Leveraging Graph Machine Learning for Moonlighting Protein Prediction: A Local PPI Network and Physiochemical Feature Approach

H Zhou and R Sarkar

School of Informatics, University of Edinburgh, 57 George Square, Edinburgh, EH8 9JU, UK

varianzhou@gmail.com, rsarkar@inf.ed.ac.uk

Abstract. Moonlighting proteins (MPs) play critical roles in cellular functions and disease, yet their complex nature challenges traditional identification methods. This research delves into the potential of Graph Machine Learning (GML) to effectively utilize Protein-Protein Interaction (PPI) networks and physicochemical properties for MP prediction. Our study focuses on two GML architectures: Graph Attention Networks (GAT) and Graph Convolutional Networks (GCN), which are augmented by integrating quasi-sequence-order (QSOrder) and amino acid composition (AAC) to enrich the feature representation. We evaluated these models on a dataset comprising 310 proteins, finding that GML substantially outperforms conventional methodologies such as Support Vector Machines (SVM), K-nearest neighbors (KNN), and Random Forest (RF), especially in scenarios with limited data where Deep Neural Networks (DNNs) typically struggle. Notably, the GAT model, when enhanced with QSOrder and PPI data, exhibited exceptional accuracy, and robustness in precision, F1-score, and ROC-AUC metrics. This indicates a significant edge in capturing the complex predictive signals embedded in PPI networks. Our findings strongly support the integration of GML in MP research and suggest a novel approach for exploring MPs through their interactive partners.

Keywords: Graph Machine Learning, Moonlighting Proteins, Protein-protein Interaction networks, Physiochemical Features, Graph Attention Networks, Graph Convolutional Networks.

1. Introduction

Moonlighting proteins (MPs), with their multifaceted biological roles, have redefined our understanding of the proteome, challenging the traditional one gene-one enzyme paradigm (Jeffery, 1999). Their significance in cellular functions and potential as therapeutic targets in diseases has garnered much attention (Copley, 2003; Huberts & van der Klei, 2010). Despite this, identifying MPs experimentally remains a daunting task, limited by the scale and scope of experimental methods.

The emergence of computational biology has provided promising alternatives for MP identification. Tools such as MPFit and DextMP have achieved remarkable results with Gene Ontology and text mining (Khan, McGraw et al., 2017; Khan, Bhuiyan et al., 2017). However, these approaches grapple with scalability issues, predominantly due to the labour-intensive nature of protein annotation and the disproportionately rapid expansion of literature and protein databases.

In response, recent research has pivoted toward exploiting physicochemical properties, leveraging attributes such as quasi-sequence order (QSOrder) and amino acid composition (AAC) to predict MPs with notable success (Shirafkan et al., 2021; Hu & Du, 2023; Chen et al., 2008). Interestingly, in this realm, it has been observed that traditional machine learning methods, like Support Vector Machine (SVM) (Hearst et al., 1998), K-nearest-neighbour (KNN) (Peterson, 2009), and Random Forest (RF) (Breiman, 2001), surprisingly outperform Deep Neural Networks (DNNs) (LeCun et al., 2015). This counterintuitive finding suggests that the dimensionality and specific nature of the feature sets used to represent MPs may be more amenable to these traditional algorithms, which tend to be more robust with smaller, feature-rich datasets.

Parallel to this, the utility of protein-protein interaction (PPI) data has been brought to light, emphasizing the value of network-based approaches (Gomez et al., 2011; Khan & Kihara, 2016). Nonetheless, the direct use of PPI networks in traditional machine learning has fallen short, with only modest performance (Khan & Kihara, 2016).

The advent of Graph Machine Learning (GML) (Xu et al., 2018) marked a significant turn in bioinformatics, particularly in capturing complex patterns within Protein-Protein Interaction (PPI) networks, making it a promising tool for predicting moonlighting proteins (MPs). Our study evaluates two prominent GML architectures, Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT)(Kipf et al., 2016; Veličković et al., 2018), using a benchmark dataset comprising 310 proteins, split into 174 MPs and 136 non-MPs. We compared them against previous methods, including Support Vector Machines (SVM), K-nearest Neighbours (KNN), Random Forest (RF), and Deep Neural Networks (DNN).

In framing MP prediction as a graph classification task (Tsuda et al., 2010), our results demonstrate that GAT and GCN models not only show robust performance across essential metrics—accuracy, precision, F1 score, and ROC-AUC—but also surpass most traditional methodologies. These models adeptly handle smaller datasets that often impede the effectiveness of DNNs. Our research thus underscores the capabilities of GML in predicting MPs and the pivotal role of PPI network analysis in unravelling the complex nature of MPs.

2. Data & Materials

2.1 Dataset

This study utilized a benchmark dataset consistent with prior research (Shirafkan et al., 2021; Hu and Du, 2023; Chen et al., 2022), containing 351 proteins—215 of which are moonlighting proteins (MPs) and 136 non-moonlighting. These proteins represent a variety of organisms, including humans, E. coli, yeast, rats, fruit flies, and thale cress, among others. The MP data were obtained from the MoonProt database, a comprehensive resource for experimentally verified MPs (Chen, C. et al, 2021). To minimize redundancy and ensure dataset quality, proteins with more than 40% sequence similarity were filtered out using CD-hit (Fu et al., 2012). Non-MPs were sourced from the MPFit dataset (Khan, McGraw et al., 2017), selected through rigorous criteria.

For the Protein-protein interaction (PPI) data, we used the STRING database (Szklarczyk et al., 2021), which includes a spectrum of both experimentally obtained and computationally predicted interactions. These interactions, which may be functional (e.g., participating in the same metabolic pathway) or physical (e.g., direct contact between proteins), are treated equivalently in our analysis provided they surpass a confidence threshold of 0.7, which is predefined to ensure the quality of data. Out of the 215 MPs listed by Shirafkan et al. (2021), 174 were present in STRING, with all non-MPs also being successfully matched. Our final dataset, therefore, included 310 proteins, encompassing 174 MPs and 136 non-MPs. We obtained PPI networks of varying sizes for each protein, from their 20 closest neighbours up to 100, to ensure comprehensive analysis.

2.2 Problem Definition

This study frames the prediction of moonlighting proteins (MPs) as a graph classification task (Tsuda et al., 2010). The task is formally defined as **T = (G, y)**, where **G** represents a collection of graphs, and **y** is a binary vector of labels. Each graph g_i in **G** corresponds to a target protein's Protein-Protein Interaction (PPI) network, and the corresponding label $y_i \in \{0,1\}$ in **y** denotes the moonlighting status of these proteins, with 1 indicating an MP and 0 indicating a non-MP.

For each target protein, a graph **G = (X, E)** is constructed. Here, **E** is the weighted adjacency matrix that encapsulates the PPI network, with weights reflecting the strength or confidence of each interaction. **X** is the feature matrix, which encodes attributes of the nodes, such as physiochemical features, or other relevant characteristics that may influence a protein's moonlighting status. Accordingly, the set **G** includes all graphs derived from the target proteins using this approach, while **y** contains their associated classification labels.

2.3 Graph Machine Learning (GML) models

Two prevalent GML methods were tested: Graph Convolutional Networks (GCN), and Graph Attention Networks (GAT) (Kipf et al., 2016; Veličković et al., 2018). This section will briefly review their architectures.

2.3.1 Graph Convolutional Networks (GCN)

The Graph Convolutional Network (GCN), as proposed by Kipf et al. (2016), heralded a transformative approach to deep learning by adapting neural network architectures for graph-structured data. Unlike traditional neural networks that do not inherently account for relational data, GCNs excel at capturing the relationships between nodes, rendering them highly effective for graph-related tasks. A key feature of GCNs is their ability to efficiently propagate information through the network's graph structure.

Mathematically, the information propagation in a GCN is performed layer by layer as follows:

$$\boldsymbol{H}^{(l+1)} = \sigma\left(\widetilde{\boldsymbol{D}}^{-1/2}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-1/2}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}\right)$$

In this formula, $\boldsymbol{H}^{(l)}$ represents the (hidden) embedding at layer l, $\widetilde{\boldsymbol{A}}$ is the adjacency matrix with self-loops to capture nodes' self-information. $\widetilde{\boldsymbol{D}}$ is the weight matrix of $\widetilde{\boldsymbol{A}}$, and $\boldsymbol{W}^{(l)}$is the weight matrix associated with the l-th layer. σ is some non-linear activation function. To generate a holistic representation of the graph, GCN aggregates the node embeddings using a readout function—such as mean or max pooling—resulting in a comprehensive graph embedding. This embedding can then be used for downstream tasks like graph classification.

2.3.2 Graph Attention Networks (GAT)

The Graph Attention Networks (GAT) introduced by Veličković et al. (2018) represent a significant advancement in graph neural networks by incorporating attention mechanisms. These mechanisms enable nodes to weigh their neighbors' contributions adaptively, allowing for a dynamic aggregation process. Specifically, GAT calculates attention coefficients, denoted as α_{ij}, which quantify the relevance of one node's features to another, facilitating the selective emphasis on important neighbor connections.

Notably, the multi-channel architecture of GAT permits the parallel computation of these coefficients, enhancing the network's ability to discern long-range dependencies within graph data. Such capabilities have led to GAT achieving state-of-the-art results in tasks as diverse as node classification and graph classification, underscoring its adaptability and strength in leveraging complex relational data.

The update function in GAT is:

$$\boldsymbol{h}_i' = \sum_{j \in N(i)} \alpha_{ij} \boldsymbol{W} \boldsymbol{h}_j$$

Where $\boldsymbol{h_i}$ is the feature vector of node *i*, **N(*i*)** denotes the neighbours of node *i*, **W** is a weight matrix, and α_{ij} is the attention coefficient computed as $\alpha_{ij} = softmax_j(\boldsymbol{a^T}[\boldsymbol{Wh_i}||\boldsymbol{Wh_j}])$, where || is the concatenation operator.

2.4 Feature Selection

Following previous works (Shirafkan et al., 2021; Chen et al., 2022), we primarily focus on QSOrder and AAC (Chen et al., 2008), which showcased competent performances on traditional ML methods.

2.4.1 QSOrder

The quasi-sequence-order (QSOrder) descriptor, denoted as *x(k)*, is defined based on the normalized occurrence of amino acid type *k* (where *k* ranges from 1 to 50). The formula for *x(k)* is given by:

$$x(k) = \begin{cases} \frac{f_k}{\sum_{i=1}^{20} f_i + \omega \sum_{d=1}^{30} \tau_d} & if\ 1 \leq k \leq 20 \\ \frac{w\tau_{k-20}}{\sum_{i=1}^{20} f_i + \omega \sum_{d=1}^{30} \tau_d} & if\ 21 \leq k \leq 50 \end{cases}$$

In this formula, $f_k$is the frequency of amino acid type *k*. The weighting factor ω is given as 0.1 in our setting. This descriptor provides a combination of sequence composition and sequence order information, with the total number of descriptor values for QSOrder being 100. τ_d is the Sequence-order coupling number, for which details can be found in Chen et al., 2008.

2.4.2 AAC

Amino Acid Composition (AAC) refers to the proportion of each amino acid in a protein sequence. Specifically, the AAC for a given amino acid *i* is calculated as:

$$x_i = \frac{n_i}{\sum_{j=1}^{20} n_j}$$

where n_i represents the number of occurrences of amino acid *i*, and the denominator sums the occurrences of all 20 standard amino acids in the sequence. AAC provides a normalized representation of the frequency of each amino acid in the protein.

3 Experiments

In our experiments, we assessed the performance of Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) on a curated dataset, which underwent preprocessing steps including normalization and feature selection. QSOrder and AAC and their combination were applied to investigate their impact on model performance. These GML models were compared against traditional machine learning algorithms—Support Vector Machines (SVM), Random Forest (RF), K-nearest Neighbors (KNN), and Deep Neural Networks (DNN). To promote reproducibility and facilitate further research, we have made the source code publicly accessible[1]. The efficacy of each classification model was gauged using precision, accuracy, F1 score, and ROC-AUC, chosen for their robustness in evaluating the predictive power and reliability of classification models.

3.1 Predicting MPs Using Only QSOrder Number

First, we downloaded the PPI network of all proteins with their 30 closest neighbours, then we utilized the QSOrder vector of each protein within the PPI networks as the exclusive feature for the feature matrix. This approach was inspired by the methodology of Shirafkan et al., 2021, and we used the R-package ftrCOOL (Amerifar et al. 2022) to facilitate feature extraction.

[1] https://github.com/VarianZhou/MPGNN.git

While deriving the QSOrder feature from proteins in the networks, there are 11 proteins bearing unidentified amino acids, labeled as 'X'. Such occurrences are often attributed to instrumentation or experimental limitations. A closer analysis revealed this data loss is insignificant, so we proceed the experiment with these ambiguous segments excluded.

In the initial phase of our study, we applied the GCN and GAT models to our dataset, followed by traditional methods—SVM, KNN, and RF—equipped only with QSOrder features for predicting MPs in 310 proteins. We subsequently evaluated a DNN model configured with two layers, each comprising 256 neurons.

To address challenges posed by the limited dataset size, we adopted a comprehensive strategy to enhance model robustness and mitigate performance variability:

Data Preparation: The dataset was randomized to prevent model bias, setting aside a 15% subset as an independent test set. The remaining data underwent 10-fold cross-validation, providing a robust assessment of model reliability and safeguarding against overfitting.
Class Balance: Stratified sampling ensured equal representation of MPs and non-MPs, crucial for maintaining class balance. We further stratified graphs by connectivity—very sparse (< 0.25), sparse (0.25–0.5), dense (0.5–0.75), and very dense (0.75–1.0)—to preserve consistency in class and connectivity distributions across folds.
Model Evaluation: To counteract variability across folds, we instituted a thresholded ensemble approach. Models achieving a validation accuracy above 0.7 were included in an ensemble, applying a soft voting mechanism on the test set. This selective ensemble method integrated only the most reliable models, enhancing predictive reliability.

For the GCN and GAT models, EarlyStopping was implemented to monitor validation loss, halting training after 25 epochs without improvement. The learning rate was dynamically adjusted, halved after 10 epochs lacking performance gains, starting with an initial rate of 0.001. We capped the training iterations at 500, utilizing the Adam optimizer (Kingma et al. 2014). Binary cross-entropy served as the loss function, fitting for our binary classification task. Edge information was encoded in a binary edge matrix, denoting interaction presence. Our models, crafted using *PyTorch-Geometric* (Fey et al. 2019), adopted a three-layer structure with 256 nodes each, incorporating a 0.3 dropout rate to deter overfitting. To prevent oversmoothing (Chen, Lin et al. (2020)), we limited our architecture to three layers, employing a max pooling readout function for graph-level output, which then passed through a linear layer with an argmax function for classification.

Baseline models were developed using *scikit-learn*'s default settings (Pedregosa et al., 2011), providing a standard for comparison. The DNN was similarly constructed with scikit-learn and included EarlyStopping as a regularization measure.

Our findings are promising. Table 1 illustrates that the GAT model stands out as the top performer, achieving the highest accuracy at 74.9%. It also ranks second in both precision and ROC-AUC, with scores of 75.4% and 72.8%, respectively. The GAT model's F1 score is competitive at 74.5%, narrowly behind traditional machine learning methods such as RF (76.8%), SVM (77.7%), and KNN (78.9%). Similarly, the GCN model shows robust performance with an accuracy of 74.3%, slightly edging out the SVM's 74.2%. Its F1 score (74.0%), precision (74.4%), and ROC-AUC (72.2%) are also notable, showcasing its ability to rival, and potentially surpass, conventional methods. In line with previous research, the DNN model exhibits the least impressive performance across almost all metrics, which is markedly lower than its counterparts, likely due to the constraints of the limited dataset size.

3.2 Incorporation of AAC

Upon integrating Amino Acid Composition (AAC) as a feature—a feature shownd to be competitive in Hu et al. (2023)—all models were reevaluated. Table 1 shows that while the GAT model retained a competitive edge with an accuracy of 73.8%, F1 score of 73.6%, precision of 74.3%, and ROC-AUC of 72.1%, these figures reflect a decrease compared to its performance with QSOrder features.

Conversely, the GCN model saw performance gains from AAC integration, achieving the highest accuracy at 74.5% and the second-highest precision at 74.6%. Its F1 score and ROC-AUC also remained strong at 74.1% and 72.3%, respectively.

The SVM model emerged as a standout with AAC, achieving the highest F1 score of 77.8%, precision of 76.5%, and ROC-AUC of 73.8%, while coming in a close second in accuracy at 74.4%. These results suggest a particular synergy between AAC features and the SVM algorithm. Similarly, the KNN model showed slight improvements, further indicating the potential alignment of AAC with these traditional algorithms. In contrast, the RF and DNN models experienced a downturn in performance, dropping by 0.5-2% across all metrics, suggesting a less effective integration of the AAC feature with these methods.

Model	Feature	ROC-AUC	Precision	F1-score	Accuracy
GAT	QSOrder	72.8	75.4	74.5	**74.9**
GCN	QSOrder	72.2	74.4	74.0	74.3
KNN	QSOrder	70.7	71.0	**78.9**	73.1
SVM	QSOrder	**73.5**	**76.1**	77.7	74.2
RF	QSOrder	72.2	74.9	76.8	73.0
DNN	QSOrder	69.5	72.9	74.4	70.3
GAT	AAC	72.1	74.3	73.6	73.8
GCN	AAC	72.3	74.6	74.1	**74.5**
KNN	AAC	71.1	71.5	**79.3**	73.6
SVM	AAC	**73.8**	**76.5**	77.8	74.4
RF	AAC	70.9	73.4	76.3	71.9
DNN	AAC	68.6	72.1	74.0	69.8
GAT	Combined	**73.5**	75.5	74.5	**74.6**
GCN	Combined	72.4	74.9	74.2	**74.5**
KNN	Combined	70.8	71.0	**78.9**	73.1
SVM	Combined	**73.7**	**76.3**	77.8	74.4
RF	Combined	72.4	75.3	76.9	73.1
DNN	Combined	70.2	73.2	75.2	71.0

*Table 1 The experimental results of GAT, GCN, KNN, SVM, and RF models on four metrics, ROC-AUC, precision, f1-score, and accuracy, when combined with features QSOrder and AAC. For illustration, the top 3 models of each metric are given in bold format. The **Feature** colume provide the type of features used as the node/sample features: QSOrder: quasi-sequence order (see section 2.4.1); AAC: amino acid composition (see section 2.4.2); Combined: AAC + QSOrder.*

3.3 Combination of AAC and QSOrder

To improve the predictive accuracy of our models, we investigated the effects of combining quasi sequence-order (QSOrder) and Amino Acid Composition (AAC) features through concatenation. Table 1 reports the outcomes, revealing that for GAT and GCN models, this integrative strategy generally led to modest performance enhancements across multiple metrics. Contrary to expectations, however, the GAT model's accuracy decreased slightly by 0.3% with the addition of AAC, and similarly, the SVM's ROC-AUC and Precision metrics fell by 0.2%. While these declines are statistically insignificant, they introduce an important caveat: the efficacy of combining physicochemical properties is not guaranteed and requires careful consideration. It suggests that the interplay between different feature types can be complex, and their collective influence on model performance is not always straightforward.

The other models, detailed in the third section of Table 1, displayed small but uniform improvements upon utilizing the combined feature set, lending support to the idea that a careful, strategic combination of features is integral to optimizing computational tools for MP prediction. These insights point to the potential of a nuanced approach to feature selection, highlighting the need for a strategic feature synergy in enhancing the performance of models.

3.4 Graph Size and Model Performance

We also investigated the relationship between the size of each query protein's interaction graph and the accuracy of our predictions. Specifically, we varied the sizes of the networks sourced from the STRING database, testing limits of 20, 30, 50, and 100 nodes. We found that graphs with around 30 nodes yielded optimal performance; increasing the size beyond this point did not lead to significant improvements and, in some instances, even resulted in lower performance. Therefore, for consistency and efficiency in our experiments, we standardized on a graph size of 30 nodes, with a minimum threshold of 24 nodes. With these parameters, 85% of the 310 proteins are finally used in the training and testing of GML models.

4 Discussion

Our study validates the efficacy of Graph Machine Learning (GML) in predicting moonlighting proteins (MPs), with Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) demonstrating significant promise. These GML models, even in their rudimentary forms, pave the way for the development of advanced, domain-specific models that could yield deeper insights into MP behaviour.

Deep Neural Networks (DNNs) have previously shown limitations in predicting moonlighting proteins (MPs), likely due to their reliance on extensive datasets—a challenge highlighted both in this study and in the work of Shirafkan et al. (2021), which reported a maximum accuracy of 71%. In contrast, Graph Machine Learning (GML) models utilize the relational data intrinsic to Protein-protein Interaction (PPI) networks. This approach not only facilitates more efficient parameter tuning but also mitigates the risk of overfitting, making GML models more effective given the limited MP data.

The empirical success of the GML model corroborates the hypothesis that PPI networks contain markers of MPs, as suggested by Khan and Kihara (2016). Particularly, GAT's ability to prioritize influential nodes and pathways highlights the importance of interaction patterns in identifying moonlighting proteins. While the mechanisms behind GAT's effectiveness warrant further investigation, this study points towards the potential of exploring the interacting partners of MPs and developing advanced protein embedding techniques to enhance the precision of MP predictions.

Notably, our study revealed discrepancies in model performance from previous research; for instance, our SVM's accuracy was 74.2%, below Shirafkan et al. (2021)'s 77%. We have obtained consistent results on their dataset, which confirm our implementation's accuracy. The variance is likely due to our smaller MP dataset and a novel evaluation approach.

However, there are several limitations in our study. The most prominent one is the limited data size. Due to the limited data resources on STRING, our dataset only includes 174 MPs, covers only 81% of the 215 MPs in the previous benchmark dataset. In further work, these missed data can be imputed by alternative methods like computational PPI predictions or collected from other PPI databases.

Moreover, while focusing on QSOrder and AAC features has proven beneficial, we recognize that incorporating additional features, such as structural characteristics and evolutionary information, could offer more comprehensive insights. This calls for future work to expand the feature set.

Furthermore, while the GML models has proven effective empirically, it beckons a deeper theoretical explanation. Further profound analysis is still needed to provide more insights in the mechanics of their success.

Lastly, in this study, we didn’t look deep into the specific types of interactions, which are potentially informative. For further work, we can delve into the nature of these interactions and involve these details to our model.

With these in mind, the promising results of GML models on MP prediction are just the beginning. Further theoretical and practical refinement of these models, including exploration into other graph classification methods, are needed to enhance our understanding and prediction of MPs.

5 Conclusion

Our investigation has showcased the impressive potential of Graph Machine Learning (GML) for the prediction of moonlighting proteins (MPs). By harnessing physicochemical properties such as quasi-sequence-order (QSOrder) and Amino Acid Composition (AAC) in conjunction with local Protein-Protein Interaction (PPI) network data, GML models, including Graph Attention Networks (GAT) and Graph Convolutional Networks (GCN), have delivered robust performance. These models have not only excelled in accuracy but also have shown superior precision, ROC-AUC, and F1-score metrics, outstripping a wide array of traditional approaches, including Random Forest (RF) and K-nearest Neighbors (KNN). Notably, the GAT model has even surpassed the accuracy of the Support Vector Machine (SVM), which is recognized as the state-of-the-art method. This is particularly significant considering that GML models have also addressed the challenges posed by data scarcity that often limit the efficacy of Deep Neural Networks (DNNs).

Our findings suggest a promising direction for the use of neural networks in MP prediction, offering a new paradigm that could reshape future research in this domain. However, our journey is not complete. The principal challenge is the small size of the existing dataset—a limitation well-known to hinder the efficacy of data-dependent neural networks. To mitigate this, our future endeavours will involve leveraging newly proposed computational PPI prediction methods (Soleymani et al., 2022) alongside diverse PPI datasets to enrich the training and enhance the robustness of our GML models. In addition, this study only used the simple implementation of two GML models and treated the prediction of MPs as a graph classification task. In our next stage, we plan to explore other frameworks, problem settings, and feature extractors, and involve specific interaction types, to develop a more domain-specific and robust model. Furthermore, we intend to probe the theoretical foundations of the GAT model to improve its interpretability. Our goal is to gain a more granular understanding of the mechanisms underlying MP predictions. Such efforts will not only consolidate the practical applicability of our model but also deepen our understanding of the MP and its connection with PPI.

6 Acknowledgements

We extend our sincere gratitude to James Ward for his invaluable suggestions on the manuscript's composition; to Dr. Peihua Liang for offering crucial chemical insights in the initial phases of our research; and to Professor Shlomo Ta'asan, whose workshop on 'Data Science and its Applications' in the summer of 2023 enriched our understanding of the methodologies applied in this study.

References

[1] Kipf T N and Welling M 2016 Semi-supervised classification with graph convolutional networks Preprint arXiv/1609.02907

[2] Xu K, Hu W, Leskovec J, and Jegelka S 2018 How powerful are graph neural networks? Preprint arXiv/1810.00826

[3] Chen C, Chen L X, Zou X Y, and Cai P X 2008 Predicting protein structural class based on multi-features fusion *J. Theor. Biol.* **253(2)** 388-392

[4] Chen Y, Li S, and Guo J 2022 A method for identifying moonlighting proteins based on linear discriminant analysis and bagging-svm *Front. Genet.* **13** 963349

[5] Copley, S. D. (2003). Enzymes with extra talents: moonlighting functions and catalytic promiscuity *Curr. Opin. Chem. Biol.* **7(2)** 265-272

[6] Fey M, and Lenssen J E 2019 Fast graph representation learning with PyTorch Geometric Preprint arXiv/1903.02428

[7] Fu L, Niu B, Zhu Z, Wu S, and Li W 2012 CD-HIT: accelerated for clustering the next-generation sequencing data *Bioinformatics* **28(23)** 3150-3152

[8] Gomez A, Hernandez S, Amela I, Piñol J, Cedano J, and Querol E 2011 Do protein–protein interaction databases identify moonlighting proteins? *Mol. Biosyst.* **7(8)** 2379-2382

[9] Hu J and Du Y 2023 Predicting moonlighting proteins from protein sequence Proc. 16th Int. Joint Conf. Biomedical Engineering Systems and Technologies (BIOSTEC 2023) vol 3 pp 270–275

[10] Huberts D H, and van der Klei I J 2010 Moonlighting proteins: an intriguing mode of multitasking *Biochim. Biophys. Acta Mol. Cell Res.* **1803(4)** 520-525
[11] Jeffery C J 1999 Moonlighting proteins *Trends Biochem. Sci.* **24(1)** 8-11
[12] Khan I, McGraw J, and Kihara D 2017 MPFit: computational tool for predicting moonlighting proteins *Methods Protoc.* **1611** 45-57
[13] Khan I K, and Kihara D 2016 Genome-scale prediction of moonlighting proteins using diverse protein association information *Bioinformatics* **32(15)** 2281-2288
[14] Khan I K, Bhuiyan M, and Kihara D 2017 DextMP: deep dive into text for predicting moonlighting proteins *Bioinformatics* **33(14)** 83-91
[15] Shirafkan F, Gharaghani S, Rahimian K, Sajedi R H, and Zahiri J 2021 Moonlighting protein prediction using physico-chemical and evolutional properties via machine learning methods *BMC bioinformatics* **22(1)** 1-14
[16] Veličković P et al. 2017 Graph attention networks Preprint arXiv/1710.10903
[17] Pedregosa F et al. 2011 Scikit-learn: Machine learning in Python *J. Mach. Learn. Res.* **12** 2825-2830
[18] Hearst M A, Dumais S T, Osuna E, Platt J, and Scholkopf B 1998 Support vector machines *IEEE Intell. Syst. Appl.* **13(4)** 18-28
[19] Breiman L 2001 Random forests *Mach. Learn.* **45** 5-32
[20] Peterson L E 2009 K-nearest neighbor *Scholarpedia* **4(2)** 1883
[21] Chen D, Lin Y, Li W, Li P, Zhou J, and Sun X 2020 Measuring and relieving the over-smoothing problem for graph neural networks from the topological view Proc. AAAI Conf. on artificial intelligence vol 34 No. 04 pp 3438-3445
[22] Kingma D P, and Ba J 2014 Adam: A method for stochastic optimization Preprint arXiv:1412.6980
[23] Chen C et al. 2021 MoonProt 3.0: an update of the moonlighting proteins database *Nucleic acids Res.* **49(D1)** 368-372
[24] Szklarczyk D et al. 2021 The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets *Nucleic acids Res.* **49(D1)** 605-612
[25] Amerifar S, Norouzi M, and Ghandi M 2022 A tool for feature extraction from biological sequences *Brief. Bioinform.* **23(3)** 108
[26] LeCun Y, Bengio Y, and Hinton G 2015 Deep learning *nature* **521(7553)** 436-444
[27] Soleymani F, Paquet E, Viktor H, Michalowski W, and Spinello D 2022 Protein–protein interaction prediction with deep learning: A comprehensive review *Comput. Struct. Biotechnol. J.* **20** 5316-5341
[28] Tsuda K, and Saigo H 2010 Graph classification *Manag. Min. Graph Data* **40** 337-363