Applied Biological Research 25(2): 143-150; (2023)
DOI: 10.5958/0974-4517.2023.00017.4

EXPLOITING THE SEQUENCE AND EVOLUTIONARY INFORMATION FOR THE IDENTIFICATION OF VIRULENCE FACTORS

A Shivram, Piyusha Sharma and Abhigyan Nath*

[1]*Department of Biochemistry, Pt. Jawahar Lal Nehru Memorial Medical College, Raipur - 492 001, India*
*****e-mail: *abhigyannath01@gmail.com*

(Received 30 September, 2022; accepted 3 March, 2023)

ABSTRACT

The pathogenicity of a bacterium depends on its virulence factors (VF); and predicting the VF allows to understand the pathogenesis of infection. It can be used to identify a possible point of interference for disease treatment or vaccination. In present work, a machine-learning based prediction model for (VF's was developed, using two different types of feature extraction: classical sequence-based features and evolutionary information-based features for the representation of VFs and non-VFs protein sequences. For splitting data into training and testing sets, uniform sampling based approach was used (Kennard-Stone algorithm) to create diversified and representative training/testing sets. For accurate prediction of VF's, different machine learning algorithms (Random forest and Gradient boosting machine) and deep learning algorithms were used and further analysed using model agnostic interpretation methods. The highest accuracy of 84.5% was obtained by RF with PSSM feature set, followed by GBM with PSSM feature set (accuracy 83.9%). With feature fusion, the best performance evaluation metric was obtained by GBM with 92.0% sensitivity, 86.6% specificity, 89.3% accuracy, 0.788 mcc and 0.953 AUC on 10-fold cross validation. The feature importance was captured by Variable importance plots (VIP) and Shapely plots. Evolutionary information based features i.e. PSSM and PSE-PSSM are the two most important features in discriminating the VFs from non VFs.

Keywords: Deep learning, gradient boosting machines, Random forest, variable importance, virulence factors

INTRODUCTION

The spread of infectious diseases poses a serious threat to the human and animal health. Bacteria exist both inside and on the surface of human body, especially on mucous membranes and skin. Some of these bacteria are harmless and some are beneficial. The pathogenic bacteria colonize, invade, and damage the host cells thus cause illness. The pathogenicity of a bacterium depends on its virulence factors (VF). Virulence refers to the quantitative measure of pathogenicity or likelihood of a pathogen causing infection (Kaur and Forster, 2013), the examples are toxins, capsules, and surface-associated proteins. In past decade, an increase in the resistance to antibiotics has been reported globally (WHO, 2020). Advances in the field of genomics and bioinformatics have made it possible to use sequence data as raw material which can be processed by various computational tools to understand the characteristics of a disease (Tanwer *et al.,* 2020). Prediction of VF allows the understanding of pathogenesis of infection so can be used to identify a possible point of interference for treatment or vaccination.

The software program SPAAN (Sachdeva *et al*., 2004) was developed previously to predict a class of VF, adhesins, and adhesin-like proteins using a neural network and it achieved a considerable

accuracy. Another server MP3 (Gupta *et al*., 2014) was developed using support vector machines (SVM) and Hidden Markov Models (HMMs) approach for the prediction of pathogenic proteins in both genomics and metagenomics datasets. Recently, a Deep-learning based method has been developed for using sequence information to identify VFs from genome sequence data called 'DeepVF' (Xie *et al*., 2020). In present study, we used an up-to-dated benchmark dataset from DeepVF and extracted classical sequence-based and evolutionary information-based features to represent the characteristic features of VFs. To make a balanced dataset, we used representative sampling method, and split the data into testing and training datasets. These models were trained using classical machine learning algorithms, including random forest (RF) (Liaw and Wiener, 2002) and Gradient boosting machines (GBM) (Ke *et al.,* 2017). For Deep-learning, Deep neural networks (DNNs) (Chien, 2019) were used (with and without dropout). The present study was aimed to develop an accurate predictor for VFs, and use trained machine learning models (model agnostic methods) for better understanding of the sequence propensities in VFs.

MATERIALS AND METHODS

In present study we used the dataset as used in DeepVF (Xie *et al.*, 2020). The data set comprised of 8486 samples (3576 VFs and 4910 non-VFs). Representative sampling method was used to split the dataset into training and testing datasets. Among these, 576 VFs and 576 non-VFs were selected as independent test dataset, while the remaining were used as training dataset. To solve the data imbalance problem, we constructed balanced training datasets (each having 3000 VFs and 3000 non-VFs) by uniform sampling using Kennard-Stone algorithm (Kennard and Stone, 1969), selecting the same number of non-VFs as that of VFs.

Feature extraction

In this study, we used two different sets of features classical sequence-based features and evolutionary information-based features for the representation of VFs and non-VFs protein sequence. The features calculated for each protein sequence in the dataset were classical sequence based features which represent the information regarding physical, chemical and physiochemical properties of amino acids and proteins such as hydrophobicity or polarity.

Pseudo-amino acid composition (PAAC): PAAC has successfully been used in many bioinformatics applications (Chou, 2009) to improve the prediction quality of subcellular localization of proteins and other attributes based on their sequence.

Evolutionary features: Evolutionary information-based features were extracted based on the probability of substitution of amino acids through their evolution process. Position-specific scoring matrices (PSSM) were used as a means of evolutionary information in this study (Dehzangi *et al.*, 2015; Khanh Le *et al.,* 2019). PSSM profiles were obtained using position specific iterative-blast (PSI-BLAST) (Jones and Swindells, 2002). It provides a means for detecting the distant relationships between query proteins, and works on the iterative implementation of BLAST algorithm. Multiple sequence alignment was constructed from the sequences obtained after each round of PSI-BLAST, which was used for the construction of PSSM profiles. N length of sequence generates N x 20 matrix based on 20 amino acid types. We used training set sequences as database for search algorithm. Accordingly, different types of evolutionary information-based features were represented through three different types of PSSM, which included PSSM composition, DPC-PSSM, and PSE-PSSM, respectively.

PSSM composition: A PSSM is a matrix based on the amino acid frequencies at every position of a multiple alignment. PSSM expresses the patterns inherent in a multiple sequence alignment of a set of homologous sequences (Altschul, 1997; Gromiha, 2010). These profiles were used to match the query

sequences from the database against the sequences in alignment table, giving higher weight to positions that are conserved than to those that were available.

Dipeptide PSSM composition (DPC-PSSM): The dipeptide composition provides information about the fraction of amino acid as well as the local order of amino acids. In DPC -PSSM, the values located in two consecutive rows and corresponding two different columns in the L X 20 PSSM matrix are first multiplied then results summed up, followed by division by L-1 (length of protein sequence-1) resulting in 400 dimensional feature vector.

Pseudo-position - specific score matrix (PSE-PSSM): The PSE-PSSM feature (Chou and Shen, 2007) was used for this work. These features are adequate to explore the evolutionary information and the sequence-order information embedded in PSSMs. In this, the first 20 numbers are the mean of 20 columns in PSSM matrix, and the next numbers for each column are the mean squares of difference between the elements of row i and i + lag in this column. The lag value varies between 1 and 15 so constructing the final feature vector of length 320.

Representative and diversified training set

If a dataset is highly imbalanced when trained with the classifiers, then poor performance is expected. For developing an unbiased prediction model, the training and testing sets should be representative and diversified in terms of patterns present in dataset (Nath and Subbiah, 2015). If the training set is not representative, it may influence the testing set leading to a poor generalization of results for machine learning algorithms. Representative testing data facilitates the reliable estimation of generalization ability of a machine learning algorithms. Uniform design-based sampling is commonly used for representative sample selection. The main objective of uniform design-based sampling is the uniform coverage of data space (*e.g.* Kennard-Stone algorithm).

Kennard-Stone (KS) algorithm

The KS algorithm starts by finding the two samples which are farthest apart using Euclidean distance. Subsequent points are selected from the sample which have greatest separation distance from the selected samples. The process results in the selection of a desired number of samples for the representative training set. The closest sample is included in the representative sample subset; samples are selected to smoothly fill the data space. The remaining samples are placed into the test set. The major advantage of representative sampling using KS algorithm is giving a chance to the different diverse patterns present in the dataset to be represented in the training set enabling the machine learning algorithms for complete learning.

Model training using ensemble learning

In this study, we used three different algorithms *viz*., Random forest (RF), Gradient boosting machines (GBM), Deep learning neural networks without dropout (DLNNWOD), and Deep learning neural network with dropout (DLNNWD) to develop a discriminatory model for VFs and non-VFs. The following hyper parameters were used for the algorithms:

RF: ntrees = (100, 500, 1000 and 1500), max depth = (5, 10, 15, 20 and 25) min rows = (5, 8, 10, 12 and 15).

GBM: ntrees = (100, 500, 1000 and 1500), max depth = (5, 10, 15, 20 and 25), min rows = (5, 8, 10, 12 and 15).

DLNNWOD: Activation function (Rectifier, Maxout and Tanh), Hidden layer (10, 50, 100, and 450), (10, 20, 50 and 100), (100, 50, 20 and 10), (450, 100, 50 and 10), L1 regularization (0.00001 and 0.0001), L2 regularization (0.00001 and 0.0001), input dropout ratio (0, 0.1, 0.2 and 0.3).

DLNNWD: Activation function (Rectifier with Dropout, Maxout with Dropout, Tanh with Dropout), Hidden layers (10, 50, 100 and 450), (10, 20, 50 and 100), (100, 50, 20 and 10), (450, 100, 50 and 10), L1 regularization (0.00001 and 0.0001), L2 regularization (0.00001 and 0.0001), input dropout ratio (0, 0.1, 0.2 and 0.3) hidden dropout ratio (0.3, 0.3, 0.3 and 0.3), (0.2, 0.2, 0.2 and 0.2), (0.5, 0.5, 0.5 and 0.5).

We used the H2O package in R to implement all the classification algorithms (H2O.ai. (2020) *h2o: R Interface for H2O*. R package version 3.30.0.6. (*https://github.com/h2oai/h2o-3*). To obtain the best hyper parameters, a random grid search strategy was used.

Performance evaluation

The following performance evaluation metrics were calculated for the obtained data i.e. sensitivity (SN), specificity (SP), accuracy (ACC), Matthew's correlation coefficient (MCC). Ten-fold cross validation scheme was used to evaluate all the models. These measures are defined as follows:

$$SN = \frac{TP}{TP+FN} \quad (1)$$

$$SP = \frac{TN}{FP+TN} \quad (2)$$

$$ACC = \frac{TP+TN}{TP+FN+FP+TN} \quad (3)$$

$$MCC = \frac{(TP\times TN)-(FP\times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}, \quad (4)$$

Where TP, TN, FP and FN represent the true positives, true negatives, false positives and false negatives, respectively. We also used receiver operating characteristic (ROC) as an evaluation factor (higher the area under ROC curve, better the model performance).

RESULTS AND DISCUSSION

Four machine algorithms (RF, GBM, DLNNWOD, DLNWD) were used to generate classification models, the results of four algorithms were comparable to each other, although the tree based algorithms (RF and GBM) performed slightly better than deep neural networks.

Performance evaluation of different training models

Table 1 presents the performance evaluation metrics (PEM) of various machine learning algorithms with different individual feature sets. Very high sensitivity values were obtained with PAAC feature

Table 1: Performance evaluation matrix (PEM) of the various machine learning algorithms

Models	Encoding	Sensitivity	Specificity	Accuracy	MCC	AUC
RF	PAAC	0.879	0.431	0.657	0.353	0.730
	PSSM	0.856	0.833	0.845	0.690	0.900
	DPC PSSM	0.860	0.807	0.834	0.670	0.895
	PSE PSSM	0.867	0.796	0.832	0.667	0.896
GBM	PAAC	0.918	0.340	0.630	0.319	0.724
	PSSM	0.858	0.821	0.839	0.681	0.896
	DPC PSSM	0.855	0.802	0.829	0.662	0.890
	PSE PSSM	0.871	0.789	0.831	0.665	0.896
DLNNWOD	PAAC	0.956	0.134	0.545	0.187	0.557
	PSSM	0.875	0.760	0.817	0.642	0.877
	DPC PSSM	0.822	0.782	0.802	0.611	0.846
	PSE PSSM	0.837	0.695	0.766	0.541	0.836
DLNNWD	PAAC	0.921	0.264	0.592	0.243	0.653
	PSSM	0.838	0.832	0.835	0.672	0.891
	DPC PSSM	0.692	0.901	0.796	0.607	0.802
	PSE PSSM	0.684	0.851	0.768	0.544	0.768

set for all the machine learning algorithms, but with lower specificity values, which consequently resulted in lower overall accuracy values in comparison to the other evolutionary information based feature sets (PSSM, DPC PSSM and PSE-PSSM).

Among all the evolutionary information-based features sets, simple PSSM resulted in better specificity, accuracy, MCC and AUC values for the machine learning algorithms, except DLNNWD which has higher specificity with DPC_PSSM. The highest accuracy of 84.5% was obtained by RF with PSSM feature set followed by GBM with PSSM feature set (accuracy = 83.9%).

Performance validation on independent test

Table 2 presents the PEMs of various machine learning algorithms with different feature sets on testing set. In terms of accuracy, GBM surpassed all other algorithms with 83.7% accuracy using DPC_PSSM feature set.

Table 2: Performance evaluation matrix (PEM) of the various machine learning algorithms on testing set

Models	Encoding	Sensitivity	Specificity	Accuracy	MCC	AUC
RF	PAAC	0.783	0.598	0.645	0.333	0.764
	PSSM	0.801	0.804	0.803	0.605	0.869
	DPC PSSM	0.832	0.850	0.841	0.682	0.905
	PSE PSSM	0.872	0.803	0.834	0.671	0.903
GBM	PAAC	0.808	0.586	0.634	0.325	0.747
	PSSM	0.799	0.802	0.801	0.602	0.867
	DPC PSSM	0.853	0.823	0.837	0.676	0.896
	PSE PSSM	0.891	0.780	0.827	0.663	0.901
DLNNWOD	PAAC	1.000	0.500	0.501	0.041	0.496
	PSSM	0.800	0.831	0.815	0.631	0.862
	DPC PSSM	0.872	0.777	0.817	0.642	0.883
	PSE PSSM	0.856	0.748	0.792	0.594	0.875
DLNNWD	PAAC	0.921	0.264	0.592	0.594	0.875
	PSSM	0.750	0.579	0.620	0.282	0.713
	DPC PSSM	0.775	0.649	0.693	0.405	0.777
	PSE PSSM	0.819	0.812	0.816	0.632	0.865

Performance evaluation of combined feature set

The performance evaluation metrics for the combined feature set consisted of PAAC, PSSM and PSE-PSSM on 10-fold cross validation (Table 3). The performance of RF and GBM were comparable in terms of all th PEMs; while GBM achieved slightly better sensitivity of 92% as compared to the sensitivity of 91.9% by RF. The best PEM was obtained by GBM with 92% sensitivity, 86.6% specificity, 89.3% accuracy, 0.788 mcc and 0.953 AUC on 10-fold cross validation.

Table 3: PEM for combined feature set (i.e. PAAC, PSSM, PSE-PSSM

Model	Encoding	Sensitivity	Specificity	Accuracy	MCC	AUC
RF	PSSM +	0.919	0.839	0.880	0.763	0.943
GBM	PAAC +	0.920	0.866	0.893	0.788	0.953
DLNNWOD	PSEPSSM	0.881	0.833	0.857	0.718	0.923
DLNNWD		0.852	0.840	0.846	0.707	0.865

Table 4: PEM for combined feature set on testing set (i.e. PAAC, PSSM, PSE-PSSM)

Model	Encoding	Sensitivity	Specificity	Accuracy	MCC	ACC
RF	PSSM+PAAC	0.866	0.876	0.871	0.743	0.936
GBM	+PSEPSSM	0.907	0.859	0.881	0.765	0.943
DLNNWOD		0.878	0.825	0.849	0.701	0.913
DLNNWD		0.908	0.851	0.877	0.757	0.897

On testing set, the best sensitivity was achieved by DLNNWD (sensitivity = 90.8%) and GBM (sensitivity = 90.7%). But in terms of specificity, accuracy, mcc and AUC, the performance of GBM and DLNNWOD were comparable, with GBM showing best PEM of 85.9% specificity, 0.765 MCC, 0.943 AUC and 88.1% accuracy.

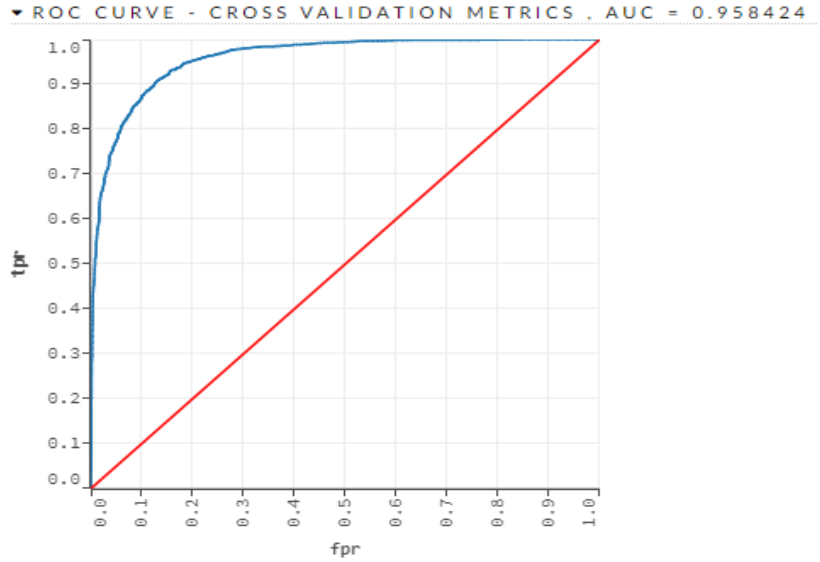

Fig. 1: ROC curve for the best classification algorithm (GBM)

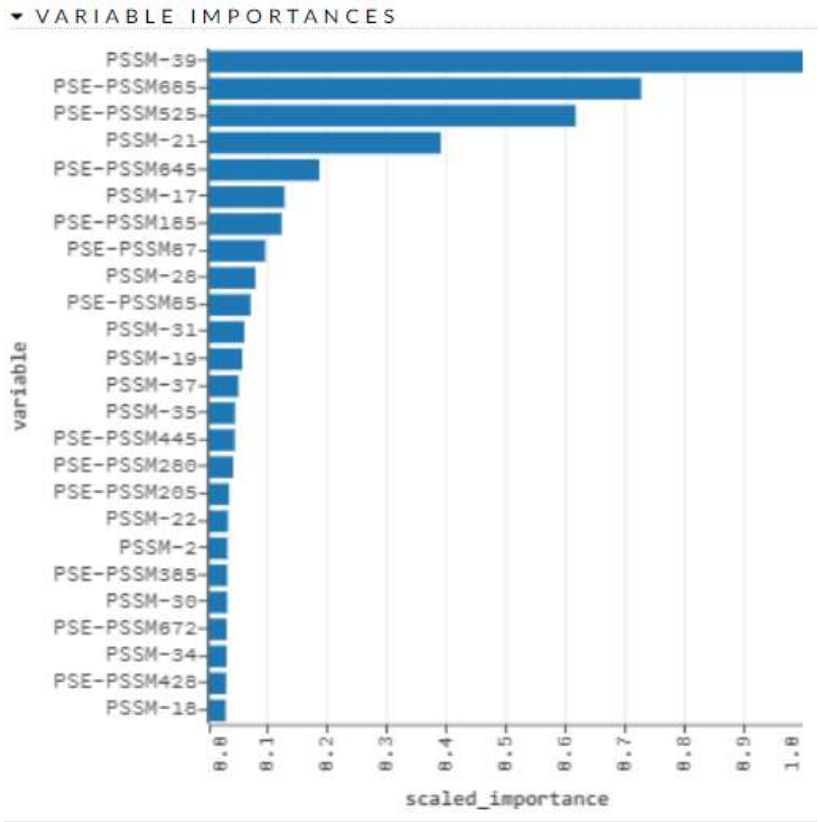

Fig. 2: Variable importance plot for the best performing algorithm (gbm)

Not all the features are equally important for the trained model; some of them have greater influence on the model's prediction. Global feature importance is captured by variable importance plots (VIP). VI plot provides the list of most significant feature, with best features at the top with its score. Another way to capture both local and global feature importance is through Shapely plots. SHAP values show how much each feature in a model has contributed to the prediction. It combines feature importance with feature effects. Evolutionary information based features i.e. PSSM and PSE-PSSM are the two most important features in discriminating the VFs from non-VFs. Among these models, in most cases, the performances of evolutionary information based features (e.g., PSSM and PSE-PSSM) were found to be higher.

The current methodology utilising the representative sampling and use of feature fusion (Evolutionary features+ sequence based features) resulted in the overall accuracy of 89.3%. Further, the implementation of representative training and testing set using uniform sampling resulted in

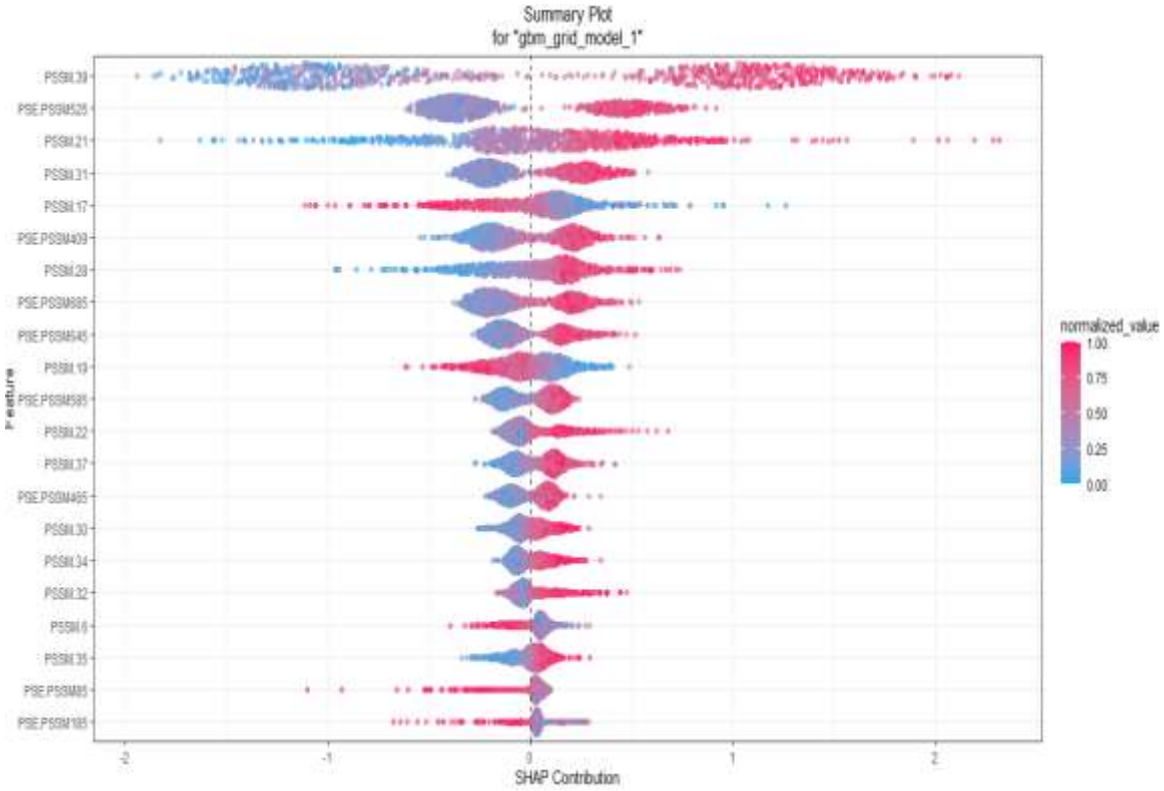

Fig. 3: summary plot for the GBM algorithm

balanced performance evaluation metrics for the learning algorithms. The present work can also complement other laboratory based methods in reducing the search space for putative VIFs.

Conclusion: Virulence factors facilitate the pathogens to infect a host. The interaction between host and pathogen decidedly shapes the outcome of an infection, thus understanding this interaction is critical to understand the mechanism of pathogenesis. To predict virulence factors, it is important to choose a classifier and a set of reasonable information parameters from protein sequences. An accurate VF predictor not only reduces the time invested on finding VF, but also decreases the search space for the experimental wet lab based conformation methods. In this paper, Pseudo-amino acid composition (PAAC), Position specific scoring matrix (PSSM), DPC-PSSM, and PSE-PSSM were selected to predict the VFs. The generation of features was performed by using ifeature (Chen *et al.*, 2018) and PSSMCool (Mohammadi *et al.*, 2022) packages in Python. The high values of different performance evaluation metrics showed that our method is suitable for VF prediction with notable prediction performance. Out of the four classifiers used (GBM, RF, DLNNWOD and DLNNWD), GBM surpassed the performance evaluation of other published results. The results indicate that the present method is fairly good enough for predicting VFs.

REFERENCES

Altschul, S. 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, **25**(17): 3389-3402.

Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T.T., Wang, Y., Webb, G.I., Smith, A.I., Daly, R.J., Chou, K.C. and Song, J. 2018. iFeature: A Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics*, **34**(14): 2499-2502.

Chien, J. 2019. *Deep Neural Network*. Elsevier EBooks, 259-320. [*https://doi.org/10.1016/b978-0-12-804566-4.00019-x*].

Chou, K. 2009. Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology. *Current Proteomics*, **6**(4): 262-274.

Chou, K.C. and Shen, H.B. 2007. MemType-2L: A web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. *Biochemical and Biophysical Research Communications*, **360**(2): 339-345.

Dehzangi, A., Sharma, A., Lyons, J., Paliwal, K.K. and Sattar, A. 2015. A mixture of physicochemical and evolutionary-based feature extraction approaches for protein fold recognition. *International Journal of Data Mining and Bioinformatics*, **11**(1): 115. [*https://doi.org/10.1504/ijdmb.2015.066359*].

Gromiha, M.M. 2010. Protein sequence analysis. *Protein Bioinformatics*, Dec., 2010: 29-62. [*https://doi.org/10.1016/b978-8-1312-2297-3.50002-3*].

Gupta, A., Kapil, R., Dhakan, D.B. and Sharma, V.K. 2014. MP3: A software tool for the prediction of pathogenic proteins in genomic and metagenomic data. *PLoS ONE*, **9**(4): e93907. [*https://doi.org/10.1371/journal.pone.0093907*].

Jones, D.T. and Swindells, M.B. 2002. Getting the most from PSI–BLAST. *Trends in Biochemical Sciences*, **27**(3): 161-164.

Kaur, S. and Forster, J. 2013. Virulence, genetics of. *Brenner's Encyclopedia of Genetics*, **2**: 287-289. [*https://doi.org/10.1016/b978-0-12-374984-0.00636-7*].

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T. 2017. LightGBM: A highly efficient gradient boosting decision tree. *Neural Information Processing Systems*, **30**: 3149-3157.

Kennard, R.W. and Stone, L.A. 1969. Computer aided design of experiments. *Technometrics*, **11**(1): 137-148.

Khanh, Le, N.Q., Nguyen, Q.H., Chen, X., Rahardja, S. and Nguyen, B.P. 2019. Classification of adapt or proteins using recurrent neural networks and PSSM profiles. *BMC Genomics*, **20** (Suppl. 9): 966. [*https://doi.org/10.1186/s12864-019-6335-4*].

Liaw, A. and Wiener, M. 2002. Classification and regression by randomForest. *R news*, **2**(3): 18-22.

Mohammadi, A., Zahiri, J., Mohammadi, S., Khodarahmi, M. and Arab, S.S. 2022. PSSMCOOL: A comprehensive R package for generating evolutionary-based descriptors of protein sequences from PSSM profiles. *Biology Methods and Protocols*, **7**(1): 18-22.

Nath, A. and Subbiah, K. 2015. Maximizing lipocalin prediction through balanced and diversified training set and decision fusion. *Computational Biology and Chemistry*, **59**: 101-110.

Sachdeva, G., Kumar, K., Jain, P. and Ramachandran, S. 2004. SPAAN: A software program for prediction of adhesins and adhesin-like proteins using neural networks. *Bioinformatics*, **21**(4): 483-491.

Sharma, A.K., Dhasmana, N., Dubey, N., Kumar, N., Gangwal, A., Gupta, M. and Singh, Y. 2016. Bacterial virulence factors: Secreted for survival. *Indian Journal of Microbiology*, **57**(1): 1-10.

Tanwer, P., Kolora, S.R.R., Babbar, A., Saluja, D. and Chaudhry, U. 2020. Identification of potential therapeutic targets in *Neisseria gonorrhoe* by an *in-silico* approach. *Journal of Theoretical Biology*, **490**: 110172. [*https://doi.org/10.1016/j.jtbi.2020.110172*].

Xie, R., Li, J., Wang, J., Dai, W., Leier, A., Marquez-Lago, T.T., Akutsu, T., Lithgow, T., Song, J., and Zhang, Y. 2020. Deep VF: A deep learning-based hybrid framework for identifying virulence factors using the stacking strategy. *Briefings in Bioinformatics*, **22**(3): [*https://doi.org/10.1093/bib/bbaa125*].