Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 5, No. 4, November 2020, Pp. 253-262

Prediction insulin-protein interactions associated based on ontology genes using extreme gradient boosting and centrality method

Mohammad Hamim Zajuli Al Faroby*[1], Mohammad Isa Irawan[2], Ni Nyoman Tri Puspaningsih[3]
Department of Mathematics, Faculty of Science and Data Analytics, Institut Teknologi Sepuluh Nopember, Indonesia[1,2]
Department of Chemistry, Faculty of Scinece and Technology, Airlangga university, Indonesia[3]

Article Info

Keywords:
Extreme Gradient Boosting, Centrality, Insulin, Machine Learning, Gene Onthology, Protein-protein Interaction

Article history:
Received 13 May 2020
Accepted 25 August 2020
Published 30 November 2020

Cite:
Al Faroby, M., Irawan, M., & Puspaningsih, N. (2020). XGBoost and Network Analysis for Prediction of Proteins Affecting Insulin based on Protein Protein Interactions. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 5(4). doi:https://doi.org/10.22219/kinetik.v5i4.1076

*Corresponding author.
Mohammad Hamim Zajuli Al Faroby
E-mail address:
hamim.18061@mhs.its.ac.id

Abstract

Protein Interaction Analysis (PPI) can be used to identify proteins that have a supporting function on the main protein, especially in the synthesis process. Insulin is synthesized by proteins that have the same molecular function covering different but mutually supportive roles. To identify this function, the translation of Gene Ontology (GO) gives certain characteristics to each protein. This study purpose to predict proteins that interact with insulin using the centrality method as a feature extractor and extreme gradient boosting as a classification algorithm. Characteristics using the centralized method produces 12 features as a central function of protein. Classification results are measured using measurements, precision, recall and ROC scores. Optimizing the model by finding the right parameters produces an accuracy of 74.56% and a ROC score of 0.6383. The prediction model produced by XGBoost has capabilities above the average of other machine learning methods.

1. Introduction

Insulin is a crucial protein for biological processes. Insulin protein converts glucose in the blood into energy. If the work of insulin is disrupted it can lead to Diabetes Mellitus [1]. In biological processes, Insulin does not work alone, but is assisted by other proteins that supporting function and activation of insulin. Therefore, to find out what proteins have an influence on insulin, an analysis of Protein-Protein Interactions (PPIs) is needed. Recent developments in high throughput Experimental biology and Computational biology have produced large data protein-protein interactions (PPIs), which are represented as networks, where nodes correspond with proteins and edges correspond to interactions between proteins [2]. Protein interactions have a correlation with protein function, so the equation of protein function forms the interaction between one protein with another protein. It is known that proteins that interact physically tend to be involved in the same cellular processes, and mutations in their genes can cause similar disease phenotypes [3].

The functions of proteins known by analyzing the structure of Gene Ontology (GO) [4]. The same functional between proteins can be measured by semantic similarity, the function that returns numerical values reflects the closeness of meaning between the two ontological terms affixing protein information. GO is a repository of biological ontologies, gene annotations and gene products. Although the annotation data are based on published evidence originating from most unreliable high throughput experiments, they are often used as a benchmark for functional characterization due to their completeness [3][5]. In the research of G. Montanez and Y. Cho assess the reliability of PPI using GO annotation data determined experimentally and concluded computationally. While using the inferred annotation data to trim the inferred protein interactions can be surprising, the resulting bias is in the direction of confirming the validity of PPI, so that true interactions cannot be classified as wrong, at the expense of leaving some fake PPIs undetectable. While acknowledging the disadvantages of such an approach, this allows leveraging freely available GO data to potentially improve the reliability of PPI data sets [6].

GO annotation data is represented as Directed Acyclic Graph (DAG) which only provides information on the relationship of functions to one another. To give a meaningful value to the DAG, the network analysis method can be used for weighting each leaf [7]. The graphical approach has the advantage of determining centralization in the network, and the central node determines an important role in biological processes [8]. Therefore, the centrality method is suitable to be used as a weighting value on DAG to get features on each function of protein molecules. Some researchers use the centrality method to analyze PPI such as Centrality Closeness (CC), Edge Clustering Centrality Coefficient (NC), Intermediate Centralness (BC), Degree Centrality (DC), Eigen Vector Centrality (EC), Information Centrality (IC) and

Subgraph Centrality [9]. However, not all methods used can be used for DAG weighting, because it is classified as directed graph, so the selection of certain methods only.

The role of computing is more often in the preparation of predictive models for biological objects. Computing Update has the resolution and speed in building predictive models. The most popular early method for processing large databases is the machine learning method, such as Bayesian, Probabilistic Decision Tree, Logistic Regression, and Support Vector Machine [10] have been introduced to solve the problem of predicting protein-protein interactions by using the property of proteins to classify data [11], however, the primitive methods above are less able to provide good predictive models on the accuracy of the model. For example, the best accuracy so far is the Classification using Support Vector Machine on protein interaction that affects diabetes mellitus produce an accuracy of 73.6%, using as many as 2653 data [12]. An updated method is needed that can better solve the problem of data complexity.

This paper proposes a method for predicting PPIs based on GO by using Extreme Gradient Boosting (XGBoost) with feature extraction using the centrality method to build a dataset, Other researchers conducted feature selection experiments based on the knowledge they knew [13], because of the compatibility of the data with the form of directed graphs and the method used. XGBoost has the advantage of using a gradient enhancement strategy, the increase is obtained from the combination of a decision tree which is the basis of a weak classification [14]. The advantage of this method lies in the ability to empower several simple classifiers to model small datasets and can prevent overfitting caused by the complexity of the model. More importantly, the XGBoost method allows researchers to evaluate the contribution of each feature contained in the dataset in generating predictive models [15]. In some biological activity dataset testing the XGBoost method outperformed other machine learning algorithm methods such as Random Forest (RF), Support Vector Machines (SVM), Radial Basis Function Neural Networks (RBFN) and Naïve Bayes (NB). The ability of XGBoost shows exceptional performance on high and low diversity datasets, and in detecting minority activity classes on unbalanced data [16].

The final goal of this research is to build a PPIs dataset based on gene ontology using the centrality method. The results of the dataset are used to get the Extreme Gradient Boosting Classification prediction model, the model can predict whether a protein has a strong interaction with insulin or not by considering the XGBoost prediction probability value.

2. Research Metodology

This paper proposes the use of the Betwenness Centrality (BC), Closeness Centrality (CC), and Pagerank Centrality (PRC) methods to build a dataset. The resulting dataset is classified using the XGBoost method to find the best model from the three datasets. The process of the research as shown in Figure 1.

The initial stage is collecting GO data and selecting data related to the function of protein molecules. After that, GO annotation codes are converted to DAG on the www.ebi.ac.uk/QuickGO page, then the centrality value is calculated using BC, CC, and PRC which generates a dataset for each method. After that, each dataset is preprocessed, before it is classified to produce a prediction model.

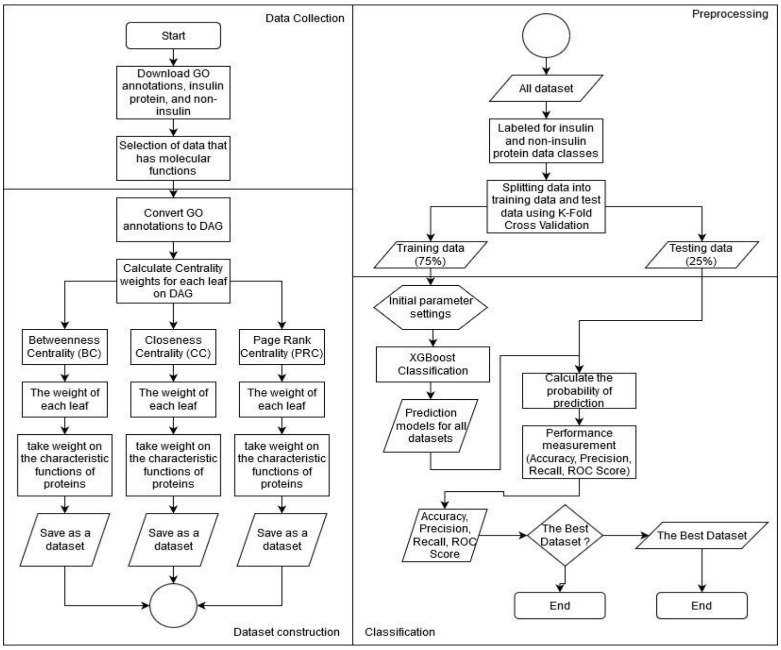
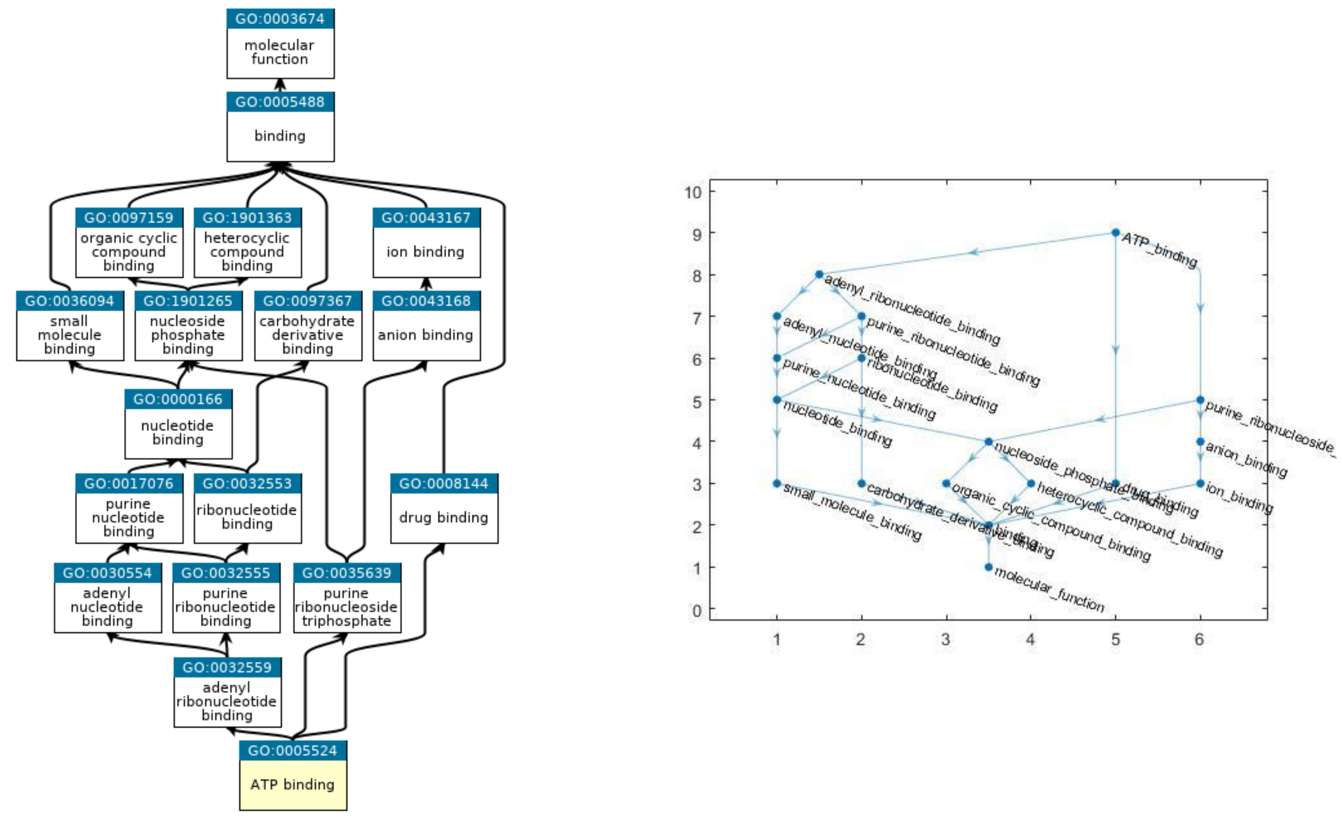

Data Collection
Start
Download GO annotations, insulin protein, and non-insulin
Selection of data that has molecular functions
Convert GO annotations to DAG
Calculate Centrality weights for each leaf on DAG
Betweenness Centrality (BC)
Closeness Centrality (CC)
Page Rank Centrality (PRC)
The weight of each leaf
The weight of each leaf
The weight of each leaf
take weight on the characteristic functions of proteins
take weight on the characteristic functions of proteins
take weight on the characteristic functions of proteins
Save as a dataset
Save as a dataset
Save as a dataset
Dataset construction
Preprocessing
All dataset
Labeled for insulin and non-insulin protein data classes
Splitting data into training data and test data using K-Fold Cross Validation
Training data (75%)
Testing data (25%)
Initial parameter settings
XGBoost Classification
Prediction models for all datasets
Calculate the probability of prediction
Performance measurement (Accuracy, Precision, Recall, ROC Score)
Accuracy, Precision, Recall, ROC Score
The Best Dataset ?
The Best Dataset
End
End
Classification

Figure 1. The Proposed Method

2.1 Data Collection and Dataset Construction

To build datasets, we select 1595 data insulin protein synthesize and 865 data non-insulin. Data will be classified into two classes of proteins namely insulin protein and non-insulin protein. A total of 2460 data was obtained from UniProt and Protein Databank (PDB). From all the data collected, they are selected to separate proteins that have molecular functions and do not have functions in their molecules. So, the results of the selection obtained 2004 selected data, consisting of 1295 positive data as insulin protein and 709 negative data as non-insulin protein. The shape of the dataset matrix, the row shows the amount of protein data that builds the dataset, while the column shows the features used. The features of the molecular functions used are Binding, Cargo Receptor Activity, Catalytic Activity, Molecular Function Regulator, Molecular Transducer Activity, Positive Regulation of Molecular Function, Negative Regulation of Molecular Function, Regulation of Molecular Function, Structural Molecule Activity, Transcription Regulator Activity, and Transporter Activity. The features used above have influence values that play a role in the performance of molecular functions in genes related to insulin. The processing of these features is based on the gene ontology of each protein that is used as data.

The centrality method is one of the graph methods which gives weight to the nodes of the graph. These methods are often used in network analysis problems, to get weight by paying attention to the relationship of each node on the network (graph). In GO data, the network that occurs can be assumed to be a directed graph, so that the centrality method can be calculated the measure of each process in the DAG [17]. If we pay attention to Figure 2a, it is a depiction of GO before looking for the centrality values. This description is assisted by MATLAB in drawing graphs and calculating centrality values. In this study, three centrality methods are used namely Closeness Centrality, Betweenness Centrality, and Page Rank Centrality. For example, we use a molecular function as Figure 2a. Furthermore, the ancestor chart was changed to DAG in Figure 2b.


GO:0003674
molecular function
GO:0005488
binding
GO:0097159
organic cyclic compound binding
GO:1901363
heterocyclic compound binding
GO:0043167
ion binding
GO:0036094
small molecule binding
GO:1901265
nucleoside phosphate binding
GO:0097367
carbohydrate derivative binding
GO:0043168
anion binding
GO:0000166
nucleotide binding
GO:0017076
purine nucleotide binding
GO:0032553
ribonucleotide binding
GO:0008144
drug binding
GO:0030554
adenyl nucleotide binding
GO:0032555
purine ribonucleotide binding
GO:0035639
purine ribonucleoside triphosphate
GO:0032559
adenyl ribonucleotide binding
GO:0005524
ATP binding
10
9
8
7
6
5
4
3
2
1
0
1
2
3
4
5
6
ATP_binding
adenyl_ribonucleotide_binding
adenyl_nucleotide_binding
purine_ribonucleotide_binding
purine_nucleotide_binding
ribonucleotide_binding
nucleotide_binding
purine_ribonucleoside_
anion_binding
nucleoside_phosphate_binding
drug_binding
ion_binding
small_molecule_binding
carbohydrate_derivative_binding
organic_cyclic_compound_binding
heterocyclic_compound_binding
binding
molecular_function

(a) (b)

Figure 2. DAG of Functions Related to ATP Binding in the Molecular Function (a) Protein Function Networks After GO Annotation Translation on the www.ebi.ac.uk/QuickGO page, and (b) DAG Depiction for Weight Calculation using the Centrality Method in MATLAB

The Closeness Centrality (CC) method is a method that calculates the best distance at each node which is formulated in Equation 1, d_{ij} shows the distance from leaf i to leaf j. The weight of the distance is measured as the minimum of hops needed to move from i to j [18]. The average distance from a node i to the other is given as,

$$l_i = \frac{1}{n}\sum_j d_{ij} \tag{1}$$

The Betweenness Centrality (BC) method of node i can be said as how often node i finds the shortest path between two random nodes of a network [19]. For example, g_{st} becomes the shortest amount of distance between s and t, then n_{st}^{i} is the shortest amount of distance between s and t that passes through node i. The BC value of i can be stated as Equation 2.

$$x_i = \sum_{s,t\in V}\frac{n_{st}^{i}}{g_{st}} \tag{2}$$

with the convention of $\frac{n_{st}^{i}}{g_{st}} = 0$ if both values are 0.

Page Rank Centrality (PRC) is a method of weighting in directed graphs that results from random travel on a network. At each node in the graph, the next node is selected with the probability of a series of nodes being passed on to the initial node (in the case of an undirected graph it can be called a neighbor). If a node has no successor, then the next node is selected from all the existing nodes. The score from the PRC is the average time spent on each random search. If the node has a loop to itself, then there is a possibility that the algorithm will pass through the vertex, therefore looping to itself can increase the PRC score [7]. Score calculation with PRC as in Equation 3,

$$p_i = \alpha\sum_{j=1}^{n} A_{ij}\frac{p_j}{\sigma_j^{+}} + \beta \tag{3}$$

where σ_j^{+} is out-degree of node j. However, some nodes have $\sigma_j^{+} = 0$, which will cause division with a value of 0. So, in this case, a vertex is added which is a loop from j to j itself, to produce $\sigma_j^{+} = 1$. For the record, the node still has no contribution value concerning the centrality of the other nodes.

2.2 Extreme Gradient Boosting

Extreme Gradient Boosting was formulated as the sum of leaf weights in K-CART (Classification and regression trees) trees. Let dataset with n samples and m features, $\varphi = \{(x_i, y_i)\}(\|\varphi\| = n, x_i \in \mathbb{R}^m)$, the tree ensemble model uses the additive function K to predict output formulated by Equation 4 [14],

$$\hat{y}_i = \phi(x_i) = \sum_{i=1}^{K} f_k(x_i), f_k \in \mathcal{F} \tag{4}$$

where $\mathcal{F} = \{f(x) = w_q(x)\}$ for $q: \mathbb{R}^m \to T, w \in \mathbb{R}^T$ which is the space of the regression tree. q is a representation of the structure of each tree that maps the dataset to the corresponding leaf index. T is the number of leaves in a tree. Each f_k corresponds to an independent tree structure q and leaf weight w. Like the decision tree, each regression tree has a continuous score on each leaf, w_i representing the score on the i-leaf. For example, we use the decision tree rule on a tree (in q) to classify it into leaves and calculate the final prediction by adding up the scores on the corresponding leaves (on w).

The learning function of the model used is to minimize the objective function $\mathcal{L}$ on Equation 5 [20],

$$\mathcal{L}(x) = \sum_{i=1} l(y_i, \hat{y}_i) + \sum_{K=1} \Omega(f_k) \tag{5}$$

where,

$$\Omega(x) = \gamma T + \frac{1}{2}\lambda\|w\|^2$$

l is a loss function that can be distinguished between prediction $\hat{y}_i$ and target y_i or it can be said that the function must be differentiable, whereas Ω is the risk of complexity model. The loss function used is a loss function used in logistic regression. The ensemble tree of Equation 5 contains functions as parameters and cannot be optimized with traditional optimization methods in the Euclid space. Instead, the model is additive trained. Given $\hat{y}_i(t)$ to be the prediction of the i-th event in the t-iteration, it takes f_t to minimize the following objectives,

$$\mathcal{L}^{(t)} = \sum_{i=1} \left(l(y_i, \hat{y}_i^{t-1}) + f_t(x_i)\right) + \Omega(f_t) \tag{6}$$

the second-order Taylor expansion approach can be used to rapidly optimize the objectives on Equation 6 [21],

$$\mathcal{L}^{(t)} \cong \sum_{i=1}^{n} \left(l(y_i, \hat{y}_i^{t-1}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right) + \Omega(f_t) \tag{7}$$

where, $g_i = \frac{\partial\, l(y_i, \hat{y}_i^{t-1})}{\partial, \hat{y}_i^{t-1}}$ and $h_i = \frac{\partial^2\, l(y_i, \hat{y}_i^{t-1})}{\partial, \hat{y}_i^{t-1^2}}$, g is the first derivative of a loss function commonly known as a gradient. Whereas the h value is the second derivative of the loss function known as the Hessian.

Unlike random forests which reduce this loss function by splitting features on the biggest information gain and randomly ensembled CART trees, XGBoost transforms the loss function into a new scoring function for selecting the best threshold,

$$\mathcal{L}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{\left(\sum_{i \in I_j} g_i\right)^2}{\left(\sum_{i \in I_j} h_i + \lambda\right)} + \gamma T \tag{8}$$

Where $\mathcal{L}^{(t)}(q)$ is the second-order approximation of the loss function at the t-th iteration for weighting q tree, g_i, and h_i are the first and second-order loss gradient on the i-th data. I_j is the instance set of a certain leaf node j. In this way, XGBoost able to iteratively reduce loss and achieve better performance than other ensemble methods [16].

The predicted value of a leaf node can be calculated with the optimal weight equation based on the first derivative of Equation 7, $f'(x) = 0$, so the predicted score is,

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{9}$$

Equation 8 and Equation 9 can be used as an extraction function to measure the quality of a tree structure q. Because it is not possible to mention all the possible tree structures q. The greedy algorithm starts with a single leaf and iteratively adds branches to the tree that is used as a replacement. Assume I_L and I_R are sample sets for the left and right sides after separation. Given $I = I_L \cup I_R$, then the loss reduction equation after separation is,

$$\mathcal{L}_{gain} = \frac{1}{2}\left[\frac{\left(\sum_{i \in I_L} g_i\right)^2}{\left(\sum_{i \in I_L} h_i + \lambda\right)} + \frac{\left(\sum_{i \in I_R} g_i\right)^2}{\left(\sum_{i \in I_R} h_i + \lambda\right)} + \frac{\left(\sum_{i \in I} g_i\right)^2}{\left(\sum_{i \in I} h_i + \lambda\right)}\right] \tag{10}$$

because the value of $\frac{1}{2}$ is a constant multiple of the equation, it can be ignored. Equation 10 is used to find the best root of a decision tree to minimize the loss function.

3. Results and Discussion

3.1 The Result of Building Dataset

The datasets built using the centrality method have different values on each weight. The dataset formed was 2004 as sample data rows and 11 feature columns. The value of the features of each data is obtained from the sum of the 205 functions that affect the main function which is used as a column feature. The weighting of DAG was using 3 centrality methods, the results of the dataset formed are 3 datasets. To determine which dataset is used must be analyzed the accuracy of the model formed from each dataset [22]. The results of the quality of the three dataset models can be seen in Table1**.** it showed the results of prediction model performance against all three datasets. The measurement is based on the value of accuracy, precision, recall and ROC Score.

Table 1. XGBoost Model Performance Measurement of the Three Datasets

Datasets	Accuracy		Precision	Recall	ROC Score
	Train	Test			
$data\_bc\_insulin$	73.48%	74.56%	85.33%	80.65%	0.6742
$data\_cc\_insulin$	73.30%	72.05%	83.32%	80.32%	0.5329
$data\_pr\_insulin$	72.55%	74.39%	85.84%	81.01%	0.6383

By analyzing Table 1, the dataset built with weights obtained from Betweenness Centrality has better accuracy than the dataset built with weights obtained from Closeness Centrality and Page Rank Centrality, it has accuracy 74.56%. These results indicate the centrality method influences the XGBoost model used for classification. GO feature extraction with the Betweenness Centrality method makes the XGBoost prediction model better than the other two methods. Besides, if observed from the three datasets, the resulting accuracy is not much different, so feature extraction with the centrality method produces a good XGBoost classification model. In addition to the accuracy parameters, the values of precision and recall indicate differences that are not much different from the three datasets. Because the more influential parameter for the next process is only the accuracy of the model, the precision (diversity of data in class) and recall are not taken into account because the difference in values is not much different from each other. The results of the measurement accuracy of the model formed from each dataset are used as a reference for the next research process. The next process uses only one dataset, so the dataset namely $data\_bc\_protein$ is chosen as the main dataset. From these results, it can also be predicted that the BC method is better used for weighting DAG in the establishment of datasets.

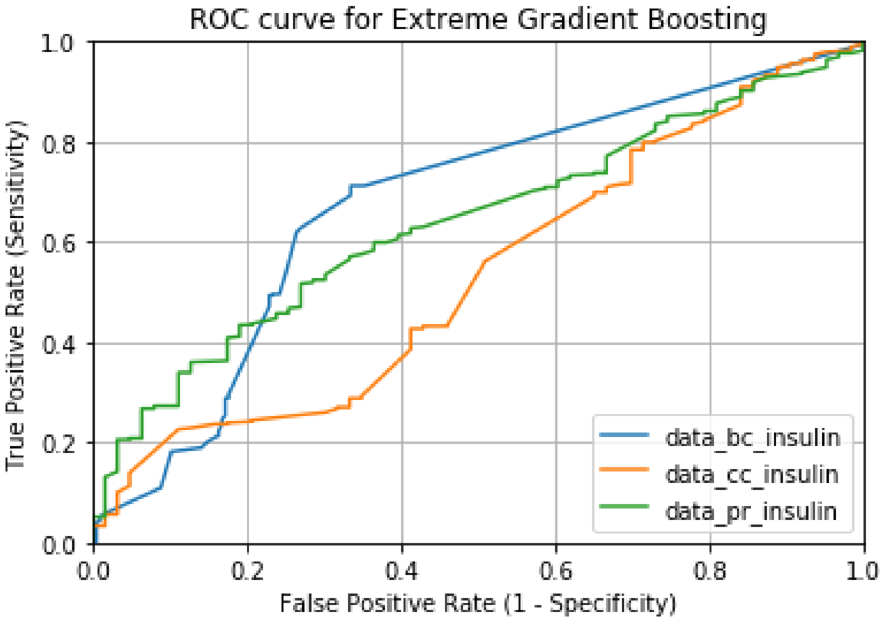


Figure 3. ROC Curves of the Three Datasets Against the XGBoost Model

The results of selecting the dataset used as an interaction model are strengthened by analyzing the ROC curve. The ROC curve in Figure 3 shows the area under the curve of each model built using the XGBoost model. The ROC value is obtained from the Area Under Curve (AUC), The greater AUC value is a better model [23]. The curve of $data\_bc\_insulin$ has the largest area than other curves or has a value of 0.6742, while $data\_cc\_insulin$ and $data\_pr\_insulin$ have values of 0.5329 and 0.6383, respectively. The points in the ROC graph illustrate all possible TP and FP if we run the threshold from the bottom until the top.

3.2 Prediction Result and Optimization

The performance of the built model can be seen, how the model can minimize errors when classifying the objective function of XGBoost model is to minimize the loss function and the model complexity function. In each CART tree model formed has varying error values, when the formation of the first tree will have a greater error value than the next trees, while as the iteration goes up to the specified tree, the model error will be smaller until the model has an error constant.

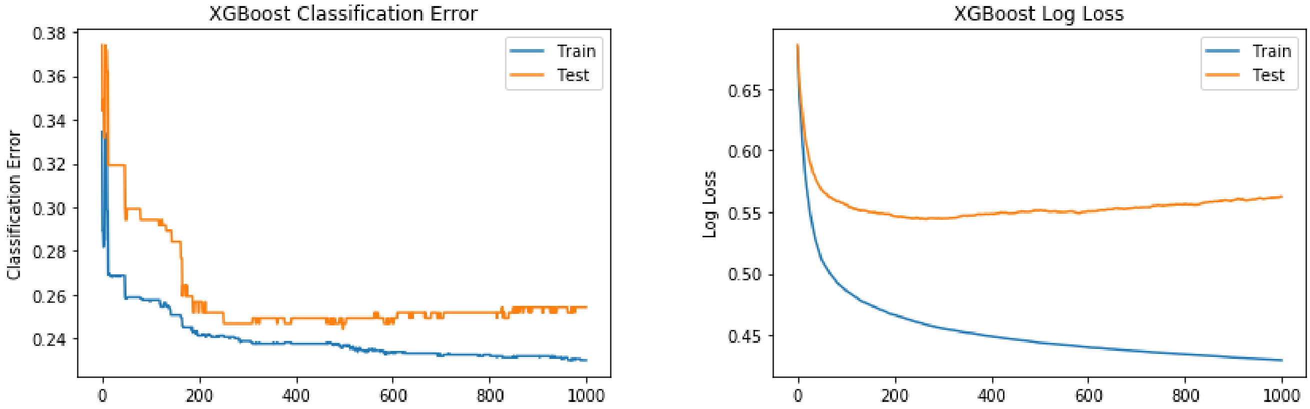


Figure 4. Misclassification and Loss Function Values at Each CART Tree Formation

Testing the performance of XGBoost in constructing the CART tree model is visualized in Figure 4. The XGBoost Log Loss graph shows errors that occur during the formation of the 0th tree to the $1000th$ tree. The graph shows the error in the training process resulting in a smaller value, every time the model builds new trees, it is evident in modeling the XGBoost data that it tries to minimize the loss function used. Whereas the prediction process uses test data, the same as during the learning process (training) the error value in the formation of the 0th CART tree has an error value greater than the other built CART trees. This is reasonable because the formation of the first tree is a weak classification with a decision tree. As more trees are formed, the resulting error value decreases. However, at a certain n-tree value, the error value on the graph begins to increase again, it is taught because the model tries to accept all the feature specifications contained in the dataset

If observed from the results of its classification on the graph presented in Figure 4, the classification model validated using training data will be better at each new CART tree formation. The results indicate the XGBoost algorithm works by correcting the misclassification of the nth tree against the $(n-1)th$ tree. Whereas in the test data, the classification model experienced many adjustments to the complexity and diversity of the value features. Futhermore, more CART trees that are formed the possibility of errors getting bigger too. As the number of trees used the errors that occurred in the training process decreased while in the test process increased. This situation can lead to an overfitting model. An overfitting model has a low loss during training but functions poorly when predicting new data [24]. The overfitting model has a small bias and a large variant. However, if the model is too simple it allows the model to be underfitting, the model has high bias and low variance. These conditions can make the model worse, most researchers these two conditions tend to be avoided. To get a good model in the sense of not experiencing overfitting or underfitting, the formation of CART trees can be limited to equilibrium error states that are not too complex and not too simple. From Figure 5 it can be analyzed that the model formed is not very overfitting, so the model is well-formed.

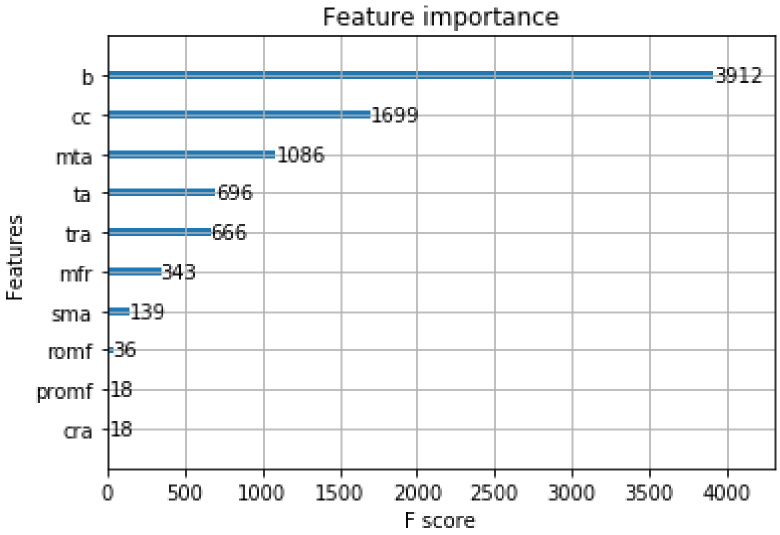


Figure 5. Important Features that Influence the Formation of the XGBoost Prediction Model

Some features have a large influence on the classification model. In Figure 5 shows, the Binding feature has a higher $F - score$ than the others, the value indicates that the Binding feature has a considerable influence compared to the other features [25]. Other features that have a major influence on the formation of the model are Catalytic Activity and Molecular Transducer Activity. The effect of these features is usually due to the diversity of values that exist on the feature. The Negative Regulation of Molecular Function (NROMF) feature in this model is not considered or can be ignored. Because, these two features may have no diversity value, so they did not affect when they are made into nodes in each CART tree formed in the model. The classification model that is formed uses only 9 features, which has a significant influence on the model being built.

3.3 Comparison of XGBoost Prediction Models with other Machine learning Methods

Less complete if the model produced by XGBoost is not compared to other machine learning methods. The dataset used for comparison uses the highest quality dataset, the dataset that was built using the centrality betweenness method. Some popular machine learning methods such as Logistic Regression, K-Nearest Neighbors, Support Vector Machine Classification, Multilayer Perceptron, Random Forest, and its predecessor Gradient Boosting are used as a comparison to the XGBoost prediction model. The settings of each machine learning are set by default, so there is no optimization of one method.

Table 2. Performance Measurement of Several Machine Learning Methods as a Comparison of Predictive Models

Method	Accuracy	Percision	Recall	ROC Score
Logistic Regression	0.61676646706586	0	1.0	0.684887405609493
K-Nearest Neighbors	0.67265469061876	0.50520833	0.77669902912621	0.7193230852211434
Support Vector Machine	0.69061876247504	0.390625	0.87702265372168	0.6923375134843581
Multilayer Perceptron	0.69660678642714	0.53125	0.79935275080906	0.7632483818770225
Random Forest	0.72255489021956	0.5572916	0.82524271844660	0.7782665857605179
Gradient Boosting	0.70459081836327	0.40625	0.8899676375404	0.7773732470334412
Extreme Gradient Boosting	0.72654690618762	0.58854166	0.81229773462783	0.789231054476807

The results of comparison of several machine learning methods are presented in Table 2. This method is the default design and has not been optimized. The method of increasing Extreme Gradients is superior among other methods, the value of accuracy is above all methods. Other evidence is shown by the ROC score, which has a higher value than the rival method. XGBoost has proven superiority in complex data and unbalanced data sets. The resulting prediction model also has very good accuracy, close to real conditions.

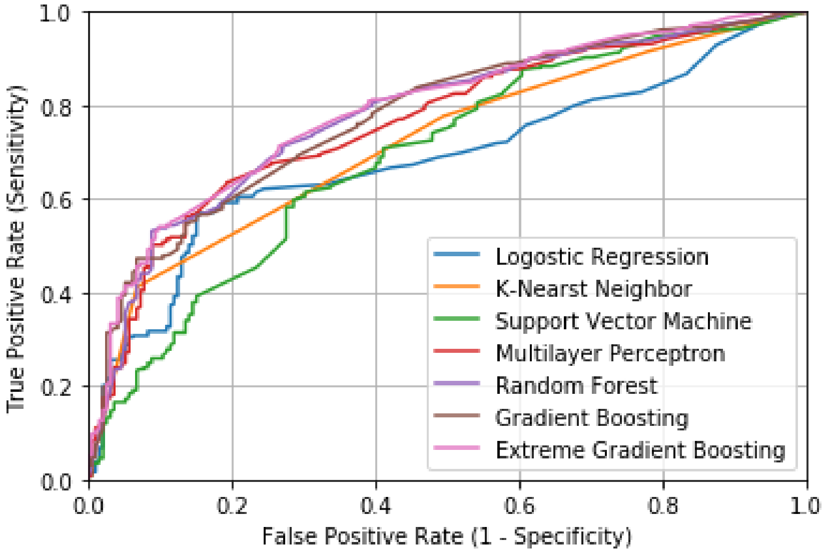


Figure 6. ROC Graph Simulations of Several Methods are Compared

Clearer visualization of the performance of machine learning methods by looking at the ROC graphs of each method. ROC value means that the higher the value of the model, the better. The large value of ROC is assessed from the integral Area Under Curve (AUC). Some of the graphs published in Figure 6 show that the XGBoost method has lines over other methods. More precisely the method of predicting data.

Prediction models that have been formed can be used to predict proteins by extracting features that similar analyzed. More specifically, the resulting prediction model can be further analyzed to construct Protein Interactions to produce a significant target protein. Significant protein results can be used as a reference for making drugs related to insulin, especially Diabetes Mellitus.

4. Conclusion

Predicting proteins that interact with insulin can be done by analyzing the structure of gene ontology. Building a dataset in the form of DAG Ontology Genes is more beneficial by network analysis using the centrality method. The best centrality method used to build a dataset is Betweenness Centrality. The XGBoost algorithm studies data sets based on data constructions built from GO to produce predictive models. Producing 1000 trees, the algorithm has no problems and remains effective between the time and the accuracy of the model, gives an accuracy of 74.56%. The XGBoost prediction model is also better when compared to other machine learning methods. ROC score is above other methods, namely 0.789231054476807. The XGBoost algorithm for the analysis of proteins that interact with insulin produces a better prediction model combined with a centrality method as a feature extractor

Acknowledgement

This research was supported and fully funded by a grant from the Ministry of Research and Technology of the Republic of Indonesia. The authors fully acknowledge the Ministry of Research and Technology for the approved funding that makes this research feasible and effective.

References

[1] J. Calles-Escandon and M. Cipolla, “Diabetes and endothelial dysfunction: A clinical perspective,” *Endocr. Rev.*, vol. 22, no. 1, pp. 36–52, 2001. https://doi.org/10.1210/edrv.22.1.0417

[2] P. Sun *et al.*, “Protein Function Prediction Using Function Associations in Protein-Protein Interaction Network,” *IEEE Access*, vol. 6, pp. 30892–30902, 2018. https://doi.org/10.1109/ACCESS.2018.2806478

[3] W. Xiong, L. Xie, S. Zhou, and J. Guan, “Active learning for protein function prediction in protein-protein interaction networks,” *Neurocomputing*, vol. 145, pp. 44–52, 2014. https://doi.org/10.1016/j.neucom.2014.05.075

[4] G. S. Oliveira and A. R. Santos, “Using the gene ontology tool to produce de novo protein-protein interaction networks with IS_A relationship,” *Genet. Mol. Res.*, vol. 15, no. 4, 2016. https://doi.org/10.4238/gmr15049273

[5] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble, “Investigating semantic similarity measures across the gene ontology: The relationship between sequence and annotation,” *Bioinformatics*, vol. 19, no. 10, pp. 1275–1283, 2003. https://doi.org/10.1093/bioinformatics/btg153

[6] G. D. Montañez and Y. R. Cho, “Assessing reliability of protein-protein interactions by gene ontology integration,” in *2012 IEEE Symposium on Computational Intelligence and Computational Biology, CIBCB 2012*, 2012, pp. 21–27. https://doi.org/10.1109/CIBCB.2012.6217206

[7] G. Iván and V. Grolmusz, “When the web meets the cell: Using personalized PageRank for analyzing protein interaction networks,” *Bioinformatics*, vol. 27, no. 3, pp. 405–407, 2011. https://doi.org/10.1093/bioinformatics/btq680

[8] S. Iyer, T. Killingback, B. Sundaram, and Z. Wang, “Attack Robustness and Centrality of Complex Networks,” *PLoS One*, vol. 8, no. 4, 2013. https://doi.org/10.1371/journal.pone.0059613

[9] J. Zhong, J. Wang, W. Peng, Z. Zhang, and M. Li, “A feature selection method for prediction essential protein,” *Tsinghua Sci. Technol.*, vol. 20, no. 5, pp. 491–499, 2015. https://doi.org/10.1109/TST.2015.7297748

[10] S. Mei and H. Zhu, “A novel one-class SVM based negative data sampling method for reconstructing proteome-wide HTLV-human protein interaction networks,” *Sci. Rep.*, vol. 5, p. 8034, 2015. https://doi.org/10.1038/srep08034

[11] C. Pizzuti and S. E. Rombo, “Algorithms and tools for protein-protein interaction networks clustering, with a special focus on population-based stochastic methods,” *Bioinformatics*, vol. 30, no. 10, pp. 1343–1352, 2014. https://doi.org/10.1093/bioinformatics/btu034

[12] R. Vyas, S. Bapat, E. Jain, M. Karthikeyan, S. Tambe, and B. D. Kulkarni, “Building and analysis of protein-protein interactions related to diabetes mellitus using support vector machine, biomedical text mining and network analysis,” *Comput. Biol. Chem.*, vol. 65, pp. 37–44, 2016. https://doi.org/10.1016/j.compbiolchem.2016.09.011

[13] H. Zhou *et al.*, “Improving neural protein-protein interaction extraction with knowledge selection,” *Comput. Biol. Chem.*, vol. 83, no. May, p. 107146, 2019. https://doi.org/10.1016/j.compbiolchem.2019.107146

[14] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, vol. 13-17-Augu, pp. 785–794. https://doi.org/10.1145/2939672.2939785

[15] A. Gupta, K. Gusain, and B. Popli, “Verifying the Value and Veracity of eXtreme Gradient Boosted Decision Trees on a Variety of Dataset,” in *2016 11th International Conference on Industrial and Information Systems (ICIIS)*, 2015, pp. 457–462. https://doi.org/10.1109/ICIINFS.2016.8262984

[16] I. Babajide Mustapha and F. Saeed, “Bioactive Molecule Prediction Using Extreme Gradient Boosting,” *Molecules*, vol. 21, no. 8, pp. 1–11, 2016. https://doi.org/10.3390/molecules21080983

[17] T. W. Valente, K. Coronges, C. Lakon, and E. Costenbader, “How Correlated Are Network Centrality Measures?,” *Connect. (Tor).*, vol. 28, no. 1, pp. 16–26, 2008.

[18] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, “Computing classic closeness centrality, at scale,” in *COSN 2014 - Proceedings of the 2014 ACM Conference on Online Social Networks*, 2014, pp. 37–49. https://doi.org/10.1145/2660460.2660465

[19] S. Oldham, B. Fulcher, L. Parkes, A. Arnatkeviciūtė, C. Suo, and A. Fornito, “Consistency and differences between centrality measures across distinct classes of networks,” *PLoS One*, vol. 14, no. 7, pp. 1–23, 2019. https://doi.org/10.1371/journal.pone.0220061

[20] J. Zhong, Y. Sun, W. Peng, M. Xie, J. Yang, and X. Tang, “XGBFEMF: An XGBoost-Based framework for essential protein prediction,” *IEEE Trans. Nanobioscience*, vol. 17, no. 3, pp. 243–250, 2018. https://doi.org/10.1109/TNB.2018.2842219

[21] J. H. Friedman, “Stochastic gradient boosting,” *Comput. Stat. Data Anal.*, vol. 38, no. 4, pp. 367–378, 2002. https://doi.org/10.1016/S0167-9473(01)00065-2

[22] T. Saito and M. Rehmsmeier, “The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets,” *PLoS One*, vol. 10, no. 3, pp. 1–21, 2015. https://doi.org/10.1371/journal.pone.0118432

[23] C. Marzban, “The ROC curve and the area under it as performance measures,” *Weather Forecast.*, vol. 19, no. 6, pp. 1106–1114, 2004. https://doi.org/10.1175/825.1

[24] X. Ying, “An Overview of Overfitting and its Solutions,” *J. Phys. Conf. Ser.*, vol. 1168, no. 2, 2019. https://doi.org/10.1088/1742-6596/1168/2/022022

[25] M. Sokolova, S. Szpakowicz, and N. Japkowicz, “Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Perfor,ance Evaluation,” *AI 2006 Adv. Artif. Intell.*, vol. 4304, no. 1, pp. 1015–1021, 2006. https://doi.org/10.1007/11941439_114