GraphImmuno: Graph Convolutional Neural Networks with Physicochemical Properties for Adaptive Immune Response

Mallikharjuna Rao Sakhamuri
L00157129@atu.ie, ATU Donegal

ABSTRACT The emergence of the Covid-19 pandemic has brought much-needed attention to the adaptive human immune system. Adaptive immunity entails specific immune cells and antibodies that target and eliminate invading pathogens while also being capable of preventing illness in the future by developing a new immune method. To comprehend about the immune system, gaining knowledge about T-cell response, peptides, and HLA interactions is mandatory. Acquiring such valuable knowledge will significantly aid in the development of new drugs and antibiotics that activate T-cells to combat the malicious pathogen and improve human immunity. There are currently few models and tools based on traditional ML methods traditional AI's convolutional networks to predict and explain the immunogenicity values-based peptide and HLA's reciprocal action. Although these methods perform adequately, they fail to represent network structure and do not explain the amino acid correlations of peptide and HLA. This report proposes "GraphImmuno," a graph-based neural network endowed with physicochemical properties. The model has been validated on Covid and dengue data with high accuracies. Further, the proposed model also performed graph learning tasks on the immunogenicity data, such as supervised and unsupervised learning.

INDEX TERMS immunity artificial intelligence graph convolutional neural network peptides HLA pathogens COVID-19 anti-biotics.

I. INTRODUCTION

WITH the breakdown of covid 19 pandemic there is an increase in the awareness of the Immunogenicity and immune system of human. Immunogenicity explains how an external agent such as vaccine works well in the human body to improve the immune strength of the humans [1]. Thus, predicting the agents which increase the Immunity and their interaction with the cell's agents gained much significance in the recent days. This helps in producing the new antigens and new drugs which can effectively counter the disease-causing pathogens [2].

T-cells are responsible for fighting against the forging substances that enter a human body. Similarly, major histocompatibility complex (MHC) is a group of gens whose main functionality is to encode the peptide present on the surface of the cells [3]. Figuring out the interaction between HLA and the peptides of the foreign agents in the cell system helps to identify and make predictions about the new antigens that can boost the immunity of the humans and to identify the extern agents which can trigger HLA of the cells to fight against the virus [4]. Over the last few years with the advent of quality dataset of HLA helped in developing methods and tools which can identify the bonding between HLA types and the peptides [5].

Understanding these molecular networks and protein interaction complex networks may be best served by graph neural networks(GNN). GNN's prime objective for each amino acid node is to learn an embedding that includes details about its surroundings. Numerous applications, including node labelling, node prediction, edge prediction, and others, can make use of this embedding technique. Therefore, edges can be transformed by including feed-forward neural network layers and combine graphs with neural networks after assigning embeddings to each node. The properties of each amino acid combined with neighbour amino acids properties give better understanding in the context of peptide and HLA reactions for triggering the T-Cells.

This report proposes a novel approach to immunogenic peptides for T-cell immunity prediction using Graph-based convolutional neural networks. The proposed graph-based approach utilizes psychometric-aware encoding strategy to balance the performance across diverse test datasets. Further, it proposes a graph based neural network coupled with psychometric-aware encoding strategy that can work well with a graph model and is stable under various dataset sizes and perform graph-based tasks such as supervised classifi-

cation and unsupervised classification. The report aims to answer the following questions:

1) Can a graph model adequately describe the correlations and interactions that may exist in immunogenic peptides of covid disease data efficiently?
2) Is it feasible to conduct learning tasks on the graph representation of molecular networks, such as graph classification ?
3) Can physicochemical-aware encoding strategy improve the performance of the graph-based model to predict non-native peptides to elicit a T-cell response of covid disease data ?

II. DATA COLLECTION

Peptides are collected from the Immune Epitope Database after disease-related molecular data has been analysed. This data set contains only convalescent and unexposed Cell, Dengue, and Covid datasets. As part of the data preparation process, only MHC values with lengths greater than 4 are considered for analysis. The selected HLA and peptides are encoded to a number string using the physicochemical properties of the amino acid. The AAindex1 database contains amino acid information based on their physicochemical properties.

Dimensionality reduction is used because the initial list of properties chosen from the database is too large for calculation. Further, to normalize the outlier among the list of robust property values, I have used the RobustScaler library function. Only relevant features from the normalized feature matrix are retained after the noisy features are discorded.

Based on the peptide and T-cell response, the immunogenicity strength value is calculated using the beta distribution. Prior beta values are encoded using immunogenic class values. Totally, four immunogenic values that are taken into account (Negative, Low Positive, Intermediate Positive, and High Positive). The average values for the given combination of the HLA(MHC) and the peptide are considered after bootstrapped iterations on the distributed.

The AAindex1 database supplies 566 physicochemical attributes related to amino acids. 13 indices out of the 566 characteristics are eliminated because the data for some amino acids is lacking. A placeholder amino acid "-" is added for filling in the gaps in the 9-mer peptides and HLA paratope sequences. The average of the other 20 canonical amino acids was used to determine the corresponding AAindex values. This approach brings the total number of amino acids to 21. Principal component analysis (PCA) is used to normalize the resultant 21 x 553 numeric matrix, removing noisy features, and leaving just the pertinent components. The selection of 12 primary components, accounts for 95% of the variance. The HLA and peptide staring are formatted as per length by padding if needed and the HLA staring are encoded based on their paratope's sequences. Then the corresponding adjacency matrices are prepared for the HLA pseudo strings and peptide string values.

III. GRAPH MODEL

The combination of HLA and peptide is represented by an acyclic undirected graph. There are two kinds of edges in graphs. The connection between and within peptides and HLA is represented by intra edges, while the attraction between peptide and HLA is represented by inter edges. The weights are explicitly assigned by assigning a weight of 2 to the connection within and between HLA and a weight of 1 to the attraction between the HLA and peptide, emphasizing connections between the HLA and peptide. The node and edge data are converted into graphs for further analysis.

The model is fed HLA and peptide dictionary values, as well as the immunity data frame and physicochemical data. Peptide values are converted into strings by appending the alphabet "P" based on peptide length. This peptide string is used to create intra-peptide edges. Similarly, HLA values are converted to strings by appending the letter 'H,' and intra-HLA edges are formed. Intra edges represent interactions with HLA and peptides. The term "inter edges" refers to the relationships between the HLA and the peptide. Weights are assigned based on the edge connection type. The Graph model retunes the graph format with nodes and edges for the given input data.

A. GRAPHIMMUNO

This supervised graph classification model is made up of the graph classification classes and the dense layer. The adjacency matrices contain the Physicochemical properties of the HLA and peptides, which are fed into the model. The Physicochemical properties are graph node features and explicit weights assigned to edges based on interactions.

The imported layers from graph classification classes with 'relu' activation functions and 64 units. The mean pooling layer efficiently sums up the node representation from the graph to represent the graph. The network has two dense layers with unit sizes of 32 and 16, respectively, as well as 'relu' activation layers. The output layer has one unit, and the activation is 'sigmoid'. The sigmoid function is a non-linear function that produces an output between 0 and 1 by adding all the input weights.

The GraphImmuno model receives data in graph format, including nodes, edges, and weights. The model outputs a vector representation of the input data. In an iteration, the weights are aggregated and assigned to the edges. Convolution layers and pooling layers are combined as hidden layers to form a graph convolution model. The optimizer is used to compile the model. Loss and accuracy are mentioned as model evaluation metrics. As the test data are being fitted to the model, log values are streamed into the history parameter. The prediction function of the model predicts the value for the test input value.

Fig. 1 is the block diagram for GraphImmuno model. Graph data is fed to the model in the form of an adjacency matrix. The first few layers are convolutional graph layers that decode the graph data before processing it. The information is routed through a mean pooling layer. The data

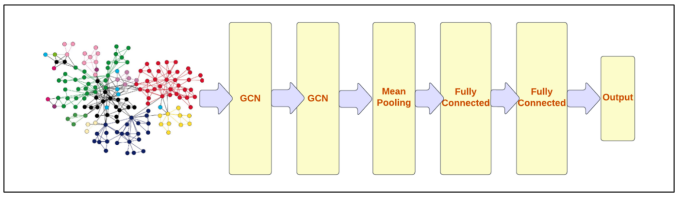

FIGURE 1. GraphImmuno - graph classification model.

in the pooling layer is represented as a graph by learning node representation. The data is then passed through fully connected layers to perform the supervised classification. The output layer is the final layer in the network and is responsible for binary classification.

B. DEEPGRAPHIMMUNO

DeepGraphImmuno is a GraphImmuno model that has been enhanced with additional layers to perform sort pooling and embedding to produce a representation of the graph data. The extraction of feature information from structured datasets is critical for graph-based neural networks. Sort pooling is a generic pooling method for aggregating graphs' given structured feature data into fixed dimensional representations [6]. Because high-dimensional vectors are translated into low-dimensional space, this is also known as embedding [7].

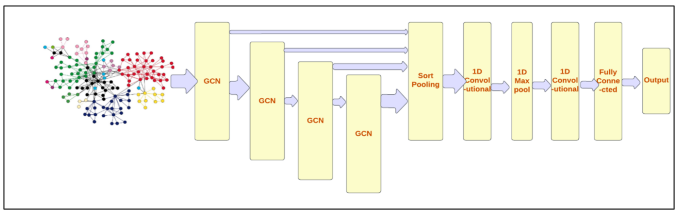

FIGURE 2. DeepGraphImmuno - graph classification model sort pooling layer

Fig. 2 illustrates the block diagram for DeepGraphImmuno. The adjacency matrix and node features of the graph serve as input to the model, which will be processed through the first four convolutional layers. The difference between these convolutional layers and the convolutional layers in the previous model is that, these layers use adjency normalized form. These layers are embedded with the "tanh" activation function and have units of 32, 32, 32, and 1 respectively. Except for the last layer, the dense layers in the network have "relu" as the activation function. The final output layer has a "sigmoid" function. The drop layer in the model's purpose is to prevent overfitting.

DeepGraphImmuno model accepts graph data in the form of adjacency matrices and node feature matrices and returns the vector representation of the given graph data. The supervised classification model builds the deep neural network layer by layer. The first layers are graph convolution layers. They are used in the network to perform normalization. The representation is made by adding a layer for sort pooling. This pooling layer is followed by a conventional layer, flatten layers, dense layer, and dropout layer. The final layer is the output layer, which has an activation function for non-linearity in the output value.

IV. PERFORMANCE EVALUATION OF GRAPHIMMUNO FOR IMMUNE RESPONSE

A. RESULTS AND ANALYSIS OF GRAPHIMMUNO

The data converted into graph format by stellargraph's mapper "PaddedGraphGenerator" is processed by a fully connected graph convolution neural network. The network's mean pooling layer pools the graph data for classification, and the results indicate that the model learns about the graph structures and features efficiently during the initial epochs and achieves a decent accuracy value during the first few rounds of training. With GCN supervised graph classification, the model performed well overall but with a significant variance after 10-fold cross validation. The small dataset size is most likely to be responsible for the high variance and the presence of noise in the data could be the other reason for variance. However, by tuning the model some improvement over the current baseline is possible. This model finishes a classification task that previous models had fail to finish. This straightforward GCN with mean pooling graph classification model is advantageous in comparison to graph kernel-based methods.

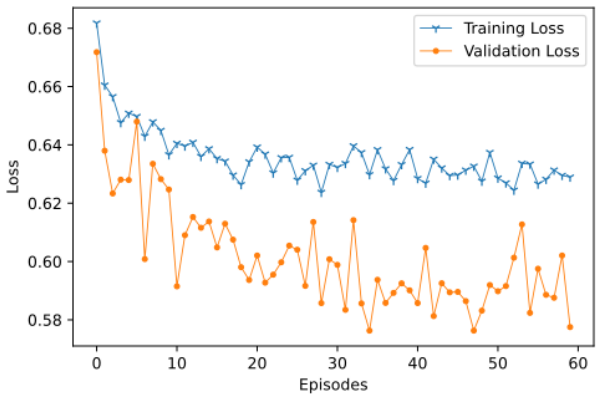

FIGURE 3. Loss value vs epochs for GraphImmuno.

Fig. 3 shows the loss value for the GraphImmuno model after 40 epochs. As seen in the graph, the test loss value dropped rapidly after the 10^{th} epoch, and the values are low when compared to the train loss values. The loss values begin at a high of 0.70 and gradually decreases to 0.58 over the epochs. According to the graph, convergence for both train and test loss values occur after the 20^{th} epoch.

Fig. 4 shows the accuracy value for the GraphImmuno model after 40 epochs. The graph shows that the train accuracy value stabilized after the 10^{th} epoch and that the values are relatively low in comparison to the test accuracy values. The accuracy values begin at 0.525 and gradually increase to 0.725 over the epochs. According to the graph,

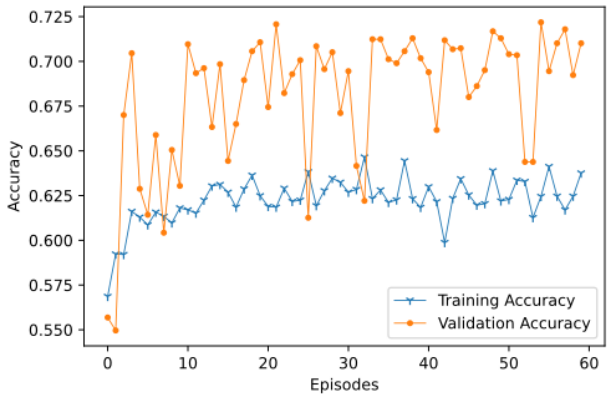

FIGURE 4. Accuracy value vs epochs for GraphImmuno.

Dataset	Measurement	Value
Validation dataset	Mean Squared Error	0.195
Dengue dataset	Accuracy	0.975
Cell data set	Recall	0.857
Covid data set (Convalescent)	Recall	0.920
Covid data set (Unexposed)	Recall	0.875

TABLE 1. GraphImmuno validation across various disease datasets.

convergence for both train and test accuracy occur after the 20^{th} epoch. The test accuracy is more unsteady, but the deviations are small. It can be assumed that as the number of epochs increases, the loss and accuracy values will improve.

B. RESULTS AND ANALYSIS OF GRAPHIMMUNO OVER DISEASE DATASETS

The graphs generated by StellarGraph's "padded graph generator" are trained using the graph classification model created by StellarGraph's GCNSupervisedGraphClassification layer. The model is run for 100 epochs and 10 folds, with the loss values dropping significantly as the model learned the features. The loss value of the test sets is noticeably lower than the loss value of the train data.

Table 1 shows various measurement values for the GCN-SupervisedGraphClassification model across various disease datasets. The accuracy value for the Dengue dataset is 0.97, indicating that the model learned the features and predicted immunogenicity very accurately. Even though the model's recall values are superior to those of Cell and Covid, the model experienced shortcut learning. The inability to perform well on different kind of data even though the model performs well on the known dataset is because of the shortcut learning [8]. Because of the shortcut learning, all of the predictive values are less than 0.5. This is because the graph's implied weight assignments may not fully reflect the true peptide-MHC interactions, resulting in ambiguous results.

C. RESULTS AND ANALYSIS OF DEEPGCN GRAPHIMMUNO

DeepGCN - GraphImmuno model updates the GCN - GraphImmuno model with sort pooling and performs a supervised graph classification task. The model is trained to predict the label of previously unseen graphs using a collection of graphs with a categorial label as input. Because of the additional layers, the model required more epochs to converge the loss and accuracy values than the GCN - GraphImmuno model. However, the additional layers did improve the accuracy. The network's convolutional and max pools receive the output from the sort-pool layer and process the graph data before sending it to the fully connected layer. As a result, the loss and accuracy values gradually change. The improved accuracy of the Supervised classification model reached 0.76 over 100 epochs and can be improved with a larger dataset, additional training, and parameter tuning.

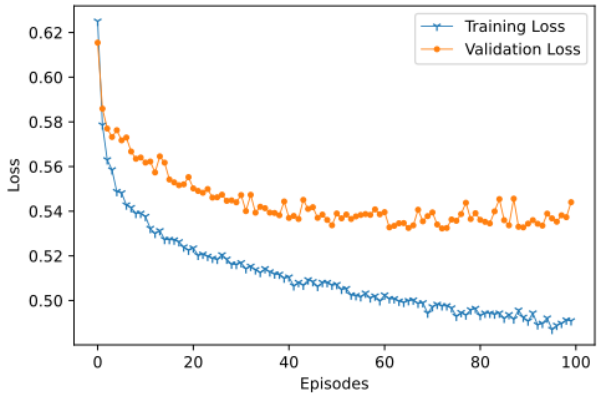

FIGURE 5. Loss value vs epochs for DeepGCN – GraphImmuno.

Fig. 5 depicts the loss value for the DeepGCN - GraphImmuno model over 100 epochs. The blue line represents the loss for the train data set, while the orange line represents the loss for the validation dataset. For both sets, the loss value begins at a high of 0.62 and gradually decreases to 0.50. The training set performs better because its loss values were lower.

Fig. 6 depicts the accuracy values for the DeepGCN - GraphImmuno model over 100 epochs. The accuracy of the supervised classification model was 0.74 for validation data and 0.76 for train data. During the 0th epoch, both values had accuracy values of 0.66, and convergence occurred after the 40th epoch. The spike in accuracy occurs during the first 5th epoch itself. Because of the sort-pooling methods and large number of epochs, the accuracy values are higher than in the GraphImmuno model.

V. CONCLUSION

After reviewing the research results, it is clear that GraphImmuno learned about the correlations and interactions that exist between peptides and HLAs. GraphImmuno also per-

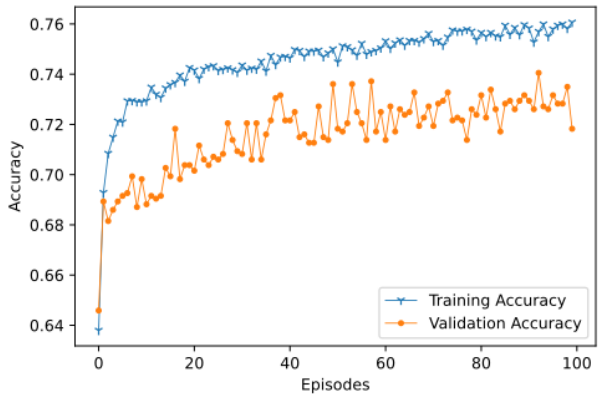

FIGURE 6. Accuracy value vs epochs for DeepGCN – GraphImmuno.

formed graph learning tasks such as supervised and unsupervised classification. GraphImmuno was validated on a dengue dataset and achieved an accuracy of 0.975, with a recall value of 0.92 for Covid convalescent. Furthermore, after a few initial epochs, the GraphImmuno loss value plummeted dramatically. This indicates that the graph model quickly learned about the feature values. By adding a sort pooling layer and more convolutional layers to the GraphImmuno model, it can now perform unsupervised classification. DeepGraphImmuno outperformed GraphImmuno in the supervised classification task. GraphImmuno with embedding performed unsupervised classification. There are a few future extensions that could be done with this graph model such as, currently the weights of the graphs are currently explicitly assigned, and these will be replaced with binding affinity values of amino acids.

VI. APPENDIX

The code files for this report are available in the GitHub. "MallikharjunaSakhamuri/MOCHAS_L00157129.git" is the reposity name in the GitHub.

REFERENCES

[1] Malone, B., Simovski, B., Moliné, C., Cheng, J., Gheorghe, M., Fontenelle, H., Vardaxis, I., Tennøe, S., Malmberg, J.-A., Stratford, R. and Clancy, T. (2020). Artificial intelligence predicts the immunogenic landscape of SARS-CoV-2 leading to universal blueprints for vaccine designs. Scientific Reports, 10(1). doi:10.1038/s41598-020-78758-5.

[2] Barra, C., Ackaert, C., Reynisson, B., Schockaert, J., Jessen, L.E., Watson, M., Jang, A., Comtois-Marotte, S., Goulet, J.-P., Pattijn, S., Paramithiotis, E. and Nielsen, M. (2020). Immunopeptidomic Data Integration to Artificial Neural Networks Enhances Protein-Drug Immunogenicity Prediction. Frontiers in Immunology, 11. doi:10.3389/fimmu.2020.01304.

[3] Doneva, N., Doytchinova, I. and Dimitrov, I. (2021). Predicting Immunogenicity Risk in Biopharmaceuticals. Symmetry, 13(3), p.388. doi:10.3390/sym13030388.

[4] Schaap-Johansen, A.-L., Vujović, M., Borch, A., Hadrup, S.R. and Marcatili, P. (2021). T Cell Epitope Prediction and Its Application to Immunotherapy. Frontiers in Immunology, 12. doi:10.3389/fimmu.2021.712488.

[5] Li, G., Iyer, B., Prasath, V.B.S., Ni, Y. and Salomonis, N. (2021). DeepImmuno: deep learning-empowered prediction and generation of immunogenic peptides for T-cell immunity. Briefings in Bioinformatics, 22(6). doi:10.1093/bib/bbab160.

[6] Naderializadeh, N., Comer, J., Andrews, R., Hoffmann, H., and Kolouri, S. 2021. Pooling by Sliced-Wasserstein Embedding. In Advances in Neural Information Processing Systems (pp. 3389–3400). Curran Associates, Inc..

[7] Bai, Y., Ding, H., Qiao, Y., Marinovic, A., Gu, K., Chen, T., Sun, Y. and Wang, W., 2019. Unsupervised inductive graph-level representation learning via graph-graph proximity. arXiv preprint arXiv:1904.01098.

[8] Geirhos, R., Jacobsen, J., Michaelis, C., Zemel, R., Brendel, W., Bethge, M. and Wichmann, F., 2020. Shortcut learning in deep neural networks. Nature Machine Intelligence, 2(11), pp.665-673.