

RESEARCH ARTICLE

GNNGLY: Graph Neural Networks for Glycan Classification

ALHASAN ALKUHLANI^{1,2}, WALAA GAD², MOHAMED ROUSHDY³,
AND ABDEL-BADEEH M. SALEM²

¹Faculty of Computer and Information Technology, Sana'a University, Sana'a, Yemen

²Faculty of Computer and Information Science, Ain Shams University, Cairo 11566, Egypt

³Faculty of Computers and Information Technology, Future University in Egypt, New Cairo 11835, Egypt

Corresponding author: Alhasan Alkuhlani (alhasan.alkuhlani@gmail.com)

ABSTRACT Glycans are important biological molecules that can be found on their own or attached to other molecules. They have complex, branching structures that do not follow the linear structure. Glycans are crucial for many biological processes and they are involved in the development of several important diseases. Due to the complexity and the branched structure of glycans, most of the current studies have mainly focused on the other attached molecules instead of glycans themselves. This paper proposes, GNNGLY, a graph neural networks model for glycans classification. Firstly, Glycans are represented as molecular graphs, where atoms are represented as nodes and bonds are represented as edges. Graph convolutional networks (GCNs) are then used to make predictions on eight taxonomic classification levels and for the level of immunogenicity property. The performance results indicate that GNNGLY outperforms traditional machine learning methods and when compared to other existing tools for glycan classification, GNNGLY showed considerable performance results. GNNGLY could have a significant impact on the field of glycoinformatics and related research areas.

INDEX TERMS Glycan, glycoinformatics, machine learning, graph neural networks, graph convolutional networks.

I. INTRODUCTION


Glycans or carbohydrates are important biological molecules that can be found on their own or attached to proteins, lipids, and other molecules. They are extremely diverse molecules that are found on the surface of all cells [1]. They have complex, branching structures made up of many different monosaccharides, and they do not follow the linear structure of DNA, RNA, and proteins, which are made up of only four nucleotides or 20 amino acids, respectively. Glycans do not conform to the central dogma of biology and cannot be studied using traditional sequencing techniques [2]. However, they are crucial for many biological processes such as protein function, cell-cell interaction, immune response, and overall organismal function. In addition, glycans are involved in the development of several important diseases [3], [4].

The study of glycans, or carbohydrates, has been limited by various challenges including the vast amount of data avail-

able in chemistry and biology, the complexity and diversity of carbohydrate molecules, and the branched structure of glycans, which is not template-driven like the synthesis of other biomolecules. As a result, current studies have mainly focused on proteins associated with glycans rather than the glycans themselves [3], [5]. Machine learning and deep learning have been used effectively for analyzing other types of biomolecules such as proteins and RNA, but these approaches rely on sequence-based representations that work well for linear structures but not for more complex structures like glycans [6].

A more general way to represent biomolecules is as a graph, with nodes representing atoms or monomers and edges representing bonds between them. This method can handle molecules with linear, branching, and cyclic structures. Artificial intelligence in graph representation has achieved excellent results in various fields including social media networks, chemistry, and bioinformatics.

In recent years, machine learning and deep learning have been applied to the analysis and classification of glycans.

The associate editor coordinating the review of this manuscript and approving it for publication was Long Wang .

SweetTalk, a glycan language model that employs recurrent neural networks, was developed by Bojar et al. [3]. It generates “glycowords” from glycans containing three monosaccharides and two bonds in order to capture the unique features and contextual information of the glycan. The model takes into account the composition and connectivity of the glycan. By employing this model, the team was able to explore motifs in glycan substructures, categorize them based on their O-/N-linkage, and accurately predict their immunogenicity at a rate of approximately 92%. Bojar et al. also created SweetOrigins [2], a language model-based method that extracts species-specific evolutionary information from glycans by training multiclass classifiers at each taxonomic level.

Burkholz et al. proposed SweetNet [1], which is based on deep learning and GCNs and converts glycans into a graph representation to predict glycan properties and features. The key aspect of their approach is the use of monosaccharides and glycan bonds as nodes and their connections as edges to build a neural network graph. Mohapatra et al. introduced a generalized approach called GLAMOUR [6], which represents glycan macromolecules as graphs with chemical information captured from molecular fingerprints. They applied supervised and unsupervised learning using different GNN models to classify glycans by taxonomic level and immunogenicity properties. GLAMOUR treats monosaccharide monomers as graph nodes and glycan bonds as graph edges. Dai et al. [7] develop a deep learning method, called glyBERT, to study the structure-function relationships of glycans. GlyBERT encodes glycans using a biochemical language and learns biologically relevant glycan representations by capturing both local and global context through an attention-based deep language model. The authors apply glyBERT to a variety of prediction tasks such as immunogenicity, glycoprotein linkage state, and taxonomic origin for the glycans.

The advancement of deep learning has resulted in the development of multiple neural network methods for handling graph and tree structures. Graph neural networks (GNNs) use the information present in the nodes and edges of a graph, as well as contextual information from the graph’s neighborhoods, to predict either individual nodes or the entire graph. The most common types of GNNs are message-passing neural networks (MPNNs) and GCNs [6]. In MPNNs, feature data is exchanged between neighboring nodes. A standard MPNN consists of several propagation layers, each of which is updated by aggregation functions based on the features of its neighbors.

There are three main types of aggregation functions used in GNNs: convolutional, attentional, and message passing [8]. GCNs, which are based on convolutional neural networks (CNNs), learn about graphs through multiple convolution operations. An iterative convolution filter is applied to the entire graph to process data from related nodes. Each convolution embeds the features of a node’s neighbors to represent the features of the node. The concept of “neighboring” then expands with each subsequent convolution, defining a larger

area of the graph. After this is done for every node and its neighbors, the resulting features are passed through the neural network for prediction [5]. GCNs have been used for studying social networks [9], protein function prediction [10], COVID-19 forecasting [11], and drug design studies [12], [13]. They have also been proposed for glycan analysis and prediction [1], [2], [3], [6].

In this research, glycans are represented as molecular graphs in which the different components of the glycan are represented as a structural formula in terms of graph theory. The atoms are depicted as nodes and the bonds between them are shown as edges. Graph neural networks (GNNs) are currently the best option for classifying these types of graphs. Given a labeled graph $L = (G_i, y_i)$, where y_i is the label of the graph G_i , the goal of graph classification is to build a model using L that can predict the labels of unlabeled graphs. The nodes and edges of the graph have associated features that can be used to classify the graph using GCNs. We developed several GCNs models for classifying glycans based on different taxonomic levels, which refer to the classification of organisms based on their evolutionary relationships. For instance, glycans can be classified based on the organism in which they are found, such as human glycans or plant glycans. We also classified glycans based on their properties, such as immunoglobulins which are glycans involved in immune defense. Moreover, we encode the glycan SMILES (simplified molecular-input line-entry system) into molecular fingerprint binary vectors and then use traditional machine learning methods for glycan classification and compare them with the proposed graph convolutional network models.

The remaining of the paper is structured as follows: Section II provides a detailed explanation of the used materials and methods, including dataset preparation, feature and graph representation, and machine learning methods. The results and analysis of the experiments are presented in Section III. Section IV is devoted to discussing the obtained results. Finally, Section V offers conclusions of the work.

II. MATERIALS AND METHODS

The proposed framework of GNNGLY is outlined in Figure 1. The first step is to gather and prepare the glycan data sets. The glycans are then converted into SMILES format and subsequently into molecular objects. These glycan molecular objects are then transformed into either sequence-based features for building traditional machine-learning classification models or into a graph for building GCNs models. Lastly, the models are evaluated and compared using various performance measures. The specific details of each step are discussed in this section.

A. DATA PREPARING

In this study, we used the Sugarbase v2.0 database [2] (<https://webapps.wyss.harvard.edu/sugarbase/>) as a source of data. The database contains 19,299 glycans represented in the IUPAC (International Union of Pure and Applied Chemistry) format. We obtained a labeled dataset on the taxonomic

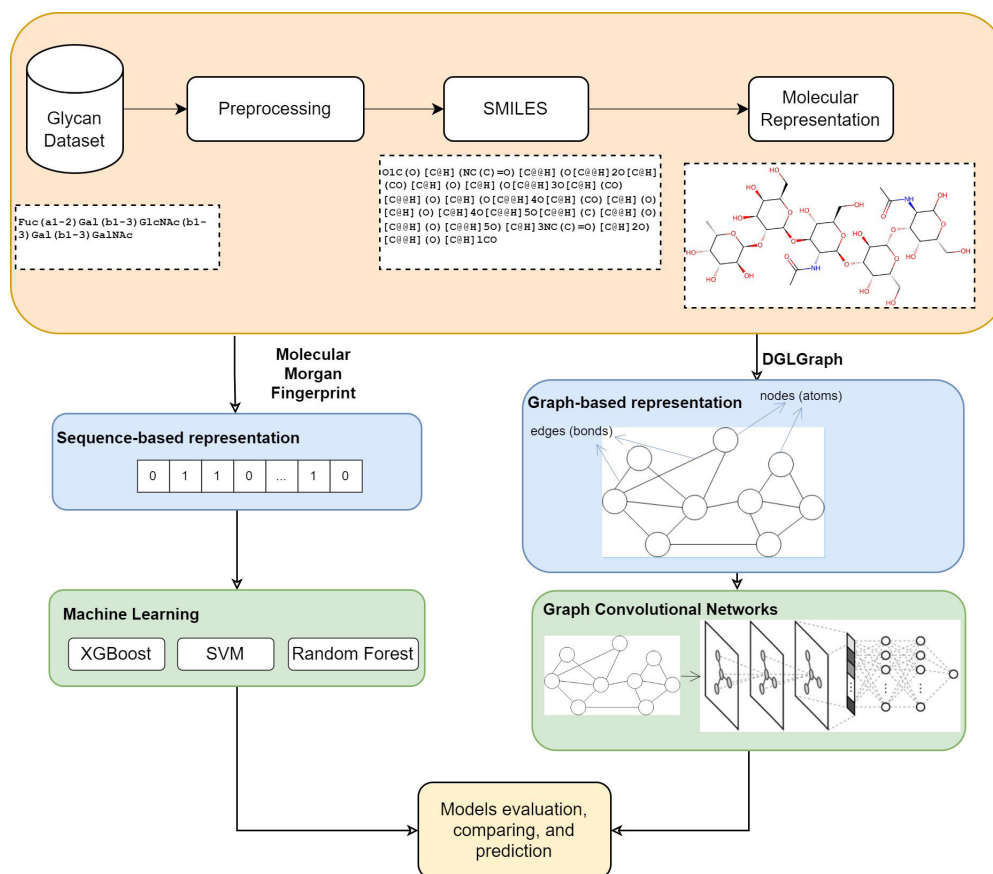


FIGURE 1. The general framework of GNNGLY.

level of glycans and their immunogenicity from Sugarbase, which had previously been used in the SweetNet [1] and SweetOrigins [2] studies. Bojar et al. collected the labeled data for SweetOrigins from three sources: previous scientific literature, UniCarbKB, and the Carbohydrate Structure Database (CSDB). This dataset includes 13,209 glycans that are classified into eight taxonomic levels: domain, kingdom, phylum, class, order, family, genus, and species. Each glycan in this dataset may belong to multiple taxonomies. Therefore, we created a new multilabel dataset from this dataset by adding each glycan to the new dataset with multiple labels separated by commas. The new dataset consists of 9,446 unique glycans, each with multiple labels at each taxonomic level. Using the GlyLES tool [14], we converted the glycan sequences to the SMILES format but had to remove 1,750 glycans because the tool was unable to convert them due to issues such as unequal or unclear parentheses or brackets, missing bonds, and branching brackets around the first monosaccharide.

Glycan classification involves considering both the immunogenic properties and the taxonomy of the glycan. Nine datasets have been created to reflect this, each containing the SMILES representation of the glycan sequence and multilabel classes for the glycan. Any labels with fewer than three glycans were removed, as were null or unknown labels and their associated glycans. Table 1 shows the

TABLE 1. Dataset class labels and the number of multilabel classes and samples for each classification level.

Class name	Number of multilabel classes	Number of samples
Immunogenicity	2	1,014
Domain	6	7,695
Kingdom	25	7,674
Phylum	49	7,601
Class	83	7,468
Order	151	7,202
Family	239	7,054
Genus	395	6,721
Species	499	6,166

number of labels and glycan samples in each dataset after preprocessing.

B. FEATURE REPRESENTATION

1) SMILES

SMILES (simplified molecular-input line-entry system) is a standard for expressing the structure of chemical compounds using brief ASCII strings that are understood by computer programs easily. For example, the SMILES string “CCO” represents the molecule ethanol, which consists of two carbon atoms, one oxygen atom, and three single bonds [14]. The glycan sequences with IUPAC notation are converted to SMILES using GlyLES python package that depends on the

ANTLR grammar parser generator (<https://www.antlr.org/>). Figure 2. Illustrates the example of the IUPAC glycan sequence, its glycan structure, its chemical structure, and its SMILES sequence.

2) FINGERPRINT REPRESENTATION

A molecular fingerprint is a fixed-length binary vector representation of a chemical compound. It is used to encode structural information about a molecule, such as its atoms and bonds, into a format that can be easily compared to other molecules. To convert a molecular object (a representation of a molecule as a set of atoms and bonds) into a fingerprint vector, a hashing algorithm is typically used to map each unique substructure of the molecule to a unique bit in the fingerprint vector. There are various methods to generate a fingerprint vector [15]. In this work, the Morgan fingerprint [16] is used to encode each glycan SMILES to a binary vector using the RDKit [17] python library. The method was first proposed by Robert Morgan in the late 1960s. It is based on the concept of a molecular “fingerprint” that is generated by the unique environment of each atom in a molecule. To generate a Morgan fingerprint, the algorithm starts with an atom in the molecule and performs a breadth-first search to identify all other atoms that are within a specified radius (typically up to a radius of 2 or 3 bonds). The resulting set of atoms and bonds is hashed to a unique bit in the fingerprint vector. The process is repeated starting from each atom in the molecule to generate the final fingerprint. This process is repeated for all atoms in the molecule to generate the final fingerprint, which in this case has a length of 128 bits or features for each glycan.

3) GRAPH REPRESENTATION

RDKit [17] is used to convert the glycan SMILES strings to molecule object that has various functions to get the chemical atoms and bonds of the chemical structure of glycan. The glycan molecule objects then are utilized to construct glycan graphs as DGL (deep graph library) graphs [18]. The molecule atoms as represented as nodes and the molecule bonds between them are represented as edges. The chemical RDKit library is also used to extract chemical features for each node (atom) and edge (bond). The general notation for the glycan graph can be represented as $G = (V, E, X)$ in which V is a set of the graph nodes (atoms) with length n , E is the set of graph edges (bonds) with length m , and $X \in R$ represents the features of the nodes and edges graphs. The adjacency matrix A for the graph is a $n \times n$ matrix where $A_{ij} = 0$ if $e_{ij} \notin E$ and $A_{ij} = 1$ if $e_{ij} \in E$. The set of neighborhood nodes of a node v is represented as $N(v) = u \in V | (v, u) \in E$. Let node $v_i \in V$, $e_{ij} \in E$ is the edge that connects the node v_i with v_j , $x_v \in R^c$ represent the feature vector of with length c , and $x_{u,v}^e \in R^d$ is the feature vector for the edge (u, v) with length d [19], [20].

4) GRAPH FEATURES

The RDKit chemical library is used to extract various features at the node (atom) and edge (bond) level for a graph.

These features, known as atom and edge feature descriptors, encode different properties of the molecule. Six types of atom features are used, including the atomic number, chiral type, degree, formal charge, number of hydrogens, and hybridization type. For edges, three bond features are used: bond type, presence in a ring, and conjugation [21]. Each feature descriptor is transformed into a one-hot vector. The resulting node feature vector has a size of 133, while the edge feature vector has a size of 14. The descriptions of these node and edge feature descriptors can be found in Table 2.

C. MACHINE LEARNING (ML)

1) GRAPH CONVOLUTIONAL NETWORKS

Glycan classification is a type of graph-level classification that aims to predict the class label(s) for a glycan based on the entire graph. One of the most effective methods for this task is the use of graph neural networks (GNNs). A specific type of GNN, known as a graph convolutional networks (GCNs), was developed by Kipf and Welling [22] to handle irregularly shaped graphs that cannot be processed by traditional convolutional neural networks (CNNs). CNNs are designed to work on regular, Euclidean structures, like images, while GCNs are better suited for irregular, non-Euclidean structures, such as graphs where the number of edges between nodes varies and the nodes are not arranged in a regular pattern [19].

In general, GCNs are composed of three main parts: (A) graph convolutional layers that extract high-level features from the graph by using an aggregation function, (B) graph pooling layers that reduce the graph structure by coarsening it into a sub-graph at each iteration, and (C) a readout layer that combines the final representations of each graph [19], [20], [23]. These components work together to create an end-to-end model for graph prediction, which can be used to classify glycan graphs. The graph convolutional layers are utilized to build a high-level node representation for node v by aggregating its features x_v with its neighbors' features x_u where $u \in N(v)$. Each layer k of the convolutional layers operates on the hidden state of nodes $h^{(k)} \in R^{N \times m_k}$, where N is the number of nodes and m_k is the number of hidden units in layer k . At each layer k , the hidden state of node v is updated as:

$$h_v^{(k)} = \sigma \left(\sum_{u \in N(v)} \frac{1}{c_{u,v}} W^{(k)} h_u^{(k-1)} + b^{(k)} \right) \quad (1)$$

where $h_u^{(k-1)}$ is the hidden state of node u at layer $k - 1$, σ is the activation function such as ReLU or sigmoid, $W^{(k)}$ and $b^{(k)}$ are the learnable weight matrix and bias vector of layer k , respectively, and $c_{u,v}$ is a normalization constant that adjusts for the degree of node u and v . The normalization constant is defined as:

$$c_{u,v} = \sqrt{\deg(u) \times \deg(v)} \quad (2)$$

where $\deg(u)$ and $\deg(v)$ are the degrees of node u and v , respectively. The graph pooling layers are used to coarsen

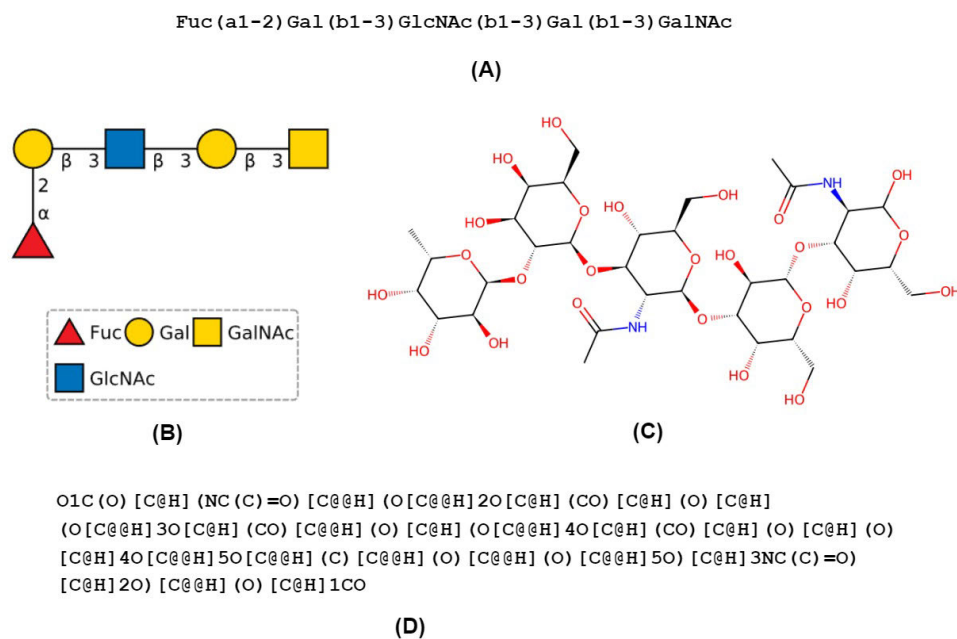


FIGURE 2. Glycan representation example. (A) Glycan in the IUPAC format. (B) Glycan structure in SNFG (Symbol Nomenclature for Glycans) standards. (C) Chemical structure of glycan. (D) SMILES notation for the glycan.

TABLE 2. The node and edge feature descriptors.

Descriptor Level	Feature Name	Description	Value Range
node	atomic number	atomic number (0-100)	0 to 100
	chiral tag	chiral type	0 to 3
	atom degree	number of directly connected neighbors.	
	charge	formal charge	-2 to 2
	number of hydrogens	number of hydrogens in the atom	0 to 4
	hybridization	hybridization	SP, SP2, SP3, SP3D, SP3D2
edge	bond type	bond type	single, double, triple, and aromatic
	is ring	bond existing in a ring or not	
	Is conjugated	bond is considered to be conjugated	

the graph structure into a sub-graph at each iteration. The pooling allows for deeper networks which can help to reduce overfitting and computational complexity [24]. The pooling operation typically involves grouping nodes into clusters or super-nodes, based on some criteria such as node degree, node importance, or clustering coefficients. The sub-graphs obtained after pooling can then be fed into the next layer of the GCNs for further processing. The readout layer then aggregates the final representations of each graph to make a prediction for the entire graph. One common way to implement the readout layer is to use a permutation-invariant function, such as summation or average pooling, over all node representations. This can be represented mathematically as:

$$h_G = \sum_{v=1}^N \alpha_v h_v^{(L)} \quad (3)$$

where h_G is the final representation of the entire graph, $h_v^{(L)}$ is the hidden state of node v at the final layer L of the GCNs,

and α_v is a scalar weight assigned to node v that depends on its importance in the graph.

After the readout layer combines the final node representations, the output is processed by a dropout layer and fully connected networks to make predictions. The dropout layer randomly drops out some of the activations to prevent overfitting. The fully connected networks consist of multiple layers of densely connected nodes that transform the output of the readout layer into a format suitable for prediction. Each node in the fully connected layers applies a linear transformation to its input, followed by a non-linear activation function, such as ReLU or sigmoid. The output of the last layer of the fully connected networks is passed through a final activation function, such as softmax or sigmoid, to obtain the final prediction.

During training, the model is optimized using the BCE-WithLogitsLoss loss function, which calculates the binary cross-entropy between the true target labels and the predicted probabilities. The Adam optimizer is also used to adjust the model's weights during training, with hyperparameters

TABLE 3. Tuned parameter setting for GCNs models.

Hyperparameter	Value
Learning rate	0
Dropout	0.1
Batch size	256
Max epoch	200
GNN layers number	5
Patience	30
in_feats	133
hidden_feats	14

such as learning rate, maximum epoch, and batch size tuned to optimize performance. An early-stopping strategy is also implemented to prevent overfitting, where training stops if the validation loss does not improve after a certain number of epochs, such as 30 in this case. To optimize the GCNs model for glycan classification, multiple hyperparameter values were experimented with. The final set of hyperparameters used in the GCNs model was determined based on the validation performance results from this experimentation process. The specific hyperparameters used can be found in Table 3.

2) TRADITIONAL ML METHODS

In addition to GCNs, this study also employs other traditional Machine Learning (ML) techniques to develop prediction models for glycan classification using fingerprint sequence representations of glycans. The effectiveness of the graph representation and GCNs are evaluated by comparing the GCNs model with three traditional machine learning algorithms: Support Vector Machine (SVM), Random Forest (RF), and XGBoost (XGB). The SVM and RF algorithms are implemented using the Scikit-learn Python library, while XGBoost is implemented using the XGBoost Python library. SVM is a widely-used classifier that aims to classify by determining the optimal separator hyperplane that maximizes the margin between classes. RF is an ensemble-based algorithm that is based on decision trees and is commonly used in computational biology for its simplicity and suitability for high-dimensional data. XGBoost is also an ensemble learning method that is based on tree boosting and uses gradient descent to deal with high dimensional data [25], [26]. Multiple hyperparameters are experimented with each classifier using training and validation datasets. Table 4 shows the hyperparameters used for the three machine learning classifiers, along with the corresponding values for each parameter.

III. RESULTS

In the GNNGLY, nine different GCNs models were developed to classify glycans based on their taxonomic level and immunogenicity level. All the models were built using the same architecture, with the only difference being the number of nodes in the output layer. The number of nodes in the output layer varied for each taxonomic level, depending on the number of classes for that level. The models were trained using the PyTorch and DGLgraph libraries on a GPU. The performance of the models was evaluated using both ten-fold

TABLE 4. The hyperparameter configurations for the machine learning classifiers used in this study, along with the corresponding values for each parameter.

ML classifier	parameter	range of values	used value
SVM	kernal	linear, poly, rbf, sigmoid	rbf
	C	1e-1, 1, ..., 1e6	1
	gama	scale, auto	scale
RF	n_estimators	100 to 1000	100
XGBoost	booster	gbtree, gblinear	gbtree
	learning_rate	0.01, 0.02, ..., 0.1	0.05
	max_depth	2 to 10	4
	n_estimators	100 to 1000	200

cross-validation and independent testing. The dataset was randomly split into training, validation, and testing sets in the ratio of 60%, 20%, and 20% respectively. The training set was used for both training the models and for cross-validation. The validation set was used to fine-tune the model's hyperparameters and weights, and the testing set was used for independent testing and comparison with other tools.

Five performance metrics were used for model evaluation and testing: ROC-AUC (Area Under the Receiver Operating Characteristic Curve), F1-score, recall, precision, and accuracy [4], [27]. The task of classifying glycans based on their taxonomic origin was treated as a multi-class and multi-label classification problem since glycans can be present in multiple types of organisms. In contrast, for the classification of glycans based on their immunogenicity, the dataset had only two classes: yes or no. The number of classes and samples for each task was shown in Table 1.

A. RESULTS USING CROSS-VALIDATION

The first strategy to evaluate GNNGLY is the ten-fold cross-validation. The data is divided into ten folds, and the model is trained and evaluated ten times, each time using a different fold as the test set and the remaining folds as the training set. The final performance metric is then the average performance across all ten iterations. Table 5 presents the results of the ten-fold cross-validation performance for glycan classification using GNNGLY, SVM, RF, and XGB on both the taxonomic levels (8 levels) of glycans and the immunogenicity property level of glycans.

The results of the ten-fold cross-validation are represented by the ten iterations' average results \pm standard deviation in terms of ROC-AUC, F1, recall, precision, and accuracy performance measures. The low standard deviations showed that the models' overall prediction abilities are stable. The results indicate that GNNGLY models surpass all other ML methods for all prediction tasks in terms of accuracy, recall, precision, F1-score, and AUC performance metrics (as seen in Table 4). GNNGLY showed an average improvement of 0.01 in accuracy for the immunogenicity level, domain level, 0.07 for the kingdom level, 0.1 for phylum level, 0.07 for the class level, 0.13 for the order level, 0.03 for the 0.13 for family level, 0.11 for genus level, and 0.11 for the species

TABLE 5. Ten-fold cross-validation performance results of GNNGLY compared to SVM, RF, and XGB. The results are represented by the average \pm standard deviation in the terms of accuracy, recall, precision, F1-score, and AUC metrics for all glycan classification levels.

Classification level	Model	Accuracy	Recall	Precision	F1	AUC
Immunogenicity	GNNGLY	0.975 \pm 0.011	0.975 \pm 0.011	0.975 \pm 0.011	0.975 \pm 0.011	0.976 \pm 0.009
	SVM	0.959 \pm 0.02	0.959 \pm 0.02	0.959 \pm 0.02	0.959 \pm 0.02	0.95 \pm 0.025
	RF	0.967 \pm 0.019	0.967 \pm 0.019	0.967 \pm 0.019	0.967 \pm 0.019	0.964 \pm 0.022
	XGB	0.963 \pm 0.026	0.963 \pm 0.026	0.963 \pm 0.026	0.963 \pm 0.026	0.955 \pm 0.033
Domain	GNNGLY	0.928 \pm 0.01	0.932 \pm 0.01	0.941 \pm 0.008	0.936 \pm 0.009	0.933 \pm 0.015
	SVM	0.847 \pm 0.009	0.865 \pm 0.011	0.857 \pm 0.01	0.861 \pm 0.01	0.915 \pm 0.006
	RF	0.91 \pm 0.008	0.928 \pm 0.006	0.914 \pm 0.008	0.921 \pm 0.006	0.95 \pm 0.004
	XGB	0.899 \pm 0.008	0.921 \pm 0.008	0.908 \pm 0.008	0.914 \pm 0.007	0.946 \pm 0.005
Kingdom	GNNGLY	0.898 \pm 0.008	0.901 \pm 0.008	0.943 \pm 0.009	0.922 \pm 0.006	0.948 \pm 0.014
	SVM	0.77 \pm 0.018	0.874 \pm 0.017	0.805 \pm 0.014	0.838 \pm 0.014	0.9 \pm 0.007
	RF	0.855 \pm 0.013	0.916 \pm 0.013	0.858 \pm 0.014	0.886 \pm 0.012	0.927 \pm 0.007
	XGB	0.846 \pm 0.009	0.918 \pm 0.012	0.857 \pm 0.01	0.886 \pm 0.007	0.927 \pm 0.005
Phylum	GNNGLY	0.832 \pm 0.016	0.841 \pm 0.014	0.896 \pm 0.014	0.867 \pm 0.013	0.934 \pm 0.01
	SVM	0.673 \pm 0.021	0.821 \pm 0.019	0.708 \pm 0.019	0.761 \pm 0.016	0.853 \pm 0.009
	RF	0.773 \pm 0.015	0.881 \pm 0.01	0.775 \pm 0.014	0.824 \pm 0.011	0.886 \pm 0.007
	XGB	0.758 \pm 0.019	0.879 \pm 0.012	0.774 \pm 0.019	0.823 \pm 0.015	0.886 \pm 0.01
Class	GNNGLY	0.655 \pm 0.024	0.65 \pm 0.05	0.817 \pm 0.025	0.723 \pm 0.036	0.827 \pm 0.023
	SVM	0.524 \pm 0.019	0.743 \pm 0.021	0.558 \pm 0.021	0.638 \pm 0.02	0.778 \pm 0.01
	RF	0.635 \pm 0.015	0.833 \pm 0.018	0.636 \pm 0.014	0.721 \pm 0.014	0.817 \pm 0.007
	XGB	0.621 \pm 0.02	0.83 \pm 0.016	0.635 \pm 0.018	0.72 \pm 0.017	0.817 \pm 0.009
Order	GNNGLY	0.583 \pm 0.019	0.607 \pm 0.026	0.785 \pm 0.018	0.684 \pm 0.016	0.79 \pm 0.026
	SVM	0.382 \pm 0.012	0.695 \pm 0.028	0.411 \pm 0.014	0.517 \pm 0.015	0.705 \pm 0.007
	RF	0.469 \pm 0.026	0.816 \pm 0.019	0.47 \pm 0.026	0.596 \pm 0.024	0.734 \pm 0.013
	XGB	0.48 \pm 0.021	0.818 \pm 0.019	0.49 \pm 0.022	0.613 \pm 0.022	0.745 \pm 0.011
Family	GNNGLY	0.534 \pm 0.026	0.55 \pm 0.029	0.797 \pm 0.02	0.65 \pm 0.023	0.775 \pm 0.027
	SVM	0.355 \pm 0.017	0.66 \pm 0.028	0.388 \pm 0.018	0.489 \pm 0.019	0.694 \pm 0.009
	RF	0.414 \pm 0.014	0.8 \pm 0.024	0.414 \pm 0.013	0.546 \pm 0.013	0.707 \pm 0.007
	XGB	0.432 \pm 0.018	0.804 \pm 0.019	0.439 \pm 0.017	0.568 \pm 0.017	0.719 \pm 0.009
Genus	GNNGLY	0.479 \pm 0.021	0.496 \pm 0.021	0.763 \pm 0.025	0.6 \pm 0.013	0.744 \pm 0.019
	SVM	0.331 \pm 0.017	0.633 \pm 0.034	0.361 \pm 0.014	0.46 \pm 0.018	0.68 \pm 0.007
	RF	0.372 \pm 0.027	0.778 \pm 0.032	0.373 \pm 0.027	0.504 \pm 0.03	0.686 \pm 0.013
	XGB	0.384 \pm 0.017	0.789 \pm 0.025	0.392 \pm 0.018	0.524 \pm 0.021	0.696 \pm 0.009
Species	GNNGLY	0.453 \pm 0.035	0.472 \pm 0.038	0.765 \pm 0.03	0.583 \pm 0.034	0.743 \pm 0.034
	SVM	0.373 \pm 0.018	0.629 \pm 0.032	0.408 \pm 0.016	0.494 \pm 0.02	0.704 \pm 0.008
	RF	0.357 \pm 0.031	0.756 \pm 0.032	0.357 \pm 0.03	0.484 \pm 0.033	0.678 \pm 0.015
	XGB	0.308 \pm 0.023	0.808 \pm 0.03	0.314 \pm 0.023	0.452 \pm 0.028	0.657 \pm 0.011

level, over the SVM, RF, and XGB methods. In comparison to traditional ML methods (SVM, RF, and XGB), it was observed that the ensemble-based methods (RF, and XGB) perform better than SVM in most multi-label class data.

B. RESULTS USING INDEPENDENT TESTING

To evaluate the GNNGLY models further, independent testing is used, where the models are tested on a separate set of data and the results are compared with those of the SVM, RF, and XGB ML methods. The performance results for this prediction are presented in Table 6.

The results of the independent test also showed that GNNGLY models outperform the traditional ML methods in the terms of accuracy, precision, recall, F1, and AUC performance metrics. GNNGLY showed an average improvement of 0.04 in accuracy for the immunogenicity level, 0.03 for the domain level, 0.07 for the kingdom level, 0.1 for phylum level, 0.13 for the class level, 0.11 for the order level, 0.11 for family level, 0.05 for genus level, and 0.14 for the species level over the SVM, RF, and XGB methods.

Generally, the utilization of graph feature representation in combination with the GCNs classification method has proven to be more efficient than using sequence-based representation with traditional ML techniques for glycans prediction tasks. This is because graph representation captures the structural information and relationships among different parts of the data, which can be more informative than linear sequence information alone. Furthermore, GCNs are able to perform convolution operations on graph-structured data, which enables them to effectively extract features from the graph, resulting in improved classification performance.

C. COMPARING WITH EXISTING METHODS

GNNGLY for glycan classification is compared to two existing tools, SweetNet [1] and GLAMOUR [6], using the same datasets. All three methods utilize GCNs for glycan classification, but they differ in their representation of the glycan graph. SweetNet uses monosaccharides and bounds as nodes and connections between them as edges, while GLAMOUR employs text files containing SMILES sequences, monomer

TABLE 6. Independent testing performance results of GNNGLY compared to SVM, RF, and XGB in the terms of accuracy, recall, precision, F1-score, and AUC metrics for all glycan classification levels.

Classification level	Model	Accuracy	Recall	Precision	F1	AUC
Immunogenicity	GNNGLY	0.985	0.985	0.985	0.985	0.989
	SVM	0.936	0.936	0.936	0.936	0.925
	RF	0.951	0.951	0.951	0.951	0.956
	XGB	0.941	0.941	0.941	0.941	0.940
Domain	GNNGLY	0.933	0.944	0.948	0.946	0.949
	SVM	0.864	0.883	0.872	0.878	0.924
	RF	0.917	0.928	0.923	0.926	0.955
	XGB	0.913	0.928	0.917	0.923	0.952
Kingdom	GNNGLY	0.912	0.913	0.939	0.926	0.955
	SVM	0.774	0.869	0.812	0.839	0.903
	RF	0.870	0.922	0.873	0.897	0.935
	XGB	0.854	0.917	0.871	0.893	0.934
Phylum	GNNGLY	0.838	0.846	0.908	0.876	0.922
	SVM	0.670	0.828	0.705	0.762	0.851
	RF	0.765	0.885	0.767	0.822	0.883
	XGB	0.757	0.883	0.771	0.823	0.884
Class	GNNGLY	0.730	0.745	0.849	0.794	0.872
	SVM	0.529	0.768	0.555	0.644	0.776
	RF	0.653	0.835	0.654	0.733	0.826
	XGB	0.637	0.831	0.653	0.731	0.826
Order	GNNGLY	0.589	0.619	0.794	0.696	0.809
	SVM	0.414	0.724	0.443	0.550	0.721
	RF	0.491	0.800	0.491	0.609	0.745
	XGB	0.504	0.823	0.507	0.628	0.753
Family	GNNGLY	0.545	0.567	0.762	0.650	0.792
	SVM	0.383	0.704	0.419	0.525	0.709
	RF	0.449	0.824	0.451	0.583	0.725
	XGB	0.467	0.810	0.476	0.599	0.738
Genus	GNNGLY	0.495	0.522	0.768	0.622	0.761
	SVM	0.342	0.667	0.373	0.479	0.686
	RF	0.401	0.771	0.401	0.528	0.701
	XGB	0.427	0.802	0.432	0.562	0.716
Species	GNNGLY	0.478	0.505	0.760	0.606	0.766
	SVM	0.358	0.634	0.399	0.490	0.699
	RF	0.346	0.752	0.346	0.474	0.673
	XGB	0.286	0.783	0.292	0.426	0.646

positions, and bond connections, with monosaccharides as nodes and bonds as edges. GLAMOUR extracts features using the Morgan fingerprint RDKit Chem library for each monosaccharide and bond. The performance results of the comparison are presented in Figure 3. From the figure we illustrated that GNNGLY exhibits competitive performance results compared to the other existing tools.

IV. DISCUSSION

The complex and branched structure of glycans makes traditional sequence-based methods used for DNA and protein sequences ineffective for studying them. A more effective approach is to use graph representation, which is a more general way to represent glycans for computational techniques. However, there are only a few studies on glycan classification using graph representation and graph neural networks, highlighting the need to improve classification results and develop new glycan classifiers. In this work, we introduce GNNGLY, a novel glycan classifier based on GCNs, that

is able to classify glycans which represented as molecular graphs on nine classification levels.

The performance results of GNNGLY are compared with three supervised ML techniques. The glycans data is represented as binary vectors using molecular fingerprint techniques for ML models, while they are represented as graphs using molecular graphs and DGL graphs for GNNGLY models. The results from Tables 5 and 6 show that the performance of GNNGLY models outperforms the ML models on all classification levels. Ensemble-based methods such as RF and XGBoost show better performance than SVM in most multilabel class data that have the largest number of classes. This indicates that tree-based and ensemble learning methods can perform better than other ML methods with large multilabel classes.

The utilization of graph feature representation in combination with the GCNs classification method has proven to be more efficient than using sequence-based representation with traditional ML techniques for glycans prediction tasks. This is because graph representation captures the structural



FIGURE 3. Performance results of GNNGLY compared to SweetNet, and GLAMOUR tools on the independent dataset in the term of accuracy metric.

information and relationships among different parts of the data, which can be more informative than linear sequence information alone. Furthermore, GCNs can perform convolution operations on graph-structured data, which enables them to effectively extract features from the graph, resulting in improved classification performance.

Moreover, GNNGLY is compared with two recent existing methods for glycan classification, SweetNet [1] and GLAMOUR [6], using the same dataset (Sugarbase). GNNGLY shows better performance results with the Species, Genus, Family, and Order classification levels. On the other hand, SweetNet exceeds GNNGLY by 1.5% on the Class classification level, and by 0.3% on the Phylum classification level. GLAMOUR exceeds GNNGLY by 0.9% on the Kingdom classification level. Additionally, SweetNet exceeds GNNGLY by 0.6% on the Domain classification level, and by 0.4% on the Immunogenicity classification level. Overall, GNNGLY shows very close performance results to SweetNet or GLAMOUR results. With the large number of multilabel classes (Species, Genus, Family, and Order), GNNGLY performs better than SweetNet and GLAMOUR.

Based on the data presented in Table 1 and the results in Section III, it can be observed that the performance of the models decreases as the number of multilabel classes increases. For more clarification, the Species classification level is the most challenging task with the largest number of multilabel classes (499), resulting in the lowest prediction performance across all models. This is because most classes have a small number of samples, making the dataset highly imbalanced. However, GNNGLY outperforms other ML methods and existing methods on the large number of multilabel classes.

To improve the classification performance with large multilabel classes, data augmentation can be used to increase the amount of data available for training. This can be done in the future to improve the performance of glycan classification.

V. CONCLUSION

The paper presents, GNNGLY, a new approach for classifying glycans, which are important biological molecules that play a crucial role in many biological processes and are involved in the development of several diseases. Due to the complexity and branched structure of the glycans, previous research has not primarily focused on studying glycans themselves. GNNGLY addresses this challenge by representing glycans as molecular graphs and using graph convolutional neural networks to make predictions on eight taxonomic classification levels and for the level of immunogenicity property. Results indicate that this approach outperforms traditional machine learning methods including SVM, RF, and XGB in terms of accuracy, precision, recall, F1, and AUC performance metrics. It is more efficient than using sequence-based representation with traditional ML techniques because graph representation captures structural information and relationships among data. Moreover, GCNs can also effectively extract features from the graph which results in improved classification performance. It was also compared to existing tools, SweetNet and GLAMOUR, and showed good performance results, suggesting that it could be a useful tool in the field of glycan classification.

REFERENCES

- [1] R. Burkholz, J. Quackenbush, and D. Bojar, "Using graph convolutional neural networks to learn a representation for glycans," *Cell Rep.*, vol. 35, no. 11, Jun. 2021, Art. no. 109251.
- [2] D. Bojar, R. K. Powers, D. M. Camacho, and J. J. Collins, "Deep-learning resources for studying glycan-mediated host-microbe interactions," *Cell Host Microbe*, vol. 29, no. 1, pp. 132.e3–144.e3, Jan. 2021.
- [3] D. Bojar, D. M. Camacho, and J. J. Collins, "Using natural language processing to learn the grammar of glycans," *bioRxiv*, 2020, doi: 10.1101/2020.01.10.902114.
- [4] A. Alkuhlani, W. Gad, M. Roushdy, and A. M. Salem, "PUStackNGly: Positive-unlabeled and stacking learning for N-linked glycosylation site prediction," *IEEE Access*, vol. 10, pp. 12702–12713, 2022.
- [5] D. Bojar and F. Lisacek, "Glycoinformatics in the artificial intelligence era," *Chem. Rev.*, vol. 122, no. 20, pp. 15971–15988, Oct. 2022.

- [6] S. Mohapatra, J. An, and R. Gómez-Bombarelli, "Chemistry-informed macromolecule graph representation for similarity computation, unsupervised and supervised learning," *Mach. Learn., Sci. Technol.*, vol. 3, no. 1, Mar. 2022, Art. no. 015028.
- [7] B. Dai, D. E. Mattox, and C. Bailey-Kellogg, "Attention please: Modeling global and local context in glycan structure-function relationships," *bioRxiv*, pp. 1–19, 2021, doi: [10.1101/2021.10.15.464532](https://doi.org/10.1101/2021.10.15.464532).
- [8] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.
- [9] X. Li, Y. Xin, C. Zhao, Y. Yang, and Y. Chen, "Graph convolutional networks for privacy metrics in online social networks," *Appl. Sci.*, vol. 10, no. 4, p. 1327, Feb. 2020.
- [10] V. Gligorijević, P. D. Renfrew, T. Kosciolk, J. K. Leman, D. Berenberg, T. Vatanen, C. Chandler, B. C. Taylor, I. M. Fisk, H. Vlamakis, R. J. Xavier, R. Knight, K. Cho, and R. Bonneau, "Structure-based protein function prediction using graph convolutional networks," *Nature Commun.*, vol. 12, no. 1, p. 3168, May 2021.
- [11] A. Kapoor, X. Ben, L. Liu, B. Perozzi, M. Barnes, M. Blais, and S. O'Banion, "Examining COVID-19 forecasting using spatio-temporal graph neural networks," 2020, *arXiv:2007.03113*.
- [12] T. Nguyen, G. T. T. Nguyen, T. Nguyen, and D. Le, "Graph convolutional networks for drug response prediction," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 19, no. 1, pp. 146–154, Jan. 2022.
- [13] K. Liu, X. Sun, L. Jia, J. Ma, H. Xing, J. Wu, H. Gao, Y. Sun, F. Boulnois, and J. Fan, "Chemi-net: A molecular graph convolutional network for accurate drug property prediction," *Int. J. Mol. Sci.*, vol. 20, no. 14, p. 3389, Jul. 2019.
- [14] R. Joeres, D. Bojar, and O. V. Kalinina, "GlyLES: Grammar-based parsing of Glycans from IUPAC-condensed to SMILES," *J. Cheminformatics*, vol. 15, no. 1, pp. 1–11, 2023.
- [15] E. Fernández-de Gortari, C. R. García-Jacas, K. Martínez-Mayorga, and J. L. Medina-Franco, "Database fingerprint (DFP): An approach to represent molecular databases," *J. Cheminformatics*, vol. 9, no. 1, pp. 1–9, Dec. 2017.
- [16] H. L. Morgan, "The generation of a unique machine description for chemical structures—A technique developed at chemical abstracts service," *J. Chem. Document.*, vol. 5, no. 2, pp. 107–113, May 1965.
- [17] G. Landrum. *RDKit: Open-Source Cheminformatics*. Accessed: Jan. 9, 2023. [Online]. Available: <http://www.rdkit.org/>, doi: [10.5281/zenodo.591637](https://doi.org/10.5281/zenodo.591637).
- [18] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang, "Deep graph library: A graph-centric, highly-performant package for graph neural networks," 2019, *arXiv:1909.01315*.
- [19] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
- [20] Y. Wang, J. Wang, Z. Cao, and A. B. Farimani, "Molecular contrastive learning of representations via graph neural networks," *Nature Mach. Intell.*, vol. 4, no. 3, pp. 279–287, Mar. 2022.
- [21] P. Reiser, M. Neubert, A. Eberhard, L. Torresi, C. Zhou, C. Shao, H. Metni, C. van Hoesel, H. Schopmans, T. Sommer, and P. Friederich, "Graph neural networks for materials science and chemistry," *Commun. Mater.*, vol. 3, no. 1, p. 93, Nov. 2022.
- [22] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
- [23] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 3844–3852.
- [24] M. Cheung, J. Shi, L. Jiang, O. Wright, and J. M. F. Moura, "Pooling in graph convolutional neural networks," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019, pp. 462–466.
- [25] A. Alkuhlani, W. Gad, M. Roushdy, M. G. Voskoglou, and A.-B.-M. Salem, "PTG-PLM: Predicting post-translational glycosylation and glycation sites using protein language models and deep learning," *Axioms*, vol. 11, no. 9, p. 469, Sep. 2022.
- [26] M. Zidan, F. University, I. Elhenawy, A. Abas, and M. Othman, "Textual emotion detection approaches: A survey," *Future Comput. Informat. J.*, vol. 7, no. 1, pp. 32–58, Jun. 2022.
- [27] N. Q. K. Le, Q.-T. Ho, V.-N. Nguyen, and J.-S. Chang, "BERT-promoter: An improved sequence-based predictor of DNA promoter using BERT pre-trained model and SHAP feature selection," *Comput. Biol. Chem.*, vol. 99, Aug. 2022, Art. no. 107732.



He is currently a Teaching Assistant with the Faculty of Computer and Information Technology, Sana'a University. His research interests include artificial intelligence, data mining, and bioinformatics.



clustering around medoids techniques. Her master's dissertation was titled "Text Clustering Based on Semantic Measures." The work was done jointly between the Faculty of Computers and Information Sciences, Ain Shams University, and the University of Waterloo. She is currently a Professor with the Faculty of Computers and Information Sciences. She is the author of several publications. Her current research interests include data science, semantic web and machine learning, data warehouse, and big data analytics.



Appreciation Award in Technological Sciences, in 2018.



involved in more than 700 international conferences and workshops as a keynote and plenary speaker. His research interests include intelligent computing, artificial intelligence, biomedical informatics, big data analytics, intelligent education, smart learning systems, information mining, knowledge engineering, and biometrics. He was a member of program committees, a workshop/invited session organizer, the session chair, and tutorials. In addition, he was a member of many international societies and a member of the editorial board of 70 international and national journals. Also, he is a member of many international scientific societies and associations elected members of Euro Mediterranean Academy of Arts and Sciences, Greece. He is a member of Alma Mater Europaea of the European Academy of Sciences and Arts, Belgrade, and the European Academy of Sciences and Arts, Austria.

...