Use of Generative AI for Data Augmentation in Protein Protein Interaction Prediction

Directed Study

Department of Computer Science, Western University

Author - Sabyasachi Patajoshi, Supervisor – Dr. Michael Domaratzki
spatajos@uwo.ca

Acknowledgements

I am grateful to Dr. Michael Domaratzki for providing me an opportunity to work on a deep and exciting topic and providing insightful supervision.

The prior courses at computer science department prepared me to conduct high quality research in this study. I am grateful to the faculty members and the administrative team for providing me the support and guidance through out my association.

Sabyasachi Patajoshi

25th April 2023, Mississauga, ON

Table of Contents

Introduction

Proteins are the real players of metabolism, their functions varying from being structural framework of cells, messengers of environment changes, movement machinery to fighting infections. They are also having important roles in cellular growth and differentiation.

The wide spanning functional expertise of Proteins is only possible through its interactions with other proteins. Only a few functions involve a protein acting on its own; structural proteins such a Keratin in hair and nails is one of the instances.

All life processes including metabolism, excretion to stimuli, movement involve several steps often orchestrated by proteins. That is the pattern in all life forms and is therefore fundamental to life. The signal transduction in cellular pathways and photosynthetic reaction chains are additional direct and apparent examples.

Understanding the protein interactions leads to better biological knowledge and eventually of pathological states. That is the reason there is a lot of interest in the scientific community in connecting the dots and reveal the mystery!

Fig 1 is a pictorial depiction of diverse functions of protein in the body. Fig 2 shows the Protein Protein Interaction (PPI) network.

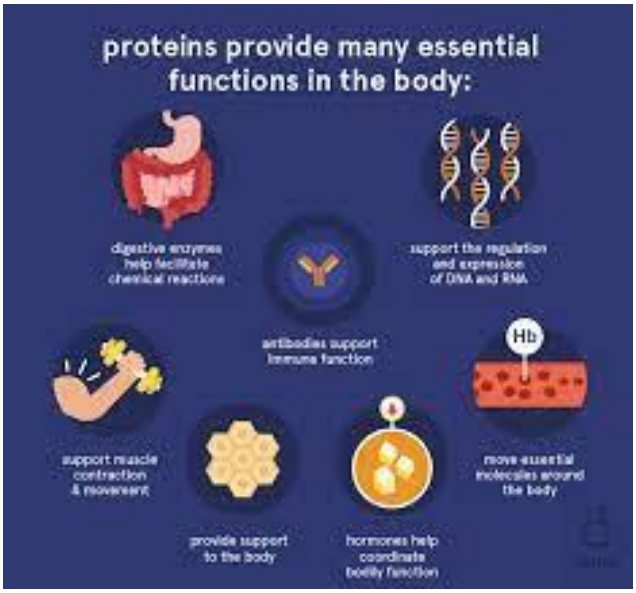

Fig 1 – Examples of Protein functions [6]

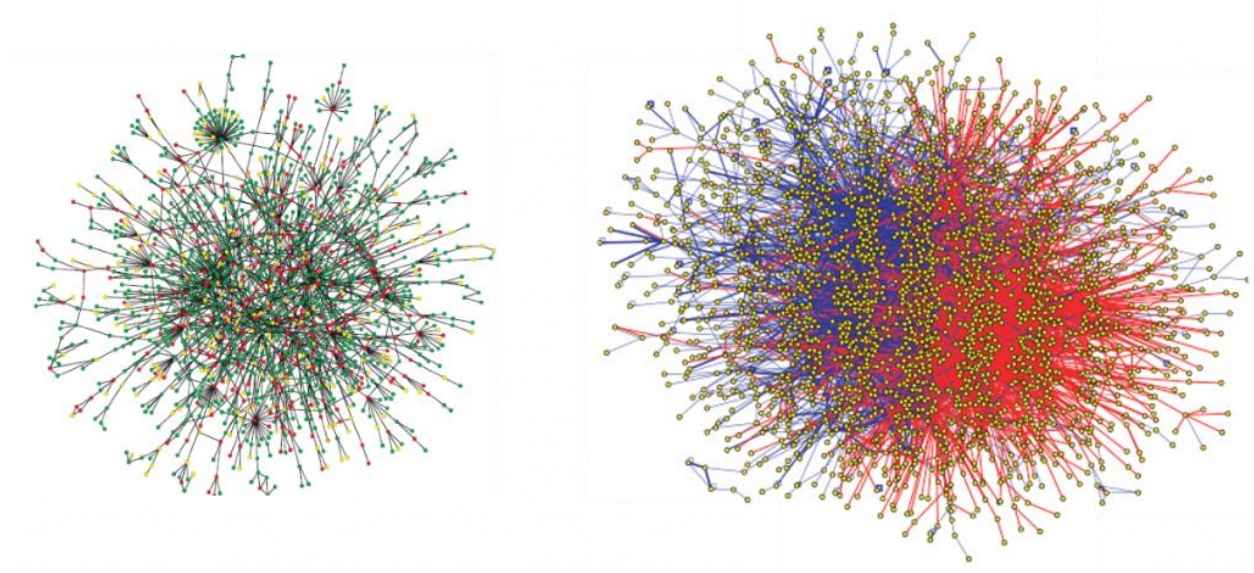

Fig 2 – Example of a PPI Network from EMBL. [7]

Existing Methods

The most popular current state of art methods to determine protein protein interactions include Tandem affinity purification (TAP) Mass spectrometry (MS) and Yeast 2 Hybrid.

TAP uses a agarose bead that attaches to the protein of interest and then can be separated from the cell lysate using centrifugation [3]. In the next step MS is applied to separate, quantify and identify the lysate. Fig 3 depicts TAP MS.

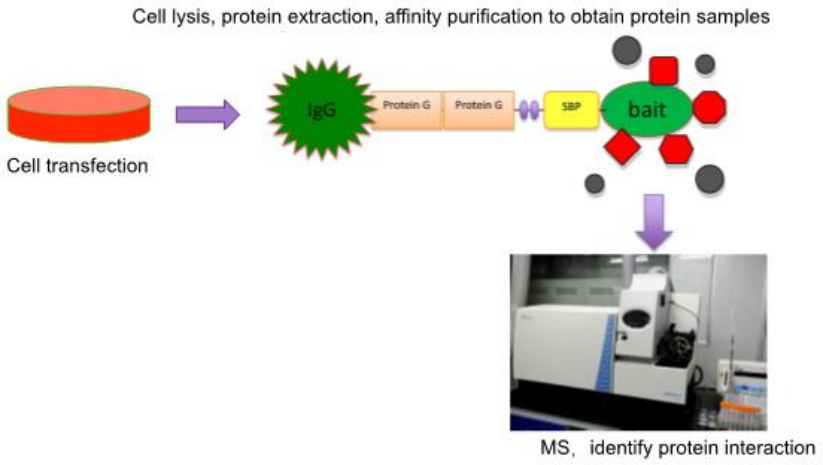

Fig 3 – TAP MS [8]

Y2H in simple terms, a reported gene is expressed when two proteins interact.

The experimental methods detect a high number of false positives and false negatives. Hence there a need for theoretical approaches to detect protein protein interactions.

Several machine learning approaches have been devised that work on the experimental data to learn patterns to identify interactions. These models once trained can be applied to test data to identify novel interactions.

One of the main problems with protein protein interaction is data imbalance discussed in next section in details.

Data Imbalance

High quality biological data is often difficult to obtain as compared to other domains. The experimental methods to detect protein protein interaction and drug target interactions suffer from significant challenges. Usually, a combination of orthogonal experimental data results is combined to produce high quality data. Hence there is a need for theoretical approaches to use the existing data to synthesize additional interaction data that would serve the purpose of data augmentation. These approaches will be more useful in high data scarce situations.

The negative data in protein protein interaction prediction problem is obtained by randomly sampling the space of protein pairs that are not deemed to interact. Proteins present in distant cell compartments, having unrelated motifs or completely unrelated functions are deemed non interacting.

The imbalance may occur at different levels of severity – Mild, Moderate and Extreme. The class that is present in higher number is called Majority and the class present in lower number is called Minority class. Mild imbalance is when the proportion of minority dataset is 20-40% of the whole dataset, moderate it is 1-20% and extreme when it is only 1% of the whole dataset [1][2].

The imbalanced dataset has the potential to cause issues with training as it will not have enough positive samples to learn the decision boundary. There has been a lot of work into dealing with data imbalance that will be discussed in the section. Before trying to apply these methods to the dataset, one can try with out any balancing techniques and evaluate the testing performance, only if that is not acceptable then can start thinking about the techniques presented next.

Data Balancing

In this section we will review several techniques available to dealing with data imbalance.

Synthetic Minority Oversampling Technique (SMOTE) using k-NN is available in popular machine learning package sklearn in python out of the box. This algorithm uses several nearby (k is the hyperparameter) datapoints to synthesize a new data point. Fig 4 depicts this process.

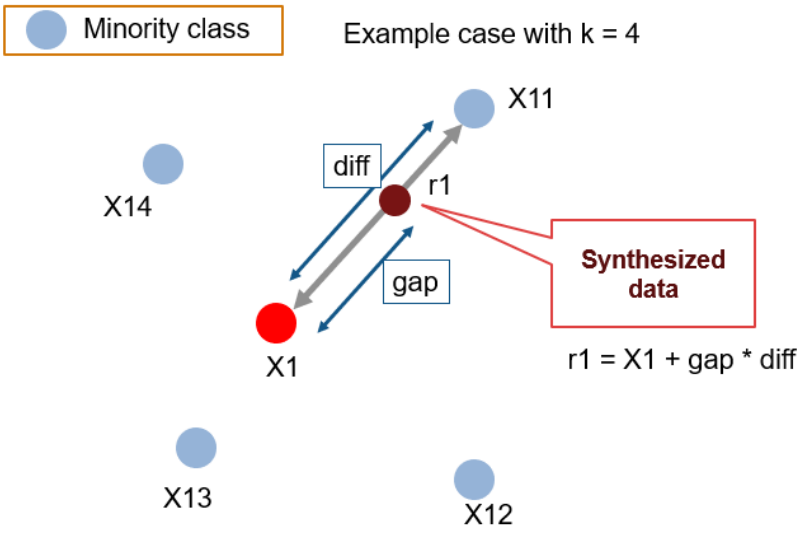

Fig 4 – Technique behind SMOTE -k NN [9]

SMOTE using k-NN can be employed to generate samples of the minority class that then can eventually be used to augment the dataset to help classification.

The next method that I want to highlight is class weighting technique, in the weights passed to the loss function is inversely proportional to their frequency of occurrence.

SMOTE using k-NN technique might serve from limitations from the method that is internally used to calculate the synthetic datapoint, linear techniques may be limited in the quality of the synthetic class and in terms of novelty of feature values calculated from the existing data.

Hence there exists an opportunity for more advanced techniques such as generative models to learn and synthesize from the minority data. In this study I try to employ the Generative Adversarial Network (GAN) algorithm on the protein protein Interaction dataset to synthesize minority positive interaction samples to augment training data and evaluate their impact on classification metrics.

GAN and Adversarial Training

The GAN framework is designed to learn data distribution using adversarial training between two models Discriminator D and Generator G. Generator captures the data distribution and Discriminator if the sample came from data or Generator. This process

can be represented by a two-player minimax game [5]. The mathematical expression is depicted in Fig 5.

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Fig 5 – Analytical expression for the GAN training (Generative Adversarial Nets GoodFellow et. al)

Further Figure 6 provides a sound mathematical analysis of how Generator learns the data distribution guided by the discriminator feedback.

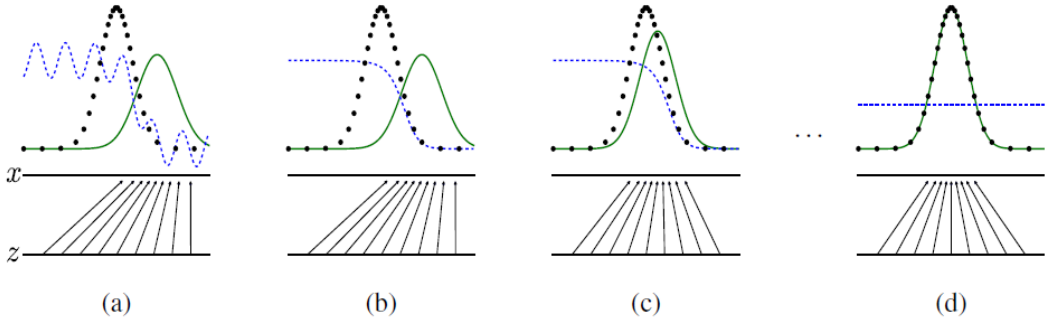

Figure 1: Generative adversarial nets are trained by simultaneously updating the **d**iscriminative distribution (D, blue, dashed line) so that it discriminates between samples from the data generating distribution (black, dotted line) p_x from those of the **g**enerative distribution p_g (G) (green, solid line). The lower horizontal line is the domain from which $\boldsymbol{z}$ is sampled, in this case uniformly. The horizontal line above is part of the domain of $\boldsymbol{x}$. The upward arrows show how the mapping $\boldsymbol{x} = G(\boldsymbol{z})$ imposes the non-uniform distribution p_g on transformed samples. G contracts in regions of high density and expands in regions of low density of p_g. (a) Consider an adversarial pair near convergence: p_g is similar to p_{data} and D is a partially accurate classifier. (b) In the inner loop of the algorithm D is trained to discriminate samples from data, converging to $D^*(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})+p_g(\boldsymbol{x})}$. (c) After an update to G, gradient of D has guided $G(\boldsymbol{z})$ to flow to regions that are more likely to be classified as data. (d) After several steps of training, if G and D have enough capacity, they will reach a point at which both cannot improve because $p_g = p_{\text{data}}$. The discriminator is unable to differentiate between the two distributions, i.e. $D(\boldsymbol{x}) = \frac{1}{2}$.

Fig 6 – GAN training schematic from the original paper (Generative Adversarial Nets GoodFellow et al.)

Generative Adversarial Network Architecture

Fig 6 and 7 depicts the multi layer perceptron design for the generator and discriminator. The generator is input random noise and samples are collected. These samples are sent out to the discriminator along with the real samples to it to be able

identify them as such. The mistakes made by the discriminator is reflected in the cross-entropy loss and is used to generate gradients to train the discriminator network through the backpropagation algorithm.

On the generator side, the loss is calculated on how it was able to fool the discriminator with the fake samples, i.e., how much the discriminator was able to identify the fake as such, the lesser it is the better for the generator. The discriminator fake loss function gets translated into the generator loss function is eventually used to train the generator network.

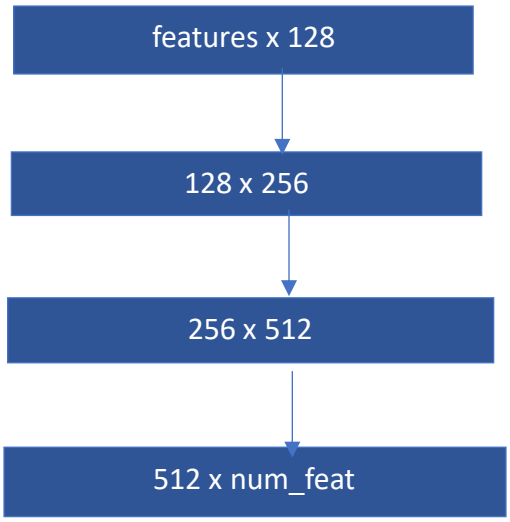

Fig 6 – Generator architecture

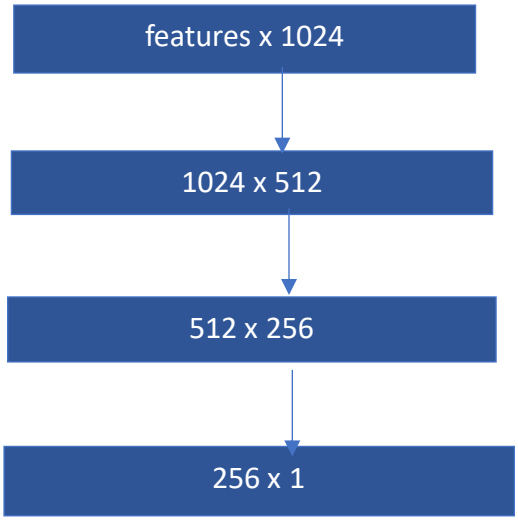

Fig 7 – Discriminator architecture

Datasets

In this study I used the datasets from the Hamp and Rost work [4] which are the benchmark datasets well accepted in the community. The work has engineered three types of datasets with varying levels of redundancy.

A and B denotes proteins in the test set. The following rules characterize the decreasing redundancy levels C1, C2 and C3.

- C1 - If both A and B, but not the interaction A–B, were used for training
- C2 if this was the case for either A or B
- C3 if neither A nor B were used for training

These are interactions in Human from the Database of Interacting Proteins at the UCLA.

Table 1 and 2 specifies the counts of different classes in training and testing sets.

C1	Positive	Negative
Training	758	7580
Testing	395	3950

Table 1 - C1 dataset number of samples in two classes

C3	Positive	Negative
Training	758	7580
Testing	395	3950

Table 2 – C3 dataset number of samples in two classes

Physicochemical Features

The proteins needs be represented as numerical vectors for performing the classification task. I used the following physicochemical features to represent the proteins. These features have been derived from decades of fundamental research.

Conjoint Triad - Three continuous amino acids are called a triad and features are calculated as unit based on dipoles and volumes of side chains.

Autocorrelation - These are a class of descriptors capturing the correlation between residues at different positions.

Dipeptide composition - This feature describes the frequency of occurrence of possible dipeptides.

Pseudo Amino Acid Composition – This encoding captures both positional and feature information.

Protein Embeddings from Pretrained Models

There has been a lot of development within deep learning with the advent of concept of Attention. Attention models have been demonstrated to outperform the sequential models in various key NLP tasks and are more efficient to train due to the architecture that lends itself to parallel computation. Bidirectional Encoding Representation from

Transformers uses Masked Language Model and Next sentence prediction task for pretraining on millions of sentences (or Proteins) to obtain a Pretrained model that captures the contextual dependencies. These pretraining has been performed on protein sequences as well, one of them is ProtBert (Elnaggar et al.) which is used by me to obtain protein embeddings. The Proteins are trained using Masked Language Model task only and not next sentence prediction.

These embeddings are used as features by the end classifier.

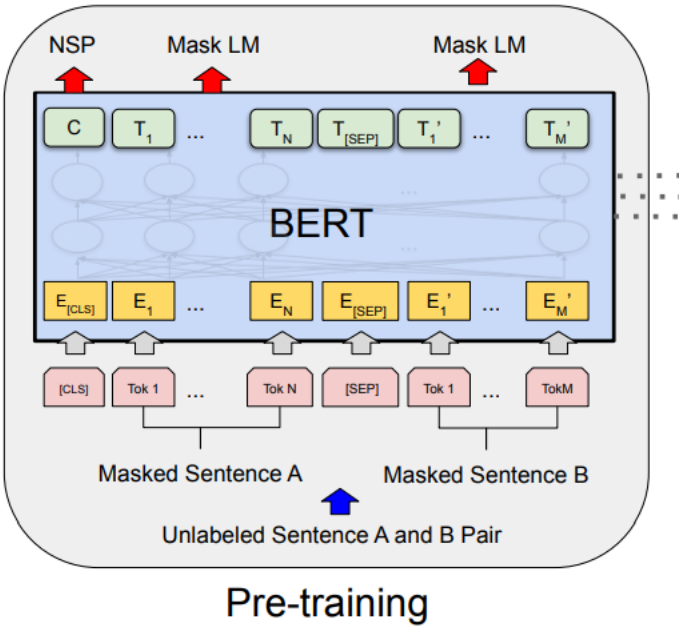

Fig 8 –BERT architecture (Devlin et al. 2019)

Classifier

Several classification algorithms – Logistic Regression, Random Forest and XGBoost were applied, and performance was evaluated on complementary metrics.

Logistic Regression had a reasonable test performance, while Random Forest and XGBoost had overfitting issues which required a large amount of hyper parameter tuning than that was available for completion of the project.

Results

I present the classification results of using GAN for oversampling versus without the use of any oversampling.

The classifier is evaluated on macro level metrics areas under the curve – Receiver Operating Characteristics (ROC) and Precision Recall (PR).

ROC curve is calculated using area under curve for true positive rate vs false positive rate at all thresholds. Similarly, PR AUC is calculated using area under curve for precision vs recall at all thresholds.

C3	PR AUC	ROC AUC
GAN	0.57	0.28
Non-Oversampled	0.58	0.28

Table 3- Results for Physicochemical properties on C3 dataset

C1	PR AUC	ROC AUC	Precision	Recall
GAN	0.57	0.28	0.7	0.44
Non-Oversampled	0.58	0.28	0.47	0.48

Table 4- Results for ProtBert embeddings on C1 dataset

C3	PR AUC	ROC AUC	Precision	Recall
GAN	0.5	0.67	0.58	0.36
Non-Oversampled	0.45	0.69	0.41	0.43

Table 5- Results for ProtBert embeddings on C3 dataset.

Results from using physicochemical properties is shown in Table 3, the ROC AUC is below 0.6 which indicates high noise levels in the feature, hence it is difficult to train a complex model such as GAN on it. Hence include GAN samples does not make a difference to the classification results.

There results using Pretrained embedding features are information rich which is reflected in higher classification performance.

The information rich features enable GAN to be able learn underlying distribution and synthesize higher quality samples.

Protein Protein Interaction prediction problem is focussed on detecting interactions, hence on the positive class i.e., on the Precision and Recall metrics, and PR AUC.

GAN sampling helps classifier improve on metric PR AUC which is achieved by a significant increase in precision with a little loss on recall.

This result is significant in data scare situations where there is a requirement of high-fidelity predictions on test data.

Conclusions and Future Directions

Generative AI approaches can help in data augmentation, where obtaining data (minority class) is costly and difficult. The high-quality samples can enable the classifier to learn patterns not easily obtained from original data, eventually boosting performance in detecting minority class.

In this study I have used a simple classifier – logistic regression which might not be able to learn complex nonlinear patterns in data. Hence, more complex neural networks can be evaluated on the data.

Secondly, GAN could be further tuned for any improvement in sample quality, eventually in classifier performance.

Finally, a broader comparison can be made with other oversampling techniques.

References

1. Google Developers https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data#:~:text=A%20classification%20data%20set%20with,smaller%20proportion%20are%20minority%20classes.
2. Google Developers https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data#:~:text=A%20classification%20data%20set%20with,smaller%20proportion%20are%20minority%20classes.
3. TAP Wikipedia https://en.wikipedia.org/wiki/Tandem_affinity_purification
4. Oxford Bioinf https://academic.oup.com/bioinformatics/article/31/12/1945/214196
5. GAN https://arxiv.org/pdf/1406.2661.pdf
6. Protein function https://www.eufic.org/en/whats-in-food/article/what-are-proteins-and-what-is-their-function-in-the-body
7. PPI Net https://www.ebi.ac.uk/training/online/courses/network-analysis-of-protein-interaction-data-an-introduction/protein-protein-interaction-networks/
8. TAP Schematic https://www.profacgen.com/tandem-affinity-purification-tap-ms.htm#:~:text=TAP)%2DMS-,Tandem%20Affinity%20Purification%20(TAP)%2DMS,screening%20interacting%20proteins%20are%20similar
9. SMOTE AV Blog https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/

.