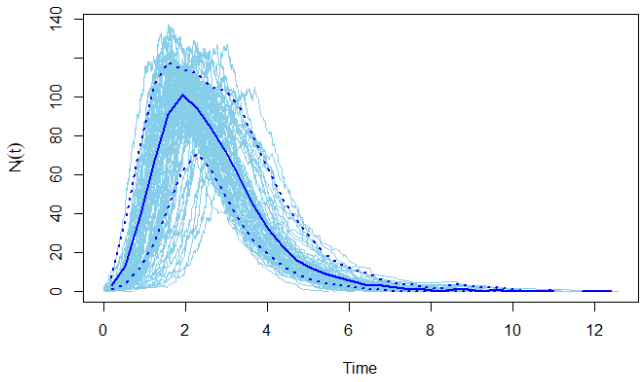


Fig. 5 Realizations of the number of infectives NI(t)for the network-based SIR process simulated on an Erdös-Rényi random graph.

Lastly, network topology plays a critical role in shaping epidemic dynamics. In Figure 5, we see the SIR process on an Erdös-Rényi graph which shows a rapid initial rise in infections due to the uniform probability of edge formation.

In contrast, Figure 6 shows the influence of hub nodes, which have faster and larger infection peaks, as shown through a Barabási-Albert graph. We also see in Figure 7 how "small-world" properties can lead to slower spread because of localized clustering.

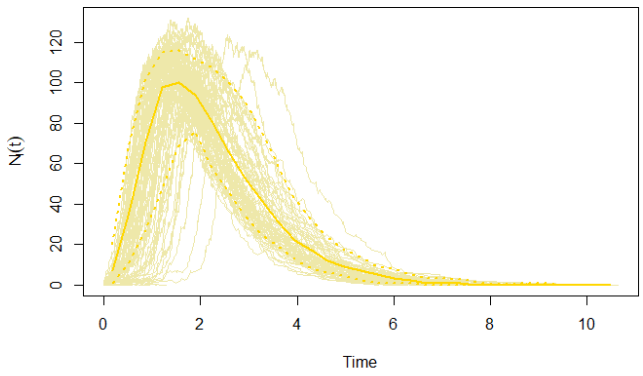


Fig. 6 A Barabási-Albert random graph.

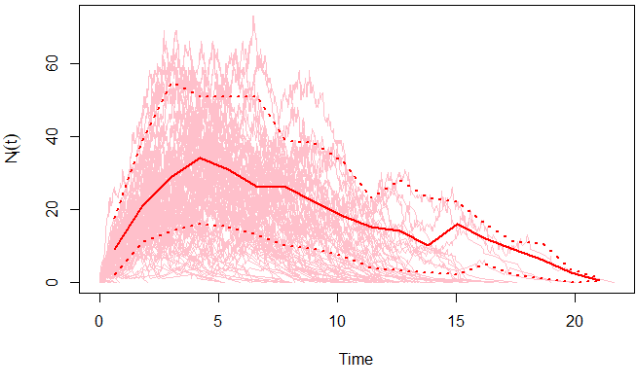


Fig. 7 A WattsStrogatz 'small-world' random graph. Darker curves indicate the median (solid) and the 10th and 90th percentile (dotted), over a total of 100 epidemics (shown in light curves).

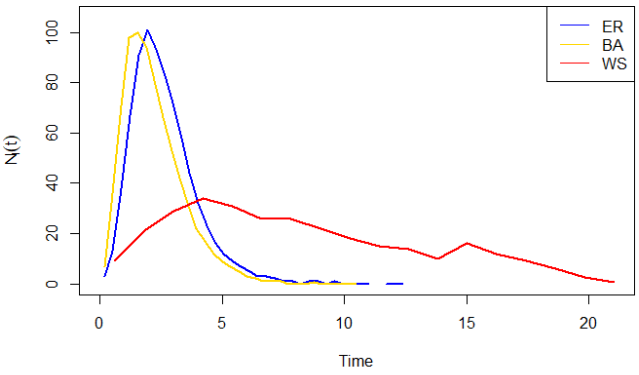


Fig. 8 The three median functions are compared.

By combining all the graphs in Figure 8, we can make a comparative analysis. This analysis shows that the Erdös-Rényi graph has the highest infection peaks (at least, this is what came out of our own running of the code), while the Watts-Strogatz graph contains the spread more effectively (through the balance of local clustering and the occasional "long-range" connections). This emphasizes the importance of understanding the network topology when designing proper intervention strategies for real-world epidemics.

## VII. Limitations of the Project

We found that there were quite a few limitations of the project and the R code that we used during the

duration of said project. Those limitations are as follows below:

Static Network Assumptions: One major limitation of this project is the use of static network models, where the structure and attributes remain constant throughout the analysis. Real-world networks, especially biological and social systems, are dynamic. Connections between nodes may change over time due to evolving conditions. For example, protein interactions can vary based on cellular states, and social networks can shift due to changing relationships. This static assumption limits the applicability of results to scenarios where networks exhibit significant change over time.

Binary Attribute Restriction: The models implemented in this project are limited to binary classification, such as determining whether a protein is involved in the intracellular signaling cascade or not. However, many real-world problems involve multi-class attributes, where nodes can belong to multiple categories or roles. For instance, proteins may have overlapping or hierarchical functions, and individuals in social networks may simultaneously exhibit different behaviors. The lack of multi-class capabilities restricts the versatility of applications and the depth of insight.

Computational Demands: The methods used, particularly Markov Random Fields and kernel regression, require significant computational resources as the network size increases. While the protein-protein interaction dataset used in this project is relatively small, scaling these methods to larger networks could lead to substantial increases in computation time and memory requirements. The inclusion of more complex kernels or Bayesian frameworks, as proposed, would exacerbate these demands without access to advanced hardware or optimized algorithms. However, making use of transforming these algorithmic processes to a GPU, as previously discussed in the “PROVISIONS THAT COULD BE ADDED” section, instead of utilizing the CPU would solve or at least lessen the burden of the computational demands.

Simplistic Epidemic Modeling: The SIR model used to simulate epidemic dynamics is a basic compartmental model that assumes fixed rates for infection and recovery. Real-world epidemics involve more complex factors, such as varying transmission rates, heterogeneous susceptibility, reinfections, and external interventions. Moreover, the SIR model assumes static network topology, while real epidemics often involve changing interaction patterns (e.g., lockdowns or travel restrictions). This limits the applicability of the results to scenarios with more nuanced dynamics.

Data Quality and Completeness: The protein-protein interaction data used in this project, like many real-world datasets, is prone to noise, incompleteness, and experimental errors. Misclassified or missing interactions can negatively impact model performance and result in inaccurate predictions. For example, the MRF model may incorrectly estimate dependencies if the network structure is incomplete or inaccurately annotated.

Interpretability of Advanced Models: While kernel regression and spectral methods provide superior accuracy, their results are often less interpretable compared to simpler models like nearest-neighbor or MRF. This trade-off between accuracy and interpretability can pose challenges when results need to be communicated to non-technical stakeholders or used to guide experimental validation.

## VIII. Conclusion

This project demonstrates the use of advanced statistical and computational methods (advanced for us anyway) for analyzing/predicting processes on network graphs. By applying nearest-neighbor methods, Markov Random Fields, and kernel regression to a protein-protein interaction network, we explored each approach's strengths and limitations. Adding exogenous data and spectral methods significantly enhanced prediction accuracy, with kernel regression achieving the lowest error rate of 6.29%. These results show the importance of

integrating network topology and domain-specific node attributes for adaptable predictions.

Dynamic processes, like epidemic modeling, show network topology's key role in shaping outcomes. Simulations using the SIR model on Erdős-Rényi, Barabási-Albert, and Watts-Strogatz networks revealed the distinct infection dynamics, with hub-dominated networks having rapid infections and small-world networks mostly containing the spread effectively. These findings show the practical implications for designing interventions in a real-world system (for example, having targeted vaccinations or social distancing).

Despite what we completed/discovered during the project, there were many limitations, including reliance on static networks, binary classification tasks, and basic epidemic modeling. Future work/projects/research papers should address those limitations (by incorporating things such as dynamic graph models, multi-class predictions, or more sophisticated epidemic simulations). The predictions could also be improved by having Bayesian Markov Random Fields and hierarchical kernel regression to quantify uncertainty can capture latent structures.

In conclusion, this project attempts to bridge theoretical network practices and practical applications in order to showcase the potential of network-based modeling in important fields like biology, epidemiology, and the social sciences. By addressing the current limitations of our project (and related R code) and extending the methodologies, future work/research can discover better insights into complex systems and really contribute to solving meaningful real-world problems/challenges.

## References


[1] "Structure your paper (IEEE Format)," IEEE Author Center Conferences, https://conferences.ieeeauthorcenter.ieee.org/write-your-paper/structure-your-paper/#:~:text=Conclusion,suggest%20future%20areas%20for%20research. (accessed Nov. 6, 2024).

[2] J. Prosise, "Multiclass classification with Neural Networks," Atmosera, https://www.atmosera.com/blog/multiclass-classification-with-neural-networks/ (accessed Nov. 12, 2024).

[3] Author links open overlay panelF. Harary and AbstractResearch in graph theory has focused on studying the structure of graphs with the assumption that they are static. However, "Dynamic Graph Models," Mathematical and Computer Modelling, https://www.sciencedirect.com/science/article/pii/S0895717797000502 (accessed Nov. 8, 2024).

[4] Markov random fields and Markov Logic Networks, https://www.cs.rochester.edu/~schubert/444/lecture-notes/markov-logic-networks.pdf (accessed Nov. 8, 2024).

[5] E. D. Kolaczyk and G. Csárdi, *Statistical Analysis of Network Data with R*. Cham, Switzerland: Springer, 2020.

[6] D. P. Agrawal and Q.-A. Zeng, *Introduction to Wireless and Mobile Systems*. Boston, MA: Cengage Learning, 2016.

[7] "Tensorflow," TensorFlow, https://www.tensorflow.org/ (accessed Nov. 28, 2024).

[8] "Difference between barabási-albert model and Erdos-Renyi model," Stack Overflow, https://stackoverflow.com/questions/63951391/difference-between-barab%C3%A1si-albert-model-and-erdos-renyi-model (accessed Nov. 16, 2024).

[9] "Network science by Albert-László Barabási," BarabásiLab, https://networksciencebook.com/chapter/5 (accessed Nov. 17, 2024).

[10] Lecture 8:motifs and motifs finding, https://www.ncbi.nlm.nih.gov/CBBresearch/Pr

zytycka/download/lectures/PCB_Lect08_Bind_Motifs.pdf (accessed Nov. 28, 2024).

[11] C.-L. Hsu, “Watts-Strogatz model of small-worlds,” An Explorer of Things, https://chih-ling-hsu.github.io/2020/05/15/watts-strogatz (accessed Nov. 16, 2024).

[12] GeeksforGeeks, “Erdos renyl model (for generating random graphs),” GeeksforGeeks, https://www.geeksforgeeks.org/erdos-renyl-model-generating-random-graphs/ (accessed Nov. 16, 2024).

[13] Markov random fields, http://www.cs.toronto.edu/~fleet/courses/2503/fall11/Handouts/mrf.pdf (accessed Nov. 16, 2024).

[14] Ph. D. Niranjan Pramanik, “Kernel regression-with example and code,” Medium, https://towardsdatascience.com/kernel-regression-made-easy-to-understand-86caf2d2b844 (accessed Nov. 9, 2024).

[15] A. Velimirovic, “What is GPU computing? {benefits, use cases, limitations},” phoenixNAP Blog, https://phoenixnap.com/blog/what-is-gpu-computing (accessed Nov. 28, 2024).