Available online at http://scik.org
Commun. Math. Biol. Neurosci. 2020, 2020:48
https://doi.org/10.28919/cmbn/4653
ISSN: 2052-2541

PHOSPHORYLATION SITE PREDICTION USING GRADIENT TREE BOOSTING

BHARUNO MAHESWORO[1,*], TJENG WAWAN CENGGORO[1,2], ARIF BUDIARTO[1,2], FAVORISEN ROSYKING LUMBANRAJA[3], BENS PARDAMEAN[1,4]

[1]Bioinformatics and Data Science Research Center, Bina Nusantara University, Jakarta, Indonesia

[2]Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia

[3]Department of Computer Science, Faculty of Mathematics and Natural Science, University of Lampung, Bandar Lampung, Indonesia

[4]Computer Science Department, BINUS Graduate Program - Master of Computer Science Program, Bina Nusantara University, Jakarta, Indonesia

Abstract: As one of the most important Post-Translational Modification (PTM), phosphorylation is responsible for cellular signaling pathways and activation of enzymes. With current computational power and algorithm, it is possible to process big data, especially biomedical data, to find a complicated pattern with reasonable computation time. Computational approach for phosphorylation site prediction is more time-efficient and need fewer resources compared to traditional. However, the accuracy of current computational methods for phosphorylation site prediction still needs to be improved. This paper aims to create a computational method for phosphorylation site prediction with better classification performance compared to previous studies. The data used in this research to train the XGBoost models are extracted features from 2 different databases from the previous studies. The test result show that our model gave the highest accuracy on 4 out of 6 datasets. To extend our research, the XGBoost model was retrained which focused on 100 most important features from previous experiment. However, the result does not imply that it has a better result compared to our first models. As the result showing that our models gave better accuracy compared to the previous studies in most of the datasets, we can conclude that XGBoost model is better in predicting phosphorylation sites compared to other methods.

[*] Corresponding author

E-mail address: bharuno.mahesworo@binus.edu

Received April 22, 2020

Keywords: phosphorylation; post-translational modification; prediction; XGBoost; gradient tree boosting; protein sequence; proteomics.

2010 AMS Subject Classification: 92C40.

1 INTRODUCTION

Phosphorylation of a molecule is the addition of a phosphoryl group (PO3-) to an organic molecule. While the removal of a phosphoryl group from a molecule is called dephosphorylation. Phosphorylation in protein is a crucial phenomenon since the characteristic of the proteins can be altered after they are formed, for example, activate or deactivate enzyme, metabolite sugar, and store or release energy [1]–[4]. This protein phosphorylation occurs in three amino acids, Serine, Threonine, and Tyrosine, and are carried out by enzymes, such as kinases, phosphotransferases [5].

Formerly, identifying phosphorylation sites methods are commonly using an experimental approach such as mass spectrometry (MS/MS) [6]. With the computational power available today, it is more plausible and convenience to do computational approach for phosphorylation site prediction than in the past. Besides, the experimental approach requires specific and expensive equipment, advance skill and technique, and intensive labor. Therefore, a computational approach for Phosphorylation site prediction is now becoming more popular. In 2011, Trost and Kusalik summarized numerous computational approach of phosphorylation site prediction [7].

This paper is further research on our previous research [8]. We introduce a new approach to predict phosphorylation site by utilizing machine learning algorithms under the Gradient Boosting framework using XGBoost library [9]. The aim of this research is to achieve better classification performances with fewer steps compared to our previous methods.

2 RELATED WORKS

This study is the continuation of Lumbanraja et al study in 2018 [8] and their study in 2019 [10]. In 2018, Lumbanraja et al proposed a new phosphorylation site prediction method for non-kinase-specific, which gave better accuracy compared to the state-of-the-art methods. The method used feature extraction to enrich their datasets, besides fixed-length polypeptide sequences. The features, along with the fixed-length polypeptide sequences are then ranked according to its significance using random forest [11] and Gini Impurity Index (GII). Support Vector Machine was implemented to classify the phosphorylation site.

In 2019, Lumbanraja et al continued their study in phosphorylation site prediction [10]. In this research, a deep neural network was implemented as the classification method. By conducting this, the feature extraction and feature selection step were not implemented. Although the performance of this method was below their first method, the performance of the deep neural network was better compared to the methods that come before Lumbaranraja et al [8].

2.1 Gradient Tree Boosting and XGBoost

Gradient Boosting Machine [12] is a machine learning algorithm which uses an ensemble of weak learners. The algorithm is popularly used with decision trees as the weak learners. As the name suggests, a gradient boosting machine utilizes gradient descent to optimize an ensemble model with boosting paradigm. In other words, the algorithm builds a weak learner in iterative fashion, which the learner reduces the error gradient of the previous ensemble model. Thus, given a differentiable loss function $\boldsymbol{L(y, \hat{y})}$ the new weak learner is fitted to the previous error $\Delta\mathbf{y}$ computed with the following formula:

$$\Delta \mathrm{y} = -\frac{\partial L(y, \hat{y}_{t-1})}{\partial \hat{y}_{t-1}} \tag{1}$$

where y is the ground truth and $\boldsymbol{\hat{y}_t}$ is the prediction of the $\boldsymbol{t^{th}}$ ensemble model. Afterwards, the prediction of the next ensemble model is calculated as follows:

$$\hat{y}_t = \hat{y}_{t-1} + \alpha f_t(x) \tag{2}$$

where $\boldsymbol{f_t(x)}$ is the t^{th} weak learner prediction given $\boldsymbol{x}$ as input data. For a better generalization, the new prediction is smoothed by multiplying it with a learning rate $\boldsymbol{\alpha}$. The value of $\boldsymbol{\alpha}$ can be between 0 and 1 to control the effect of the new weak learner to the previous prediction. This process is repeated for each iteration until the error is sufficiently low. Figure 1 illustrates the process of GBM that uses a decision tree as the weak learners, popularly known as Gradient Boosted Trees (GBT). In the figure, a decision tree is added to the ensemble for each iteration to decrease the error of the ensemble.

Among the implementation of GBM, XGBoost is currently the most popular for use in many applications. It successfully records the state-of-the-art performance in many machine learning challenges [9]. It is a family of GBT that uses various regularization techniques such as L1, L2, and tree pruning. Most of the regularization techniques were originally designed to optimize the speed of the algorithm. However, they are happened to also contribute to its superior performance among other popular machine learning algorithms for classification. It is noticeably powerful to model structured data, which is proven by the fact that it is currently the state-of-the-art in numerous datasets with a structured format. This research is conducted by using the XGBoost library, an open-source package that is used as a scalable machine learning system for tree boosting [9].

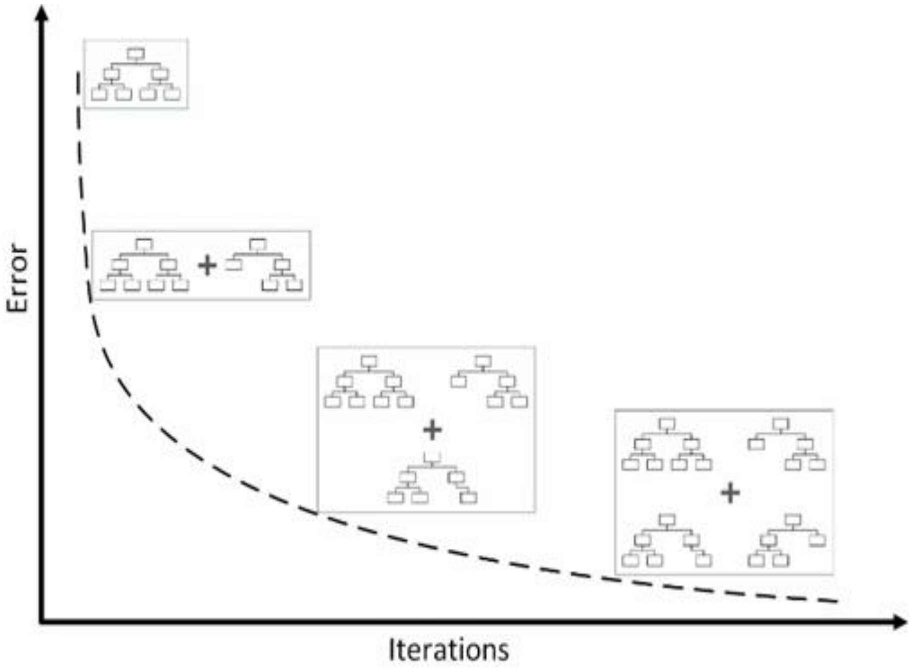

Fig. 1. Illustration of Gradient Tree Boosting algorithm

Another appealing feature of XGBoost is its straightforward implementation of feature importance. Because it is essentially an ensemble of decision trees, standard feature importance method for a decision tree can be employed. For instance, the feature importance can be calculated by summing the decrease in node impurity for each decision tree in the ensemble. The node impurity can be measured with various metrics such as entropy or Gini Impurity Index. In practice, this feature importance is used afterward to select features for developing a more powerful model that learns only from the most important features.

3 Materials and Method

3.1 Dataset

The datasets used in this study are the same polypeptide sequences datasets that were used in Lumbanraja studies [8]. The sequences are composed of 9 amino acids where the amino acids in the middle are the possible location of phosphorylation, Serine (S), Threonine (T), or Tyrosine (Y). While the first to fourth and sixth to ninth amino acids are the amino acids adjacent to the phosphorylatable residues in the protein amino acids sequences. The sequences are generated from Phospho.ELM (P.ELM) database version 9 [13] and PhosPhAt (PPA) database [14].

The sequences are labelled as "positive" and "negative" sequences. Positive sequences are the sequences that are known as phosphorylated. Then, these positive and negative sequences with 80% to 100% similarities are removed from the datasets to decrease redundancy using “skipredundant” [15]. The parameters used for reducing redundancy are:

- Acceptable percentage of similarity: 0% - 20%
- Value for gap opening penalty: 10
- Gap extension penalty: 0.5

The sequences are then classified according to its database source and phosphorylatable residue. Table 1 shows the size of each dataset. As seen in Table 1, Most phosphorylation sites occur in Serine and Threonine residue. On the other hand, Tyrosine is the least phosphorylatable residue.

Table 1. Dataset size of Phosphorylation site in P.ELM and PPA dataset

Dataset		**P.ELM**	**PPA**
Serine	Positive	1554	307
	Negative	1543	307
Threonine	Positive	707	68
	Negative	453	68
Tyrosine	Positive	267	51
	Negative	226	51

3.2 Method

The workflow of this research follows diagram flow in Figure 1. In this research, various features were extracted from the fixed sequence of amino acids using PROFEAT (2016) [16], PSI-BLAST [17], and protr [18], as it was conducted in the previous research8. Sixteen aspects were extracted from the amino acids sequence which is then separated into 2256 features. Which are: Amino Acid Composition (AAC), Dipeptide Composition (DPC), Moran Autocorrelation Descriptors (MORAN), Composition, Transition, Distribution (CTD), Quasi-Sequence-Order Descriptors (QSO), Amphiphilic Pseudo-Amino Acid Composition(APAAC), Total Amino Acid

Properties (AAP), BLOSUM and PAM Matrices for the 20 Amino Acid (BLOSUM), Amino Acid Properties Based Scales Descriptor (Protein Fingerprint) (ProtFP), Scales-based Descriptor derived by Principal Components Analysis (SCALES), Scales-based Descriptor derived by Multidimensional Scaling (MDDSCALES), Conjoint Triad Descriptors (CTriad).

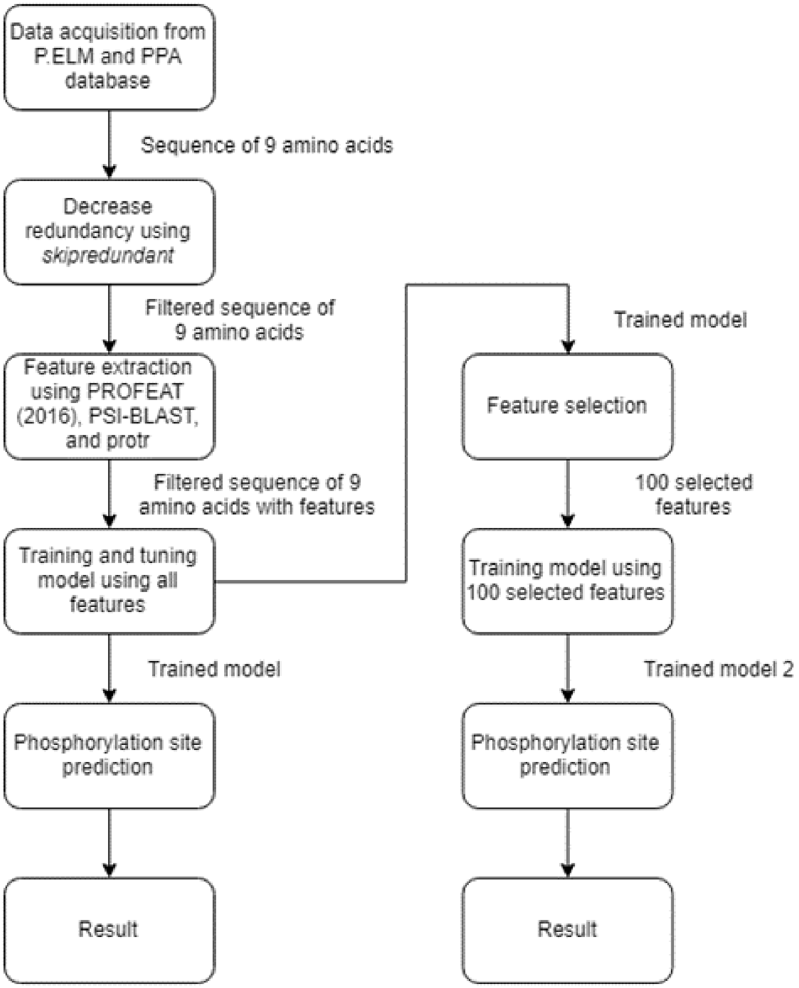
Data acquisition from P.ELM and PPA database
Sequence of 9 amino acids
Decrease redundancy using *skipredundant*
Filtered sequence of 9 amino acids
Feature extraction using PROFEAT (2016), PSI-BLAST, and protr
Filtered sequence of 9 amino acids with features
Training and tuning model using all features
Trained model
Phosphorylation site prediction
Result
Trained model
Feature selection
100 selected features
Training model using 100 selected features
Trained model 2
Phosphorylation site prediction
Result

Fig. 2. Workflow for phosphorylated site classification in this research

Each dataset is then separated into three groups of data, training dataset, validation dataset and testing dataset with the composition of 70%, 20% and 10% respectively. The training data is used to train the gradient tree boosting algorithm to predict the phosphorylation location based on the extracted features. To find the best hyperparameter for the model, we used the Grid Search method. Grid search is the process of setting hyperparameters to determine the optimal value for a given model. The hyperparameter values used for Grid Search are listed in Table 2.

The validation dataset is used to validate the performance of the models when the authors tuned the algorithm. 10-folds cross-validation is used to validate the models and to minimize bias. Then the testing dataset is used to provide the result of the models.

Table 2. Hyperparameter selection for Grid Search

Parameter	**Value**			
max_depth	3	4	5	6
learning_rate	0.005	0.01	0.05	NA
subsample	0.5	0.7	1	NA
n_estimators	500	1000	1500	NA

The 100 of the most important features of the trained model is then used to train the second model. The purpose of these steps is to have a comparison model where they read all the data and focused only on important features.

3.3 Evaluation and Comparison

To measure and compare our method to the previous methods, we use the same evaluation techniques that were used in Lumbanraja studies [8]. There are five evaluation parameters. The first one is the method accuracy. Method accuracy can be calculated by dividing the sum of the true positive and true negative with the sum of the total N. The second and third parameters are sensitivity and specificity. Sensitivity is the capability to predict correctly those that are phosphorylatable (true positive rate), whereas specificity is the ability of the method to correctly identify those which are not phosphorylatable (true negative rate).

The last two parameters are F1 score and Matthews correlation coefficient. F1 score mostly used and gained its popularity in the machine learning research area. This parameter can be calculated by dividing the number of true positive with the sum of the number of true positive, the number of false positive and the number of false negative. Matthews correlation coefficient or commonly known as MCC is firstly introduce by Matthews in 1975 for comparing the secondary structure of proteins [19]. This parameter later become widely used in biomedical community especially in protein research [20]–[24]. Since this parameter is the most relevant to our data and research area, the discussion and analysis of this paper is more referring to MCC rather than F1 score. The evaluation techniques used in this research are formulated below:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

$$Sensitivity = \frac{TP}{TP+FN} \tag{4}$$

$$Specificity = \frac{TN}{TN+FP} \tag{5}$$

$$F1\ Score = \frac{TP}{TP+FP+FN} \tag{6}$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{7}$$

4 RESULTS

Result of our first experiments, which use all features to be inputted in the XGBoost algorithm, and comparison to the previous study, Lumbanraja et al 2018 [8] as Method 1 and Lumbanraja et al 2019 [10] as Method 2, are listed on Table 3, 4 and 5. The results of each dataset with the highest accuracy and MCC are listed in bold.

Table 3. Experiment result on Serine dataset

Parameter	**P.ELM**			**PPA**		
	Method 1	**Method 2**	**XGBoost**	**Method 1**	**Method 2**	**XGBoost**
Accuracy	0.9646	0.9146	**0.9661**	**0.8766**	0.8109	0.8211
AUC	0.9646	0.9185	0.9660	0.8766	0.8104	0.8270
Sensitivity	0.9715	0.9305	0.9689	0.8290	0.8247	0.7612
Specificity	0.9577	0.9065	0.9630	0.8611	0.7678	0.8929
F1	0.9650	0.9197	0.9674	0.8786	0.8111	0.8226
MCC	0.9298	0.8385	**0.9321**	**0.7562**	0.6211	0.6532

Table 4. Experiment result on Threonine dataset

Parameter	**P.ELM**			**PPA**		
	Method 1	**Method 2**	**XGBoost**	**Method 1**	**Method 2**	**XGBoost**
Accuracy	0.9222	0.8733	**0.9267**	**0.9118**	0.8242	0.8214
AUC	0.9222	0.8708	0.9250	0.9118	0.8288	0.8333
Sensitivity	0.9264	0.8768	0.9343	0.8823	0.8034	1.0000
Specificity	0.9157	0.8648	0.9158	0.9412	0.8542	0.6667
F1	0.9354	0.8939	0.9377	**0.9091**	0.8076	0.8387
MCC	0.8387	0.7362	**0.8488**	**0.825**	0.6597	0.6939

Table 5. Experiment result on Tyrosine dataset

Parameter	**P.ELM**			**PPA**		
	Method 1	**Method 2**	**XGBoost**	**Method 1**	**Method 2**	**XGBoost**
Accuracy	0.8019	0.7564	**0.8586**	0.5784	0.6409	**0.6667**
AUC	0.7984	0.7602	0.8613	0.5784	0.6565	0.6944
Sensitivity	0.8381	0.7272	0.8077	0.5295	0.6536	0.5000
Specificity	0.7588	0.7933	0.9149	0.6274	0.6595	0.8888
F1	0.8205	0.7609	0.8571	0.5567	0.6339	0.6316
MCC	0.6043	0.5186	**0.7235**	0.1576	0.3120	**0.4082**

The results shows that XGBoost gave the best accuracy and MCC on all amino acids, Serine, Threonine and Tyrosine in P.ELM dataset. Despite only have little lead in Serine and Threonine, XGBoost gave 5% better accuracy in Tyrosine P.ELM datasets, compared to previous methods. On the PPA dataset, XGBoost shows similar results compared to method 2. It shows a slight lead in performance on Tyrosine and Serine. However, Method 1 still has a huge lead on Serine and Threonine PPA dataset, with the accuracy of 87.66% and 91.18%.

Analyzing the result of the XGBoost alone, the accuracy and MCC each amino acid in P.ELM dataset are positvely correlated with the number of the amino acids sequence. On the other hand, the accuracy and MCC each amino acid in PPA dataset also show a correlation with the number of the protein sequence. Table 6 show the accuracy, MCC and the number of amino acid sequence for each corresponding amino acid. Based on Table 6 data, the correlation between accuracy and the number of amino acid sequence is 0.76, while the correlation between MCC and the number of amino acid sequence is 0.77. Both correlations can be classified as strong correlation.

Table 6. Accuracy, MCC and number of amino acid sequence

Dataset	**Accuracy**	**MCC**	**Number of Amino Acid Sequence**
P.ELM Serine	0.9661	0.9321	3097
P.ELM Threonine	0.9267	0.8488	1160
P.ELM Tyrosine	0.8586	0.7235	493
PPA Serine	0.8211	0.6532	614
PPA Threonine	0.8214	0.6939	136
PPA Tyrosine	0.6667	0.4082	102

The result of our first experiment also gave us 100 most important features for each amino acid on each dataset. These features were then used to train our second model which focused on those features. The result of our second experiment is shown in Table 7, 8, and 9.

Table 7. Second experiment result on Serine dataset

Parameter	**P.ELM**		**PPA**	
	All features	**100 features**	**All features**	**100 features**
Accuracy	**0.9661**	0.9581	**0.8211**	0.7967
AUC	0.9660	0.9584	0.8270	0.8002
Sensitivity	0.9689	0.9503	0.7612	0.7612
Specificity	0.9630	0.9664	0.8929	0.8393
F1	0.9674	0.9592	0.8226	0.8031
MCC	**0.9321**	0.9162	**0.6532**	0.5982

Table 8. Second experiment result on Threonine dataset

Parameter	**P.ELM**		**PPA**	
	All features	**100 features**	**All features**	**100 features**
Accuracy	0.9267	**0.9310**	0.8214	**0.8571**
AUC	0.9250	0.9294	0.8333	0.8667
Sensitivity	0.9343	0.9366	1.0000	1.0000
Specificity	0.9158	0.9222	0.6667	0.7333
F1	0.9377	0.9433	0.8387	0.8667
MCC	0.8488	**0.8555**	0.6939	**0.7488**

Table 9. Second experiment result on Tyrosine dataset

Parameter	**P.ELM**		**PPA**	
	All features	**100 features**	**All features**	**100 features**
Accuracy	0.8586	**0.8687**	**0.6667**	**0.6667**
AUC	0.8613	0.8719	0.6944	0.6806
Sensitivity	0.8077	0.8077	0.5000	0.5833
Specificity	0.9149	0.9362	0.8888	0.7778
F1	0.8571	0.8660	0.6316	0.6667
MCC	0.7235	**0.7460**	**0.4082**	0.3611

In P.ELM dataset, the second model gave a slightly better performance on Threonine and Tyrosine dataset compared to the first model. While the first model still has little lead on Serine, the biggest dataset. In the PPA dataset, the second model only lead on Threonine dataset. In Tyrosine dataset, the second model has the same accuracy, 66.67%, with the first model. However, the first model has better MCC on that particular dataset.

5 DISCUSSION

The comparison in Table 3 shows that our method gave better results on four out of six datasets. All the experiment on the P.ELM dataset gave a better result. However, the first model of XGBoost does not give a better result on Serine and Threonine in PPA dataset.

Despite losing on 2 datasets in the PPA database, the first model shows a good lead on Tyrosine dataset, the smallest dataset in this study with only 102 polypeptide sequences. Small dataset often became the main problem in prediction algorithm development. But, the XGBoost model gave reasonable accuracy, where other previous models could not. On standalone XGBoost analysis, the accuracy and MCC of the model showing a positive correlation with the size of the dataset.

On the second experiment, XGBoost model which focused on 100 most important features were compared to XGBoost model which focused on all features. The results show that the model that focused on 100 most important features were only slightly better on 3 out of 6 datasets, and slightly lower on the rest of it. The size of the dataset does not appear to give influence on the accuracy differences.

Moreover, XGBoost of 100 feature selection gave better result compared to XGBoost of all features only on three out of six datasets. This outcome may suggest that XGBoost that focused on important features does not give a better result. In the second experiment, the training time of the models were faster compared to the first experiment. This faster processing time is due to the less variables that need to be calculated by the processing unit. Beside the results, the method that we used is also less complicated compared to our previous method [8], [10].

6 CONCLUSION

In this paper, we introduce a new approach by using XGBoost to classify non-kinase-specific phosphorylation site. Based on the results of 6 different amino acid residue datasets from

P.ELM and PPA database, XGBoost model shows better classification results in a small dataset, a slight lead in large datasets, and lower accuracy on medium-sized dataset compared to the previous methods.

On small dataset, XGBoost perform better compared to the previous method. Small dataset often became the main problem in machine learning. The performance of the XGBoost model in this study shows a positive correlation with the size of the training dataset. However, the model still manages to deliver reasonable result from small dataset.

The results on second experiment which focused on 100 most important do not imply better accuracy. However, the processing time of the second experiment is a lot faster compared to the first experiment due to fewer variables that need to be calculated.

7 ACKNOWLEDGMENT

This research is a collaboration between the Department of Computer Science, University of Lampung, and Bioinformatics and Data Science Research Center (BDSRC), Bina Nusantara University. All analysis tools were provided by BDSRC. While the data provided in the study were provided by Department of Computer Science, Faculty of Mathematics and Natural Science, University of Lampung.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] P. Cohen, The regulation of protein function by multisite phosphorylation – a 25 year update, Trends Biochem. Sci. 25 (2000), 596–601.

[2] P. Vlastaridis, A. Papakyriakou, A. Chaliotis, E. Stratikos, S.G. Oliver, G.D. Amoutzias, The Pivotal Role of Protein Phosphorylation in the Control of Yeast Central Metabolism, G3: Genes, Genomes, Genetics, 7 (2017) 1239–1249.

[3] A.P. Oliveira, U. Sauer, The importance of post-translational modifications in regulating Saccharomyces cerevisiae metabolism, FEMS Yeast Res. 12 (2012), 104–117.

[4] F. Tripodi, R. Nicastro, V. Reghellin, P. Coccetti, Post-translational modifications on yeast carbon metabolism: Regulatory mechanisms beyond transcriptional control, Biochimica et Biophysica Acta (BBA) - General Subjects. 1850 (2015), 620–627.

[5] T. Hunter, L. Jolla, H. W. Longfellow, Signaling — 2000 and Beyond, 100 (2000), 113–127.

[6] R.H. Newman, J. Zhang, H. Zhu, Toward a systems-level view of dynamic phosphorylation networks, Front. Genet. 5 (2014). https://doi.org/10.3389/fgene.2014.00263.

[7] B. Trost, A. Kusalik, Computational prediction of eukaryotic phosphorylation sites, Bioinformatics, 27 (21) (2011), 2927–2935.

[8] F.R. Lumbanraja *et al.*, Improved Protein Phosphorylation Site Prediction by a New Combination of Feature Set and Feature Selection, J. Biomed. Sci. Eng. 11 (6) (2018), 144–157.

[9] T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, San Francisco California USA, 2016: pp. 785–794.

[10] F.R. Lumbanraja, B. Mahesworo, T.W. Cenggoro, A. Budiarto, B. Pardamean, An Evaluation of Deep Neural Network Performance on Limited Protein Phosphorylation Site Prediction Data, Procedia Computer Science. 157 (2019) 25–30.

[11] L. Breiman, Random Forest, Mach. Learn. 45 (1) (2001), 5–32.

[12] B.J.H. Friedman, 1999 Reitz Lecture, Ann. Stat. 29 (5) (2001), 1189–1232.

[13] H. Dinkel, C. Chica, A. Via, C.M. Gould, L.J. Jensen, T.J. Gibson, F. Diella, Phospho.ELM: a database of phosphorylation sites--update 2011, Nucleic Acids Res. 39 (2011), D261–D267.

[14] J.L. Heazlewood, P. Durek, J. Hummel, J. Selbig, W. Weckwerth, D. Walther, W.X. Schulze, PhosPhAt: a database of phosphorylation sites in Arabidopsis thaliana and a plant-specific phosphorylation site predictor, Nucleic Acids Research. 36 (2007), D1015–D1021.

[15] K. Sikic, O. Carugo, Protein sequence redundancy reduction: comparison of various methods, Bioinformation, 5 (6) (2010), 234–239.

[16] H.B. Rao, F. Zhu, G.B. Yang, Z.R. Li, Y.Z. Chen, Update of PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence, Nucleic Acids Res. 39 (2011), W385–W390.

[17] N.H. Bergman, Comparative Genomics: Volumes 1 and 2. Humana Press, Totowa, 2007.

[18] N. Xiao, D.-S. Cao, M.-F. Zhu, Q.-S. Xu, protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences, Bioinformatics. 31 (2015), 1857–1859.

[19] B.W. Matthews, Comparison of the predicted and observed secondary structure of T4 phage lysozyme, Biochimica et Biophysica Acta (BBA) - Protein Structure. 405 (1975), 442–451.

[20] J. Yang, A. Roy, Y. Zhang, Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment, Bioinformatics. 29 (2013), 2588–2595.

[21] J. Song, K. Burrage, Z. Yuan, T. Huber, Prediction of cis/trans isomerization in proteins using PSI-BLAST profiles and secondary structure information, BMC Bioinformatics, 7 (2006), 124.

[22] N. Huang, I. Lee, E.M. Marcotte, M.E. Hurles, Characterising and Predicting Haploinsufficiency in the Human Genome, PLoS Genet. 6 (2010), e1001154.

[23] L. Shi *et al.*, The MicroArray Quality Control (MAQC)-II study of common practices for the development and

validation of microarray-based predictive models, Nat. Biotechnol. 28 (8) (2010), 827–838.

[24] G. Liu, T.R. Mercer, A.-M.J. Shearwood, S.J. Siira, M.E. Hibbs, J.S. Mattick, O. Rackham, A. Filipovska, Mapping of Mitochondrial RNA-Protein Interactions by Digital RNase Footprinting, Cell Rep. 5 (2013) 839–848.