IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 11, 2022, accepted January 23, 2022, date of publication January 26, 2022, date of current version February 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3146395

PUStackNGly: Positive-Unlabeled and Stacking Learning for N-Linked Glycosylation Site Prediction

ALHASAN ALKUHLANI[1,2], WALAA GAD[2], MOHAMED ROUSHDY[3], AND ABDEL-BADEEH M. SALEM[2]

[1]Faculty of Computer and Information Technology, Sana'a University, Sana'a, Yemen
[2]Faculty of Computer and Information Science, Ain Shams University, Cairo 11566, Egypt
[3]Faculty of Computers and Information Technology, Future University in Egypt, New Cairo 11835, Egypt

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

ABSTRACT N-linked glycosylation is one of the most common protein post-translation modifications (PTMs) in humans where the Asparagine (N) amino acid of the protein is attached to the glycan. It is involved in most biological processes and associated with various human diseases as diabetes, cancer, coronavirus, influenza, and Alzheimer's. Accordingly, identifying N-linked glycosylation sites will be beneficial to understanding the system and mechanism of glycosylation. Due to the experimental challenges of glycosylation site identification, machine learning becomes very important to predict the glycosylation sites. This paper proposes a novel N-linked glycosylation predictor based on bagging positive-unlabeled (PU) learning and stacking ensemble machine learning (PUStackNGly). In the proposed PUStackNGly, comprehensive sequence and structural-based features are extracted using different feature extraction descriptors. Then, ensemble-based feature selection is employed to select the most significant and stable features. The ensemble bagging PU learning selects the reliable negative samples from the unlabeled samples using four supervised learning methods (support vector machines, random forest, logistic regression, and XGBoost). Then, stacking ensemble learning is applied using four base classifiers: logistic regression, artificial neural networks, random forest, and support vector machine. The experiments results show that PUStackNGly has a promising predicting performance compared to supervised learning methods. Furthermore, the proposed PUStackNgly outperforms the existing N-linked glycosylation prediction tools on an independent dataset with 95.11% accuracy, 100% recall 80.7% precision, 89.32% F1 score, 96.93% AUC, and 0.87 MCC.

INDEX TERMS Glycosylation, glycosylation sites prediction, machine learning, positive-unlabeled learning, stacking ensemble learning.

I. INTRODUCTION

Glycosylation of protein is one of the most common and important post-translation modifications processes (PTM) in most living organisms. It is estimated that more than fifty percent of human proteins are glycosylated [1]. It affects of various biological processes such as immune response, protein folding, signaling, and antigen recognition. Additionally, it is publicized that glycosylation is related to of various human diseases as diabetes, cancer, coronavirus, influenza, and Alzheimer's [2]–[5]. Glycosylation is categorized into four different types based on the identity of the atom of the protein's amino acid which binds the glycan chain such as N-linked, O-linked, C-linked, or S-linked. The most public type of glycosylation is N-linked glycosylation. In N-linked glycosylation, the glycan (GlcNAc) is attached to a Nitrogen atom of Asparagine (Asn or N) amino acid of the protein. In particular, N-linked glycosylation usually happens in N-X-S/T (S: serine, T: threonine) sequons, and in some uncommon cases N-X-C (C: cysteine), where X is any protein amino acid except proline [1], [6], [7].

Glycosylation sites identification is fundamental for understanding the system and mechanism of glycosylation. The experimental methods (such as mass spectrometry) for detecting the glycosylation sites are difficult, expensive, and time-consuming [6], [8]. Therefore, the technical and

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

computational tools using machine learning are playing an increasingly essential role in glycosylation site prediction.

The last two decades have seen a growing trend in developing several intelligent and computational models to predict N-linked glycosylation sites using machine learning and statistical techniques. For example, NetNGlyc [9] tool utilizes neural networks for N-linked glycosylation prediction. EnsembleGly [10] uses ensemble support vector machines (SVM) to predict N-linked, O-linked, and C-linked glycosylation sites. EnsembleGly utilizes binary profile, physicochemical properties, and PSI-Blast profile feature extraction to encode their unbalanced datasets. Hamby and Hirst propose GPP [11] tool to predict N-linked and O-linked glycosylation sites using random forest classifier and based on structural features and pairwise sequence patterns of protein peptide. GlycoPP [12] uses SVM to predict N-linked and O-linked glycosylation sites in prokaryotic based on Binary encoding, amino acid composition (AAC), position-specific scoring matrix (PSSM), secondary structure (SS), and accessible surface area (ASA) features.

NGlycPred [13] utilizes pattern and structural-based features; and uses a random forest classifier to predict N-linked glycosylation sites. GlycoEP ''in silico'' [7] tool employs SVM machine learning technique to predict N-linked, O-linked and C-linked glycosylation sites on large Eukaryotic dataset based on Binary encoding, AAC, PSSM, SS, and ASA features. GlycoMine [8] tool uses random forest (RF) classifier for N-linked and O-linked glycosylation site prediction based on heterogeneous functional and sequence-based features and using information gain and information gain (IG) and minimum redundancy maximum relevance (mRMR) methods for feature selection. GlycoMinestruct [14] combines sequence and structural features for predicting N-linked and O-linked glycosylation sites using the random forest for classification and linear SVM for feature selection. Akmal *et al.* [15] presented a comprehensive technique for prediction N-linked glycosylation sites using artificial neural networks based on position relative and statistical moments.

SPRINT-Gly [2] uses deep neural networks and SVM machine learning methods to identify N-linked and O-linked glycosylation sites on large datasets extracted from six human and mouse databases based on various sequence, profile and structural-based features. N-GlyDE [1] is a two-stage prediction tool for N-linked glycosylation site prediction which uses a similarity voting algorithm and SVM method based on Gapped dipeptide, SS, and ASA features. GlycoMine_PU [6] uses positive unlabeled (PU) learning technique to predict N-linked, O-linked, and C-linked glycosylation sites and mRMR for feature selection based on many sequence, profile and structural-based features. The authors in GlycoMine_PU found that their model outperforms RF, SVM, and one-class learner. N-GlycoGo [16] uses XGBoost, an ensemble machine learning model, for N-linked glycosylation site prediction. T-test and mRMR were used for feature selection based on eleven feature encoding approaches.

However, the success in developing useful approaches for N-linked glycosylation site prediction, many shortcomings still exist [17] as follows: 1) most of the existing studies use a small number of extracted features, which are not completed or comprehensive. Usually, utilizing more comprehensive features can produce better prediction performance results. 2) In the feature selection step, the existing studies do not consider the robustness and stability of employed feature selection methods. 3) the existing studies consider the unlabeled samples as negative sites. However, they could belong to whether the positive or negative samples.

Therefore, a novel N-linked glycosylation predictor is proposed based on bagging positive-unlabeled learning and stacking ensemble machine learning (PUStackNGly). It addresses the previous shortcomings and problems to improve the prediction performance for the N-linked glycosylation site. The proposed PUStackNGly is based on PU learning and stacking ensemble machine learning techniques. The major aims and contributions of PUStackNGly are summarized as follows: 1) integrating comprehensive sequence, profile, and structured-based features where forty-five feature extraction methods are employed to encode the protein peptides (samples). 2) applying stable and robust ensemble-based feature selection technique to select significant features for prediction purposes. 3) Using ensemble bagging PU learning to select reliable negative samples among the unlabeled samples. Four supervised learning methods (SVM, Logistic Regression (LR), RF, XGBoost) are used in this step. 4) based on ensemble-based learning, the stacking ensemble learning is presented to construct the prediction model. In this step, six classifiers are involved including LR, RF, ANN, XGB, SVM, and KNN. The final selected ensemble stacking model is constructed by four base classifiers including LR, RF, ANN, SVM because of their high performance on the development dataset. The proposed model is evaluated using accuracy, precision, F1 score, AUC, and MCC performance measures. PUStackNGly achieved significant improvement on the independent test over the existing tools with 95.11% accuracy, 100% recall, 80.7% precision, 89.32% F1 score, 96.93% AUC, and 0.87 MCC.

The remaining of the paper is organized as follows: Section II describes the material and methods in detail starting from dataset extraction to the prediction model and evaluation. The experimental performance results and discussion are presented in Section III. Section IV presents the conclusions and future works.

II. MATERIALS AND METHODS

The general framework of PUStackNGly is shown in Fig. 1. PUStackNGly consists of six main steps: data extraction and preprocessing, feature extraction, feature selection, PU Learning, stacking ensemble learning, and model evaluation. In the first step, data is collected from the UniProt database and preprocessing is done. Secondly, comprehensive numerical features are extracted for each sample. Then, informative and stable features are selected

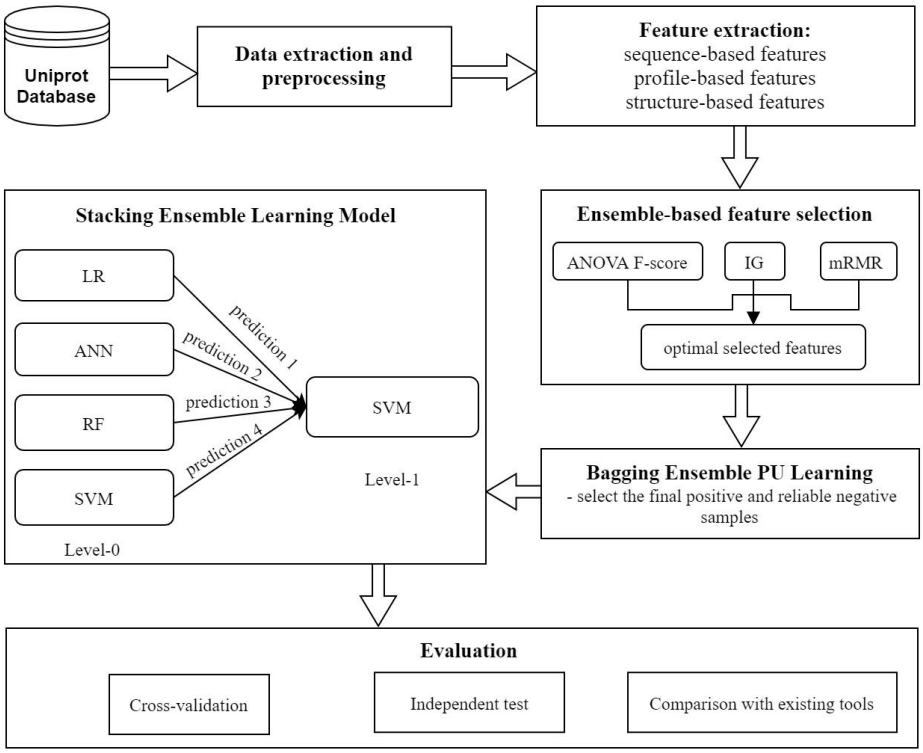

FIGURE 1. The general framework of PUStackNGly.

using an ensemble-based feature selection strategy based on Redundancy Maximal Relevance (mRMR), Analysis of Variance (ANOVA) F-score, and Information Gain (IG) feature selection methods. In the fourth step, ensemble bagging PU learning selects reliable negative samples based on SVM, LR, RF, and XGBoost machine learning classifiers. Then, the stacking ensemble-based learning model is constructed for N-linked glycosylation site prediction which integrates LR, RF, ANN, SVM base classifiers. Finally, the proposed model is evaluated using accuracy, precision, F1 score, AUC, and MCC performance measures.

A. DATA EXTRACTION AND PREPROCESSING

In this paper, UniProt [18] (https://www.uniprot.org/) ver. 202102 is used as a data source. The UniProt database is a comprehensive accessible database for functional information of proteins and their sequences. To gain the human experimentally verified N-linked glycoproteins, multiple criteria are used in the advanced search. For instance, to get the experimentally verified N-linked glycoproteins, the term ''n-linked'' is added in the ''PTM/Processing > Glycosylation [FT]'' field and ''any experimental assertion'' in the ''Evidence'' field. In addition, the searched glycoproteins must be ''reviewed'' and ''Homo sapiens''. As a result, 1086 human glycoprotein sequences are obtained initially. After that, the CD-HIT [19] tool was used to remove glycoproteins with similarities over 30%. Accordingly, glycoproteins are reduced to 819 after removing redundancy.

Each asparagine (N) amino acid in the glycoprotein sequence is considered an N-linked glycosylation site. These sites are either positive or negative. The experimentally verified sites are considered as positive sites with taking into consideration that the sites with evidence ''Probable'', ''Potential'', ''By similarity'' or ''Sequence analysis'' were excluded. Regarding the negative sites, most of the previous studies [1], [2], [7], [8], [10]–[12], [16] considered sites that are not experimentally verified (unlabeled sites) as negative sites. However, these sites may be positive. Accordingly, positive-unlabeled learning is used in this work to select reliable negative sites from unlabeled sites.

Each N-linked glycosylation site is represented by a fragment sequence, called a peptide, using a sliding window strategy. Different peptide sizes have been experimented and the optimal size was 25. So, a peptide Q with a window size of 25 is generated for each candidate N-linked glycosylation site. The candidate site is positioned at the center and surrounded by 12 residues from the left and 12 residues from the right for each Q which is represented as:

$$Q = a_1 a_2 \ldots a_{13} \ldots a_{24} a_{25} \quad (1)$$

where a_i is the ith residue in the peptide sequence Q and a_{13} represent the N-linked glycosylation site. CD-HIT is also used to remove redundancy from positive and unlabeled peptides separately with 30% identity to avoid prediction overfitting. The number of glycoproteins and peptides before and after removing redundancy is shown in Table 1.

TABLE 1. Number of glycoproteins and peptides before and after removing redundancy.

	initial	after redundancy removal
glycoproteins	1086	825
positive samples	2884	1989
unlabeled samples	18168	13737

The dataset is divided into three sets including training, development, and independent sets. The independent set is used for final evaluation and comparison with the existing tools. The training set is used for feature selection, PU learning, and constructing the final ensemble model. In contrast, the development set is utilized for experimental and optimizing the prediction model. More details about data separation are described in subsection III-A.

B. FEATURE EXTRACTION

Each peptide (sample) is a sub-part of protein sequence which is represented by 20-character amino acid {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}. To build an effective prediction model, this sequence should be transformed into a numerical representation. The numerical representation echoes the key feature information that is hidden in the peptide sequence. In this work, comprehensive features are extracted to encode each peptide. The IFeature Python package [20] was used to extract these features. They have been categorized into eight groups: sequence-based features are the first six groups, profile-based features are the seventh group, and structure-based features are the last group. Each group contains a set of descriptor or feature extraction methods.

The first group is sequence-based features and is related to the amino acid composition which includes: 1) amino acid composition (AAC) [21], 2) Enhanced amino acid composition (EAAC) [20], 3) Composition of k-spaced amino acid pairs (CKSAAP) [22], 4) Dipeptide deviation from the expected mean (DDE) [23], 5) Grouped amino acid composition (GAAC), 6) Enhanced grouped amino acid composition (GEAAC), 7) Composition of k-spaced amino acid group pairs (CKSAAGP), 8) Grouped dipeptide composition (GDPC), 9) Grouped tripeptide composition (GTPC) [20], 10) Pseudo-amino acid composition (PAAC), 11) Amphiphilic PAAC (APAAC) [24], and 12) Pseudo K-tuple reduced amino acids composition (PseKRAAC type1 to type16) [25].

The second group ''Binary profile'' includes a 20 binary-valued vector (0/1-based scheme) for each amino acid in a peptide [2], [12]. The third group ''blosum62 profile'' encodes each amino acid in a peptide by its corresponding representation in the Blosum62 matrix [26]. The fourth group ''Autocorrelation'' calculates the correlation between two amino acids index in a peptide according to their related physicochemical properties in the AAindex database [27] and it includes 1) Moran, 2) Geary, and 3) Normalized Moreau-Broto (NMBroto) [20], [28]. The fifth group ''Composition-Transition-Distribution (C/T/D)'' represents the distribution of amino acids inside the peptides based on the physicochemical property and it includes: 1) Composition (CTDC), 2) Transition (CTDT), and 3) Distribution (CTDD) [29]. The sixth group ''conjoint triad'' clusters the amino acids of a peptide sequence into groups based on their side chains volumes and dipoles and it includes: 1) Conjoint triad (CTriad) and 2) k-Spaced Conjoint Triad (KSCTriad) [20], [24].

The seventh group ''Position-specific scoring matrix (PSSM) profile'' is evolutionary information calculated by the PSI_BLAST public tool to align each peptide against a global database (human SWISSPROT database in our work) [30]. The eighth group contains three structure-based features. The first two descriptors are Secondary structure elements content (SSEC) and Secondary structure elements binary (SSEB) which require PSIPRED software [31] to predict Secondary Structure for each peptide [1], [2], [8]. The third descriptor of the last group is accessible surface area (ASA) which identifies each peptide's nature and basic structure [1], [12]. SPINE-X [32] software is used to calculate ASA for each peptide. Consequently, the total number of features extracted from all descriptor groups is 7311 features. The groups and their descriptors with feature counts are mentioned in Table 2.

C. FEATURE SELECTION

The heterogeneous and large number of extracted features may contain irrelevant and noisy features. Therefore, feature selection should be applied to select significant and relevant features for the prediction model [8], [33]. In addition, the stability and robustness of feature selection methods should be considered to avoid classification overfitting and to get high prediction accuracy. One of the best methods to improve the stability and robustness of feature selection is the ensemble feature selection technique [33]–[35]. In ensemble feature selection, multiple feature selection techniques are combined to provide robust selected features and consequently best prediction results. The PUStackNGly combines three filtering feature selection techniques: Redundancy Maximal Relevance (mRMR), Analysis of Variance (ANOVA) F-score, and Information Gain (IG).

The mRMR feature selection method ranks the features based on their correlation to the target class and minimum redundancy between features [36]. Features that have the best trade-off between minimum redundancy and maximum relevance to target class are counted as significant features. mRMR assesses the redundancy and relevance of two features, x and z based on mutual information $I(x, z)$, which is represented as:

$$I(x,z) = \int\int p(x,z) \log \frac{p(x,z)}{p(x)p(z)} dxdz \tag{2}$$

where $p(x)$ and $p(z)$ are the probability density functions of x and z, respectively and $p(x, z)$ is the joint probability density function of x and z.

The Analysis of Variance (ANOVA) feature selection method is a type of f-statistic hypothesis test. It measures either the differences between means of two or more samples data exist or calculates the ratio between variances of two or more groups [37]. The ANOVA f-score compares the ratio of between-group variability to within-group variability, which is represented as:

$$F = \frac{\sum_{i=1}^{G} n_i (M_i - M)^2 / (G-1)}{\sum_{i=1}^{G} \sum_{j=1}^{n_i} (S_{ij} - M_i)^2 / (N-G)} \quad (3)$$

where G is the number of groups, n_i denotes the number of samples in the i^{th} group, M_i is the samples mean in the i^{th} group, M denotes the whole mean of the data, S_{ij} is the observation j in the i^{th} group, and N is the whole number of samples.

The Information Gain (IG) is a statistical feature selection method based on entropy information theory. It measures the feature weights according to the relationships between features and target classes [8], [38]. The entropy of sample data is calculated as:

$$Entropy = -\sum_{i=1}^{n} p(c_i) \log p(c_i) \quad (4)$$

where n is the number of data classes, and $p(c_i)$ denotes the probability of the i^{th} class.

The following steps explain the combination of the three feature selection methods to select the top optimal features using ensemble-feature selection:

1) Subset S_1: the top 500 features of all the extracted features using mRMR.
2) Subset S_2: the top 500 features from all the extracted features using ANOVA f_score.
3) Subset S_3: the top 500 features from all the extracted features using IG.
4) Set S: the final optimal selected features by calculating the intersection between S_1, S_2, and S_3 feature subsets.

The final number of selected features is 102 features. Table 2 shows the feature extraction groups and their descriptors with the number of features before and after the ensemble feature selection.

D. POSITIVE-UNLABELED (PU) LEARNING

PU learning is a semi-supervised machine learning, where some samples are labeled as positive and the remaining as unlabeled. The hypothesis is that each unlabeled sample may belong to either a positive or negative sample [39]. One of the PU learning methods is bagging PU learning. The goal of bagging PU learning is to find the score of each sample in the unlabeled samples where the samples with the lowest scores are considered negative samples. Accordingly, the learning model is trained on reliable negative and positive samples [39], [40]. The used dataset consists of positive (glycosylated) samples and vast unlabeled samples. So, the bagging PU learning is applied to construct the dataset from positive and reliable negative samples. This dataset is employed for training the PUStackNgly model. The proposed model adapts to ensemble bagging PU learning by combining multiple bagging supervised learning algorithms: SVM, LR, RF, XGBoost classifiers.

The ensemble bagging PU learning steps are shown in Algorithm 1. The positive and unlabeled samples are represented as P and U respectively. Firstly, balanced training data is created via merging P with random samples U_s from the U with size L (the same number of samples in P) using bootstrap sampling with replacement. The subset U_s that is selected in the balanced training data is considered as negative samples during the training process. For each classifier c, the predictor $f_c(x, t)$ is built and trained. For each sample x in the U, an initial score $S_c(x)$ and a counter $n(x)$ are assigned to 0. Then, the probability prediction is applied using the training model on the samples in U that are not included in the training process. The resulted probability prediction for a sample x is added to the $S_c(x)$ and one is added to $n(x)$. This process is repeated T times. It is assumed that $T = 500$. After that, the $S_c(x)$ for each sample x in U is calculated as $S_c(x)/n(x)$. The ensemble voting by averaging the $S_c(x)$ where c in (SVM, LR, RF, XGBoost) is computed to provide the last score S_x for each sample in U. Finally, the training dataset is constructed from the positive samples P and the negative samples that are picked from U that have the lowest $S(x)$ with the same number of P. This dataset will be the input for the PUStackNGly, N-linked glycosylation site predictor.

Algorithm 1 Ensemble Bagging PU Learning

INPUT: Positive samples, P
Unlabeled samples, U
L = the size of P
T = the number of bootstraps or iterations
$c \in$ {SVM, LR, RF, XGBoost}
OUTPUT: score $S(x)\ \forall\ x \in U$
Initialize $S_c(x) = 0, n(x) = 0\ \forall\ x \in U$
for $dot = 1$ to T
  draw a subsample U_s from U with size L.
  train predictor $f_c(x,t)$ to discriminate P against U_s
  $\forall\ x \in (U \setminus U_s)$ update:
  $S_c(x) = S_c(x) + f_c(x,t)$,
  $n(x) = n(x)+1$
end for
$\forall\ x \in U$ update:
$S_c(x) = \frac{S_c(x)}{n(x)}$,
$S(x) = \frac{\sum_{c=1}^{4} S_c(x)}{4}$
Return $S(x)$

E. STACKING ENSEMBLE LEARNING

Recently, there has been an increasing interest in ensemble machine learning which shows predictive capability in many applications [41]–[45]. In ensemble learning, multiple base predictors are constructed where their results are integrated with a specific strategy to fetch the final results.

TABLE 2. The feature extraction descriptors with number of features before and after feature selection.

Descriptor group	Descriptor	NO. of features	NO. of selected features
Amino acid composition group	AAC	20	0
	EAAC	420	18
	CKSAAP	2400	2
	DDE	400	1
	GAAC	5	0
	GEAAC	105	7
	CKSAAGP	150	1
	GDPC	25	0
	GTPC	125	2
	PAAC	44	2
	APAAC	68	0
	PseKRAAC type1 to type16	457	13
Binary	BINARY	480	5
BLOSUM62	BLOSUM62	480	13
Autocorrelation	Moran	192	2
	Geary	192	3
	NMBroto	192	2
C/T/D	CTDC	39	2
	CTDT	39	0
	CTDD	195	6
Conjoint triad	Ctriad	343	3
	KSCTriad	343	3
Position-specific scoring matrix (PSSM)	PSSM	500	16
Structure-based features	SSEC	3	0
	SSEB	69	0
	ASA	25	1
Sum		7311	102

It usually reveals better prediction stability and capability than a single predictor model [41], [42]. Stacking ensemble learning gets the results from multiple models and combines them into a new model. Stacking ensemble learning consists of two levels of predictors. In the first level or level-0 called base model, multiple base models are implemented individually. In the second level or level-1 called meta-model, one meta-model is implemented on the probability results of the first level predictors to decrease the generalization error and provide the final prediction result [41], [42]. Figure 2 shows the framework of the stacking ensemble learning.

To find the best classifiers for level-0 and level-1 of the stacking ensemble learning, six supervised learning algorithms were adapted:

1) Support Vector Machine (SVM):
SVM is a state-of-the-art classifier originally developed by Vapnik and Cortes [46]. SVM has been widely used in computational biology and bioinformatics [2]. It classifies by finding the optimal separator hyperplane between two classes. Parameter configuration for SVM, such as kernel, C, and gamma, is important to adjust and optimize SVM. Finding the optimal parameter for SVM is clarified in subsection III-B.

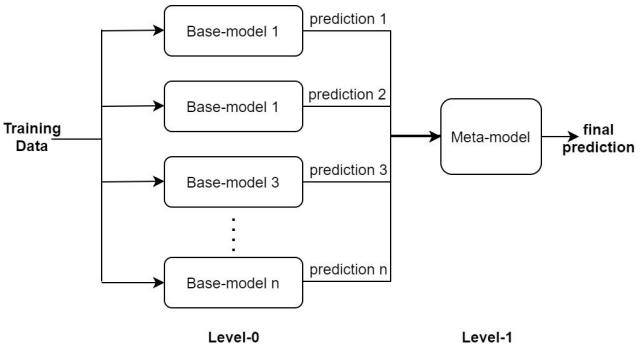

FIGURE 2. Framework of stacking ensemble learning.

2) Logistic Regression (LR):
LR is a popular classification method in clinical research because the dependent outcome is discrete e.g., positive/negative. LR classifies by measuring the probability of a discrete binary class such as glycosylated/non-glycosylated in our study [47].
3) Artificial Neural Networks (ANN):
ANN is a classification method that is widely used in many applications. Similar to the human brain, ANN learns from experience. It consists of fully connected

layers where each layer has multiple units or neurons. The general ANN architecture usually contains three basic layers: input, hidden, and output layer. ANN prediction model receives the labeled input data through the input layer which is connected to the next layer by weighted links. During the training process, ANN regulates the link weights to improve prediction performance [48].

4) Random Forest (RF):
RF [49] is an ensemble supervised learning method which is simple and suitable for high-dimension data. It is constructed from a multitude of decision trees where each decision tree contains multiple nodes and paths. Each node has rules to choose the direction between two or more paths. The final result of RF is obtained by integrating the results of decision trees [11].
5) XGBoost (XGB):
XGB [50] is an ensemble learning classification method that uses a tree boosting framework. Both Gradient Boosting Machine (GBM) and XGB are ensemble tree algorithms that implement boosting learners using the gradient descent technique. However, XGB improves the performance and avoids overfitting by systems optimization and parameter tuning which is clarified in subsection III-B.
6) K-nearest neighbors (KNN):
KNN [51] is a nonparametric simple supervised learning method that is used for classification and regression. In KNN, K is the number of nearest samples closest to the target point in the feature space. KNN classifies new samples by measuring the similarity between the new cases and the labeled cases. Then, it puts them in a class that is more similar to the available classes [52].

F. EVALUATION

The proposed PUStackGly is evaluated and assisted using accuracy (AC), recall, precision, F1 score, and Matthews correlation coefficient (MCC) performance measures which are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$F1 = \frac{2 \times recall \times precision}{recall \times precision} \tag{8}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{9}$$

where TP (True Positive) is the number of N-linked glycosylation sites that are predicted correctly. FP (False Positive) is the number of N-linked glycosylation sites that are predicted incorrectly. TN (True Negative) is the number of N-linked non-glycosylation sites that are predicted correctly. FN (False Negative) is the number of N-linked non-glycosylation sites that are predicted incorrectly. In addition, AUC, the area under the receiver operating characteristic curve (ROC), measures the predictor ability for separating data of the two classes by plotting the TP rate against the FP rate.

III. RESULTS AND DISCUSSION

In this section, we present and discuss the achieved experimental results. The data extraction, preprocessing, and samples construction are implemented using R 4.0.3 while the remaining stages of PUStackNGly are implemented in Python 3.8.5. The source code and data of PUStackNGly are available online at (https://github.com/Alhasanalkuhlani/PUStackNGly).

A. DATA PREPARATION

The dataset of 825 experimentally verified glycoproteins was used. These glycoproteins contain 1989 N-linked glycosylation samples and 13737 N-linked non-glycosylation samples after redundancy removal. This dataset is divided into three sets including independent, training, and development sets. The first set is an independent set that contains six glycoproteins with 46 positive N-linked glycosylation sites and 179 nonpositive N-linked glycosylation sites. The independent set is used for evaluating the PUStackNGly predictor and comparing it with the existing N-linked glycosylation site tools. The second dataset is a 1:1 balanced dataset used for training and constructing the PUStackNGly predictor, which contains 1651 positive samples and 1651 randomly selected reliable negative samples resulting from the PU learning stage. The third dataset is the development set which includes 292 positive samples and 2034 nonpositive samples. The development dataset is not involved in the feature selection and PU learning stage. It is used for evaluating and optimizing learning models that are integrated into the PUStackNGly. In each evaluation process using the development set, the implementation was repeated fifty times on a balanced dataset 1:1 that is randomly constructed from the development dataset and the average performance results were always calculated.

B. PARAMETER SETTING

In the PU and stacking learning models, six supervised learning methods were used including SVM, ANN, LR, RF, KNN, and XGB. To find the optimal parameters for these classifiers, classifiers' parameters are tuned depending on their performance results on the development dataset. Table 3 shows the tuned parameters for the six classifiers and the optimal value for each parameter. SVM, ANN, LR, RF, and KNN are implemented using the Scikit-learn Python library [53], while XGB is implemented using the XGBoost Python library [50].

TABLE 3. Tuned parameter setting for stacking ensemble classifiers and the optimal value for each parameter.

classifier	parameter	range of values	optimal value
SVM	kernal	linear, poly, rbf, sigmoid	rbf
	C	1e-2, 1e-1, . . . , 1e6	100
	Gama	1e-8, 1e-7, ...,1	1e-4
ANN	hidden units	10 to 100	40
	learning rate	1e-2,1e-3,. . . ,1e-6	1e-2
LR	Penalty	L1, L2	L2
RF	n_estimators	100 to 1000	100
KNN	Algorithm	ball_tree, kd_tree, brute	kd_tree
	n_neighbors (K)	2 to 9	2
XGB	booster	gbtree, gblinear	gbtree
	learning_rate	0.01, 0.02, . . . , 0.1	0.5
	max_depth	3 to 10	3
	min_child_weight	1 to 10	5
	Subsample	0.3, 0.4, . . . , 0.9	0.9

TABLE 4. Performance results of the six classifiers on the development dataset.

Metric / Model	Accuracy	Recall	Precision	F1	AUC	MCC
LR	0.9699	**1**	0.9433	0.9708	0.9699	0.9415
ANN	0.9688	**1**	0.9414	0.9698	0.9688	0.9395
RF	0.9686	**1**	0.941	0.9696	0.9686	0.939
SVM	**0.9712**	0.9966	**0.9485**	**0.9719**	**0.9712**	**0.9436**
XGB	0.969	0.9966	0.9447	0.9699	0.969	0.9396
KNN	0.9676	0.9966	0.9422	0.9686	0.9676	0.9369

C. CONSTRUCTING PUStackNGly

To select the classifiers for level-0 and level-1 of the stacking ensemble learning, the performance of six machine learning models, SVM, ANN, LR, RF, KNN, and XGB, are evaluated on the development dataset. The evaluation process of these models is repeated fifty times on the balanced data selected from the development dataset and the averaged results are recorded. The six classifiers are optimized with the optimal parameters that are shown in Table 3. The performance results for these classifiers are shown in Table 4.

Table 4 shows that SVM achieves the highest performance results in the term of accuracy, precision, F1, AUC and MCC between the six classifiers. Therefore, SVM is selected as the meta or level-1 classifier for PUStackNGly.

Regarding the base or level-0 classifiers, the six classifiers in Table 4 show similar performance. Thus, we tried random ten combinations for stacking ensemble models constructed as four, five, or six classifiers representing the base level-0 classifiers, and SVM representing the meta level-1 classifier. The combinations of the base classifiers for the ten models are as follow:

- Stacking1: ANN, RF, SVM, and KNN
- Stacking2: LR, RF, SVM, and KNN
- Stacking3: LR, ANN, RF, and SVM
- Stacking4: LR, ANN, XBG, and SVM
- Stacking5: LR, RF, XBG, and SVM
- Stacking6: LR, ANN, RF, and XBG
- Stacking7: LR, ANN, RF, and KNN

TABLE 5. Comparison between ten stacking models with different combinations for base classifiers on the development dataset.

Metric / Model	Accuracy	Recall	Precision	F1	AUC	MCC
Stacking1	0.9724	1	0.9479	0.9732	0.9724	0.9463
Stacking2	0.9706	0.9966	0.9475	0.9714	0.9706	0.9425
Stacking3	**0.9727**	**1**	**0.9485**	**0.9735**	**0.9727**	**0.947**
Stacking4	0.9715	1	0.9462	0.9723	0.9715	0.9446
Stacking5	0.9696	0.9966	0.9458	0.9705	0.9696	0.9407
Stacking6	0.9721	1	0.9473	0.9729	0.9721	0.9458
Stacking7	0.9708	1	0.945	0.9717	0.9708	0.9433
Stacking8	0.9716	1	0.9464	0.9724	0.9716	0.9447
Stacking9	0.9724	1	0.9479	0.9732	0.9724	0.9463
Stacking10	0.9724	1	0.9479	0.9732	0.9724	0.9463

TABLE 6. Cross-validation performance results of PUStackNgly compared to LR, ANN, RF, SVM, XGM, and KNN classifiers on the training dataset.

Metric / Model	Accuracy	Recall	Precision	F1	AUC
LR	0.9962	**0.994**	0.9982	0.9962	0.9989
ANN	0.9955	**0.994**	0.9966	0.9955	0.999
RF	0.9945	0.9937	0.9954	0.9945	0.9988
SVM	0.9946	0.9904	0.9988	0.9946	0.9988
XGB	0.9941	0.9902	0.9979	0.994	0.999
KNN	0.9935	0.9924	0.9947	0.9935	0.9949
PUStackNgly	**0.9964**	0.9938	**0.9989**	**0.9963**	**0.999**

- Stacking8: LR, RF, SVM, ANN and KNN
- Stacking9: LR, SVM, ANN, KNN and XBG
- Stacking10: LR, RF, SVM, ANN, KNN and XBG

Performance result comparison between the ten stacking models is evaluated on the development dataset, which is shown in Table 5. Although the results show that all models achieve high-performance results, the third model (Stacking3) outperforms the other stacking models. Moreover, Stacking3 outperforms all the six classifiers when evaluated individually on the development dataset, as shown in Table 4 and Table 5 in the term of accuracy, recall, precision, F1, AUC, and MCC. Thus, the Stacking3 model, which includes LR, ANN, RF, and SVM as base classifiers and SVM as a meta-model, is selected as the final stacking model for PUStackNGly.

D. EVALUATION USING TRAINING AND DEVELOPMENT DATASET

Firstly, PUStackNgly is compared with the LR, ANN, RF, SVM, XGM, and KNN classifiers on the training dataset with repeated ten-fold cross-validation. Table 6 shows the performance results using accuracy, recall, precision, F1 score, AUC performance metrics. In Fig. 3, the boxplot clarifies the performance accuracy results for cross-validation. The table and figure demonstrate that the performance is high with all classifiers due to the ensemble-based feature selection technique and ensemble bagging PU learning. Moreover, the table and figure show that PUStackNGly improves the prediction results in the term of accuracy, precision, F1 and AUC.

Secondly, as clarified in subsection III-C, Table 4 showed the comparison performance results of six supervised classifiers: LR, ANN, RF, SVM, XGM, and KNN on the development dataset. The SVM performed better than the other

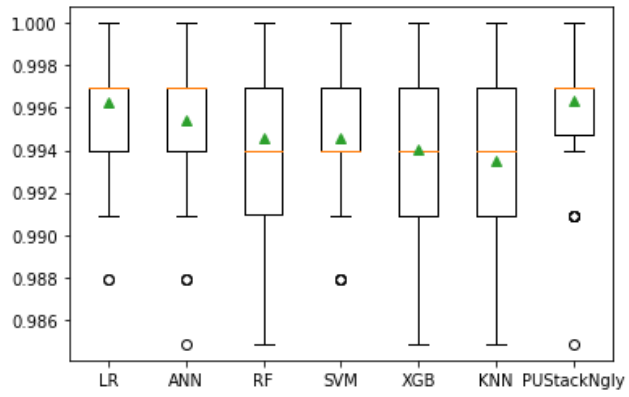

FIGURE 3. Boxplot for cross-validation accuracy results of the stacking ensemble learning compared to LR, ANN, RF, SVM, XGM, and KNN classifiers.

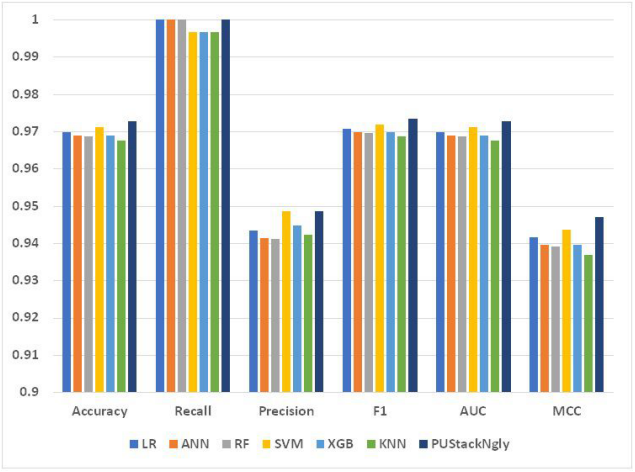

FIGURE 4. Performance results of PUStackNgly compared to LR, ANN, RF, SVM, XGM, and KNN classifiers on the development dataset.

classifiers in the term of accuracy, precision, F1, AUC, and MCC. Also, Tables 5 showed the comparison performance results of different six stacking models. The Stcking3 was selected to be the model of PUStackNGly due to its high performance. Based on Table 4 and 5, Fig. 4 clarifies the performance results of PUStackNGly compared to the optimized LR, ANN, RF, SVM, XGM, and KNN classifiers on the development dataset using Accuracy, Recall, Precision, F1, AUC, MCC metrics. The results demonstrate that the performance of all the compered models on the development dataset is as high as the performance results using cross-validation. However, PUStackNGly outperforms the LR, NN, RF, and KNN classifiers with accuracy (97.27%), precision (94.85%), F1 (97.35%), AUC (97.27%), and MCC (0.947).

E. COMPARISON WITH THE EXISTING TOOLS

In order to fairly evaluate the performance of PUStackNgly, the prediction performance of PUStackNgly is compared with the existing tools for N-linked glycosylation site prediction on the independent dataset. Since 2007, there have been about nine, sequence-based, N-linked glycosylation site predictors including EnsembleGly [10], GPP [11], GlycoPP [12], GlycoEP [7], GlycoMine [8], SPRINT-Gly [2], N-GlyDE [1], GlycoMine_PU [6], N-GlycoGo [16]. However, we faced connectivity and prediction issues with three predictors including EnsembleGly, N-GlyDE, N-GlycoGo. In contrast, with connectivity challenges in some cases, the other six predictors are available and usable. Therefore, the comparison on the independent dataset is done with the accessible six predictors including:

1) GPP (http://comp.chem.nottingham.ac.uk/glyco/) is a glycosylation site predictor using random forest machine learning method with a small dataset extracted from the OGLYCBASE dataset (http://www.cbs.dtu.dk/databases/OGLYCBASE/O-Unique.seq). GPP was predicted with 92.8% accuracy and 0.85 MCC on its training dataset.
2) GlycoPP (http://www.imtech.res.in/raghava/glycopp/) is a predictor for N- and O-glycosylation site in prokaryotic using SVM classifier. GlycoPP implements four N- glycosylation site prediction models based on the feature extraction methods including 1) Binary Profile of Pattern (GlycoPP_BPP), 2) Composition Profile of Patterns (GlycoPP_CPP), 3) PSSM Profile of Patterns (GlycoPP_PPP), and 4) GlycoPP_BPP+ASA. The GlycoPP_BPP+ASA achieved the best performance with (82.71% accuracy, 0.65 MCC) using cross-validation on the training dataset and (86.84% accuracy, 0.76 MCC) on the independent set from the four models. The average performance results of the four models of GlycoPP are reported for comparison with PUStackNGly.
3) GlycoEP (http://www.imtech.res.in/raghava/glycoep/) is an in-silico predictor for N-, O- and C-glycosylation site in Eukaryotic using SVM machine learning method. Similar to GlycoPP, GlycoEP predict N-linked glycosylation site by four models on standard dataset according to feature extraction method including 1) Binary Profile of Pattern (GlycoEP_BPP), 2) Composition Profile of Patterns (GlycoEP_CPP), 3) PSSM Profile of Patterns (GlycoEP_PPP), and 4) GlycoEP_BPP+ASA on the standard dataset containing 2604 N-linked site. GlycoEP achieved on its independent set 95.67% accuracy and 0.91 MCC. The average performance results of the four models of GlycoEP are reported for comparison with PUStackNGly.
4) GlycoMine (https://glycomine.erc.monash.edu/Lab/GlycoMine/#webserver) is a predictor for N-, O-, C-linked glycosylation site using RF classifier. Mrmr, IG, and IFS feature selection were employed for GlycoMine. It used protein functional features, local sequence features, structural features, and functional annotations for feature extraction. GlycoMine achieved 95% accuracy and 0.95 MCC using cross-validation on its training dataset. In addition, it achieved 95.6% accuracy and 0.90 MCC on its independent dataset.
5) GlycoMine_PU (https://glycomine.erc.monash.edu/Lab/GlycoMine_PU/) is N-, O-, and C-linked glycosylation site predictor using PU learning method. Six

TABLE 7. Performance results of PUStackNGly compared with SPRINT-GLY, GlycoMine_PU, GlycoMine, GlycoEP, GlycoPP, and GPP tools on the independent dataset on the term of accuracy, recall, precision, F1, AUC, and MCC.

Metric / Model	Accuracy	Recall	Precision	F1	AUC	MCC
GPP	0.7333	1	0.434	0.6053	0.8324	0.5371
GlycoPP	0.6578	0.8913	0.3628	0.5157	0.7445	0.3945
GlycoEP	0.7956	0.9565	0.5	0.6567	0.8554	0.5874
GlycoMine	0.4489	0.4565	0.175	0.253	0.4517	-0.0781
GlycoMine_PU	0.9378	1	0.7667	0.8679	0.9609	0.8407
SPRINT-Gly	0.9422	1	0.7797	0.8762	0.9637	0.8503
PUStackNGly	**0.9511**	**1**	**0.807**	**0.8932**	**0.9693**	**0.8703**

groups of sequence-based feature extraction methods were employed to encode samples and mRMR was implemented to feature selection. GlycoMine_PU achieved 88.6% accuracy and 92.7% AUC using cross-validation on its training dataset. In addition, it achieved 81.5% accuracy and 89.3% AUC on its independent dataset.

6) SPRINT-Gly (https://sparks-lab.org/server/sprint-gly/) is an N-, O-linked glycosylation site predictor based on deep neural networks and SVM machine learning methods using human and mouse datasets. It utilized seven sequence and structural-based feature extraction methods to encode protein-peptide and forward feature selection to select relevant features. SPRINT-Gly achieved 93.8% accuracy and 0.81 MCC using cross-validation on its human training dataset. In addition, it achieved 97.8% accuracy and 0.939 MCC on its independent human dataset.

To compare the performance results of our proposed PUStackNGly with these six tools, the independent dataset is submitted to these tools as well as to the PUStackNGly. Table 7 shows the predicted performance results for PUStackNGly compared with the six tools in the term of accuracy, recall, precision, F1, AUC, and MCC performance measures. In addition, Fig. 5 illustrates the comparison of PUStackNGly with SPRINT-GLY, GlycoMine_PU, GlycoMine, GlycoEP, GlycoPP, and GPP tools in the term of accuracy, precision, F1, and MCC. From Table and figure, PUStackNGly achieved 95.11% accuracy, 80.7% precision, and 0.87 MCC a significant improvement when compared to GPP (73.33% accuracy, 40.4% precision, and 0.5371 MCC), GlycoPP (65.78% accuracy, 36.28% precision and 0.3945 MCC), GlycoEP (79.56% accuracy, 50% precision and 0.5874 MCC), GlycoMine (44.89% accuracy, 17.5% precision and -0.0781 MCC), GlycoMine_PU (93.78% accuracy, 76.67% precision and 0.8407 MCC), or SPRINT-Gly (94.22% accuracy, 77.97% precision and 0.8503 MCC). It is also observed that the true positive rate (recall) is 100% for GPP, GlycoMine_PU, SPRINT-Gly, and PUStackNGly which means that all positive samples are predicted correctly by these tools.

Moreover, the AUC roc curve for PUStackNGly, SPRINT-GLY, GlycoMine_PU, GlycoMine, GlycoEP, GlycoPP, and GPP on the independent dataset is generated and demonstrated in Fig. 6. The AUC value of PUStackNGly outperforms the other Six predictors and PUStackNGly keeps

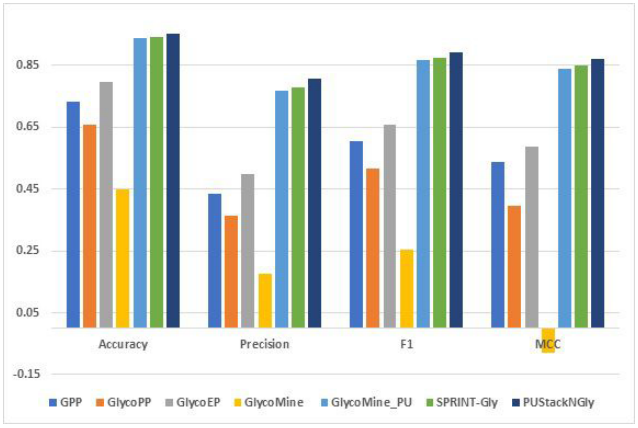

FIGURE 5. Performance results of PUStackNgly compared to SPRINT-GLY, GlycoMine_PU, GlycoMine, GlycoEP, GlycoPP, and GPP predictors on the independent dataset in the term of accuracy, precision, F1, and MCC.

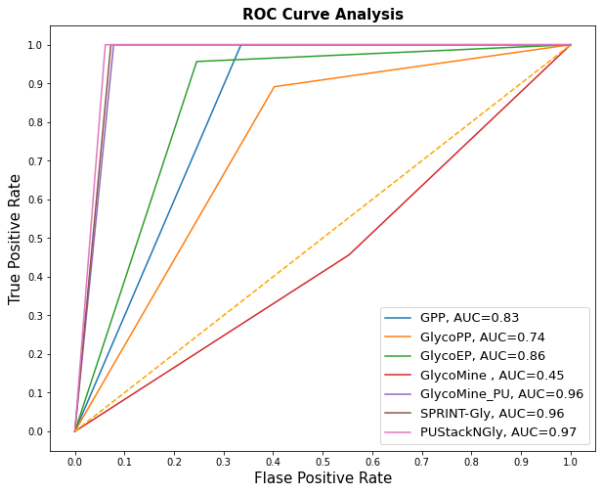

FIGURE 6. ROC AUC curve for PUStackNGly, SPRINT-GLY, GlycoMine_PU, GlycoMine, GlycoEP, GlycoPP, and GPP on the independent dataset.

a proper balance between true positive rate and false-positive rate. Generally, GlycoMine performs the lowest performance results on the independent dataset is the lowest then GlycoPP, GPP, and GlycoEP respectively. The GlycoMine_PU and SPRINT-GLY were released in 2019 and they have better performance than GlycoMine, GlycoEP, GlycoPP, and GPP.

It is clear that PUStackNGly is better than other existing tools. One of the key reasons for notable distinction is the use of PU and ensemble stacking learning. They constructed and developed using six optimized supervised machine learning methods. In addition, the comprehensive extracted features as well as the ensemble-based feature selection method play important roles to improve prediction performance. PUStackNGly will continue to develop and expand in tandem with the experimentally-verified glycoproteins uploaded to the UniProt database, which will enhance the quality of the datasets and also the performance of the proposed model. With the success of implementing PUStackNGly for N-linked glycosylation prediction in this work, we look forward to implementing it with the other glycosylation types and the other different types of PTMs.

IV. CONCLUSION

In this paper, we propose a novel model N-linked glycosylation predictor PUStackNgly based on PU learning and stacking ensemble learning. A benchmark dataset from the UniProt database is extracted and preprocessed for the model. Eight groups including forty-five feature extraction methods are employed to encode each sample. Subsequently, to remove irrelevant features, feature selection is applied using a stable ensemble-based method. After that, proposed ensemble bagging PU learning is implemented to construct the training dataset, for PUStackNGly, from positive and reliable negative samples. Four supervised machine learning methods are used in PU learning including SVM, LR, RF, XGBoost. The last step PUStackNGly is constructing the prediction model using stacking ensemble learning. In the stacking ensemble learning, the LR, ANN, RF, and SVM classification methods are selected for building the base level-0 model and the SVM classification method for the meta level-1 model. PUStackNGly was compared with six supervised classifiers: LR, ANN, RF, XGB, SVM, and KNN on the training and development dataset. The results show improvement performance than the other classifiers. Moreover, for fair evaluation, PUStackNGly is compared with six N-linked glycosylation predictors: GPP, GlycoPP, GlycoEP, GlycoMine, GlycoMine_PU, and SPRINT-GLY. The performance results of the comparison showed that PUStackNGly outperforms the six tools with 95.11% accuracy, 100% recall, 80.7% precision, 89.32% F1 score, 96.93% AUC, and 0.87 MCC. With the success of implementing PUStackNGly, we look forward to implementing it with the other glycosylation types and the other different types of PTMs.

REFERENCES

[1] T. Pitti, C.-T. Chen, H.-N. Lin, W.-K. Choong, W.-L. Hsu, and T.-Y. Sung, "N-GlyDE: A two-stage N-linked glycosylation site prediction incorporating gapped dipeptides and pattern-based encoding," *Sci. Rep.*, vol. 9, no. 1, pp. 1–11, Dec. 2019.

[2] G. Taherzadeh, A. Dehzangi, M. Golchin, Y. Zhou, and M. P. Campbell, "SPRINT-Gly: Predicting *N*- and *O*-linked glycosylation sites of human and mouse proteins by using sequence and predicted structural properties," *Bioinformatics*, vol. 35, no. 20, pp. 4140–4146, Oct. 2019.

[3] M. Hu, Y. Lan, A. Lu, X. Ma, and L. Zhang, "Glycan-based biomarkers for diagnosis of cancers and other diseases: Past, present, and future," *Prog. Mol. Biol. Transl. Sci.* vol. 162, pp. 1–24, Jan. 2019.

[4] P. Regan, P. L. McClean, T. Smyth, and M. Doherty, "Early stage glycosylation biomarkers in Alzheimer's disease," *Medicines*, vol. 6, no. 3, p. 92, Sep. 2019.

[5] Y. Watanabe, Z. T. Berndsen, J. Raghwani, G. E. Seabright, J. D. Allen, O. G. Pybus, J. S. McLellan, I. A. Wilson, T. A. Bowden, A. B. Ward, and M. Crispin, "Vulnerabilities in coronavirus glycan shields despite extensive glycosylation," *Nature Commun.*, vol. 11, no. 1, pp. 1–10, Dec. 2020.

[6] F. Li, Y. Zhang, A. W. Purcell, G. I. Webb, K.-C. Chou, T. Lithgow, C. Li, and J. Song, "Positive-unlabelled learning of glycosylation sites in the human proteome," *BMC Bioinf.*, vol. 20, no. 1, p. 112, Mar. 2019.

[7] J. S. Chauhan, A. Rao, and G. P. S. Raghava, "*In silico* platform for prediction of N-, O- and C-glycosites in eukaryotic protein sequences," *PLoS ONE*, vol. 8, no. 6, Jun. 2013, Art. no. e67008.

[8] F. Li, C. Li, M. Wang, G. I. Webb, Y. Zhang, J. C. Whisstock, and J. Song, "GlycoMine: A machine learning-based approach for predicting N-, C- and O-linked glycosylation in the human proteome," *Bioinformatics*, vol. 31, no. 9, pp. 1411–1419, May 2015.

[9] R. Gupta and S. R. Brunak, "Prediction of glycosylation across the human proteome and the correlation to protein function," in *Proc. Pacific Symp. Biocomput.*, vol. 7, 2002, pp. 310–322.

[10] C. Caragea, J. Sinapov, A. Silvescu, D. Dobbs, and V. Honavar, "Glycosylation site prediction using ensembles of support vector machine classifiers," *BMC Bioinf.*, vol. 8, no. 1, p. 438, Dec. 2007.

[11] S. E. Hamby and J. D. Hirst, "Prediction of glycosylation sites using random forests," *BMC Bioinf.*, vol. 9, no. 1, p. 500, Dec. 2008.

[12] J. S. Chauhan, A. H. Bhat, G. P. S. Raghava, and A. Rao, "GlycoPP: A webserver for prediction of N- and O-glycosites in prokaryotic protein sequences," *PLoS ONE*, vol. 7, no. 7, Jul. 2012, Art. no. e40155.

[13] G.-Y. Chuang, J. C. Boyington, M. G. Joyce, J. Zhu, G. J. Nabel, P. D. Kwong, and I. Georgiev, "Computational prediction of N-linked glycosylation incorporating structural properties and patterns," *Bioinformatics*, vol. 28, no. 17, pp. 2249–2255, Sep. 2012.

[14] F. Li, C. Li, J. Revote, Y. Zhang, G. I. Webb, J. Li, J. Song, and T. Lithgow, "GlycoMine^{struct}: A new bioinformatics tool for highly accurate mapping of the human N-linked and O-linked glycoproteomes by incorporating structural features," *Sci. Rep.*, vol. 6, no. 1, pp. 1–16, Dec. 2016.

[15] M. A. Akmal, N. Rasool, and Y. D. Khan, "Prediction of N-linked glycosylation sites using position relative features and statistical moments," *PLoS ONE*, vol. 12, no. 8, Aug. 2017, Art. no. e0181966.

[16] C.-H. Chien, C.-C. Chang, S.-H. Lin, C.-W. Chen, Z.-H. Chang, and Y.-W. Chu, "N-GlycoGo: Predicting protein N-glycosylation sites on imbalanced data sets by using heterogeneous and comprehensive strategy," *IEEE Access*, vol. 8, pp. 165944–165950, 2020.

[17] A. Alkuhlani, W. Gad, M. Roushdy, and A.-B.-M. Salem, "Intelligent techniques analysis for glycosylation site prediction," *Current Bioinf.*, vol. 16, no. 6, pp. 774–788, Sep. 2021.

[18] T. UniProt Consortium, "UniProt: A hub for protein information," *Nucleic Acids Res.*, vol. 43, no. D1, pp. D204–D212, Jan. 2015.

[19] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li, "CD-HIT: Accelerated for clustering the next-generation sequencing data," *Bioinformatics*, vol. 28, no. 23, pp. 3150–3152, Dec. 2012.

[20] Z. Chen, P. Zhao, F. Li, A. Leier, T. T. Marquez-Lago, Y. Wang, G. I. Webb, A. I. Smith, R. J. Daly, K.-C. Chou, and J. Song, "iFeature: A Python package and web server for features extraction and selection from protein and peptide sequences," *Bioinformatics*, vol. 34, no. 14, pp. 2499–2502, Jul. 2018.

[21] H. Nakashima, K. Nishikawa, and T. Ooi, "The folding type of a protein is relevant to the amino acid composition," *J. Biochem.*, vol. 99, no. 1, pp. 153–162, Jan. 1986.

[22] Y.-Z. Chen, Y.-R. Tang, Z.-Y. Sheng, and Z. Zhang, "Prediction of mucin-type O-glycosylation sites in mammalian proteins using the composition of *k*-spaced amino acid pairs," *BMC Bioinf.*, vol. 9, no. 1, p. 101, Dec. 2008.

[23] V. Saravanan and N. Gautham, "Harnessing computational biology for exact linear B-cell epitope prediction: A novel amino acid composition-based feature descriptor," *OMICS, A J. Integrative Biol.*, vol. 19, no. 10, pp. 648–658, Oct. 2015.

[24] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li, and H. Jiang, "Predicting protein–protein interactions based only on sequences information," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 11, pp. 4337–4341, 2007.

[25] Y. Zuo, Y. Li, Y. Chen, G. Li, Z. Yan, and L. Yang, "PseKRAAC: A flexible web server for generating pseudo K-tuple reduced amino acids composition," *Bioinformatics*, vol. 33, no. 1, pp. 122–124, Jan. 2017.

[26] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks," *Proc. Nat. Acad. Sci. USA*, vol. 89, no. 22, pp. 10915–10919, 1992.

[27] S. Kawashima and M. Kanehisa, "AAindex: Amino acid index database," *Nucleic Acids Res.*, vol. 28, no. 1, p. 374, 2000.

[28] B. Liu, "BioSeq-Analysis: A platform for DNA, RNA and protein sequence analysis based on machine learning approaches," *Briefings Bioinf.*, vol. 20, no. 4, pp. 1280–1294, 2019.

[29] I. Dubchak, I. Muchnik, C. Mayor, I. Dralyuk, and S.-H. Kim, "Recognition of a protein fold in the context of the SCOP classification," *Proteins, Struct., Function, Genet.*, vol. 35, no. 4, pp. 401–407, Jun. 1999.

[30] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Res.*, vol. 25, no. 17, pp. 3389–3402, 1997.

[31] L. J. McGuffin, K. Bryson, and D. T. Jones, "The PSIPRED protein structure prediction server," *Bioinformatics*, vol. 16, no. 4, pp. 404–405, Apr. 2000.
[32] E. Faraggi, T. Zhang, Y. Yang, L. Kurgan, and Y. Zhou, "SPINE X: Improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles," *J. Comput. Chem.*, vol. 33, no. 3, pp. 259–267, Jan. 2012.
[33] A. Alkuhlani, M. Nassef, and I. Farag, "Multistage feature selection approach for high-dimensional cancer data," *Soft Comput.*, vol. 21, no. 22, pp. 6895–6906, 2017.
[34] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review," *J. King Saud Univ.-Comput. Inf. Sci.*, Jun. 2019, doi: 10.1016/j.jksuci.2019.06.012.
[35] Y. Saeys, T. Abeel, and Y. Van de Peer, "Robust feature selection using ensemble feature selection techniques," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2008, pp. 313–325.
[36] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
[37] S. Suresh and V. Naidu, "Mahalanobis-ANOVA criterion for optimum feature subset selection in multi-class planetary gear fault diagnosis," *J. Vibrat. Control*, Jun. 2021, Art. no. 107754632110291, doi: 10.1177/10775463211029153.
[38] G. Zhang, J. Hou, J. Wang, C. Yan, and J. Luo, "Feature selection for microarray data classification using hybrid information gain and a modified binary krill herd algorithm," *Interdiscipl. Sci., Comput. Life Sci.*, vol. 12, no. 3, pp. 288–301, Sep. 2020.
[39] Z. Zhang, G. Wang, C. Liu, L. Cheng, and D. Sha, "Bagging-based positive-unlabeled learning algorithm with Bayesian hyperparameter optimization for three-dimensional mineral potential mapping," *Comput. Geosci.*, vol. 154, Sep. 2021, Art. no. 104817.
[40] F. Mordelet and J.-P. Vert, "A bagging SVM to learn from positive and unlabeled examples," *Pattern Recognit. Lett.*, vol. 37, pp. 201–209, Feb. 2014.
[41] S. Gattani, A. Mishra, and M. T. Hoque, "StackCBPred: A stacking based prediction of protein-carbohydrate binding sites from sequence," *Carbohydrate Res.*, vol. 486, Dec. 2019, Art. no. 107857.
[42] S. Cui, Y. Yin, D. Wang, Z. Li, and Y. Wang, "A stacking-based ensemble learning method for earthquake casualty prediction," *Appl. Soft Comput.*, vol. 101, Mar. 2021, Art. no. 107038.
[43] J. Xiao, "SVM and KNN ensemble learning for traffic incident detection," *Phys. A, Stat. Mech. Appl.*, vol. 517, pp. 29–35, Mar. 2019.
[44] A. Pernía-Espinoza, J. Fernandez-Ceniceros, J. Antonanzas, R. Urraca, and F. J. Martinez-de-Pison, "Stacking ensemble with parsimonious base models to improve generalization capability in the characterization of steel bolted components," *Appl. Soft Comput.*, vol. 70, pp. 737–750, Sep. 2018.
[45] Y. Wang, D. Wang, X. Ye, Y. Wang, Y. Yin, and Y. Jin, "A tree ensemble-based two-stage model for advanced-stage colorectal cancer survival prediction," *Inf. Sci.*, vol. 474, pp. 106–124, Feb. 2019.
[46] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1997.
[47] J. Tolles and W. J. Meurer, "Logistic regression: Relating patient characteristics to outcomes," *Jama*, vol. 316, no. 5, pp. 533–534, 2016.
[48] L. J. Lancashire, C. Lemetre, and G. R. Ball, "An introduction to artificial neural networks in bioinformatics-application to complex microarray and mass spectrometry datasets in cancer studies," *Briefings Bioinf.*, vol. 10, no. 3, pp. 315–329, Dec. 2008.
[49] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[50] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.
[51] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statistician*, vol. 46, pp. 175–185, Aug. 1992.
[52] E. El Houby and N. Yassin, "Methodology for selecting microarray biomarker genes for cancer classification," *Int. J. Intell. Comput. Inf. Sci.*, vol. 15, no. 1, pp. 25–39, Jan. 2015.
[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, and D. Cournapeau, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 10, pp. 2825–2830, Jul. 2017.

ALHASAN ALKUHLANI received the B.Sc. degree in computer science from the Faculty of Sciences, Sana'a University, Yemen, in 2007, the M.Sc. degree in computer science from the Faculty of Computers and Information, Cairo University, Egypt, in 2017. He is currently pursuing the Ph.D. degree with the Faculty of Computer and Information Sciences, Ain Shams University, Egypt. His master's degree was about computational selection of cancer DNA methylation genes. 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.

WALAA GAD received the B.Sc. and M.Sc. degrees in computers and information sciences from Ain Shams University, Cairo, Egypt, in 2000 and 2005 respectively, and the Ph.D. degree in computers and information sciences from the Pattern and Machine Intelligence (PAMI) Group, Faculty of Electrical and Computer Engineering, University of Waterloo, Canada, in 2010. Her master's degree was about designing and planning a network model in the presence of obstacles using clustering around medoids techniques. The dissertation title is "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 an Associate 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.

MOHAMED ROUSHDY received the B.Sc., M.Sc., and Ph.D. degrees from the Faculty of Science, Ain Shams University, in 1979, 1984, and 1993, respectively. His experimental doctoral research work was conducted at Bochum University, Germany, from 1989 to 1991. He is currently a Professor of computer science and the Dean of the Faculty of Computers and Information Technology, Future University in Egypt, Cairo, Egypt. He received Ain Shams University Appreciation Award in Technological Sciences, in 2018.

ABDEL-BADEEH M. SALEM has been a Full Professor of computer science with the Faculty of computer and information sciences, Ain Shams University, Cairo, Egypt, since1989. He is the Founder and the Chairman of the Artificial Intelligence and Knowledge Engineering Research Laboratories, Ain Shams University. He is the Chair of the Working Group on Bio-Medical Informatics, ISfTeH, Belgium. He has published around 700 papers. He has been 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 European Academy of Sciences and Arts, Austria.