# The Machine Learning Model for Predicting Phosphorylation Sites in Protein

- 1. Introduction
- 2. Dataset
  - a. pFeature dataset
  - b. Protparam dataset
  - c. Rdkit dataset
- 3. Models
  - a. Machine learning models

i.	Random forest	SVM	KNN				
ii.	Naive Bayes	MLP	Catboost				
iii.	SGD	LR	Voting Classifier				
iv.	Bagging Classifier	Adaboost	LightGBM				
V.	Histogram-based Gradient Boosting Classification Tree						

- b. Deep learning models
  - i. Simple DL model
  - ii. RNN
  - iii. LSTM
  - iv. Bi-directional LSTM
- 4. Optimizers
  - a. Adam
  - b. RMSProp
  - c. Adadelta
- 5. Activation Function
  - a. Sigmoid
  - b. Tanh
  - c. Relu
- 6. Cross Validation
  - a. Test train split
  - b. K-fold
  - c. Stratified k-fold
- 7. Results

## Introduction

Phosphorylation of protein is a post-translational modification (PTM) of protein that is responsible for adding phosphate groups to the residues of amino acids on proteins. It is one of the most essential and studied PTM and takes place due to the addition of phosphate group to the residues of Serine, Threonine, and Tyrosine. The present methods available for the prediction of the presence of phosphorylated sites are limited as well as time-consuming and machine learning-based techniques can play a vital role in overcoming these challenges. In this work, I have applied different machine learning techniques and deep learning models on the various features hypertuned with different optimizers, batch size, iterations, and model specific parameters. For the evaluation of the models I have used accuracy, precision, recall and f1-score and the models having high accuracy and f1-score are considered best. I have procured an maximum accuracy of 84.30% and f1-score of 85% deep learning with sgd as optimizer and for 200 iterations and 83.28 % accuracy and 84.37% f1-score using random forest classifier using 10 fold cross validation.

# **Dataset**

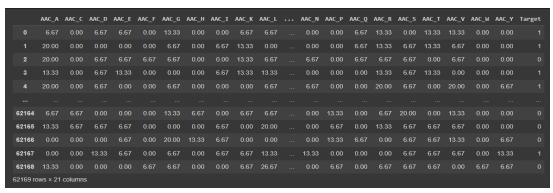
The dataset has been created using the various individual datasets combined together to form a single dataset. The positive examples have been taken from the phosphorylation dataset of asparticacid, histidine, arginine, cysteine, serine, threonine and tyrosine and a dataset containing both positive and negative samples of phosphorylated sequence. This dataset is also used for negatives samples. After combining this dataset random samples are taken thus the final dataset contains 30000 negative samples and around similar number of positive samples. The dataset has been saved in .csv (phospho\_dataset.csv) and .fasta (phospho\_dataset.fasta) format and these files are used further in all the models and to extract other features and create other dataset. This main dataset is used for creating the following datasets:

**pFeature dataset:** This dataset contains 3 different datasets i.e. amino acid composition, dipeptide composition and physio-chemical properties. Apart from this it contains a csv for label. **Protparam dataset:** It contains two different sets of features extracted using protparam library. **Rdkit dataset:** It contains features extracted using the rdkit library.

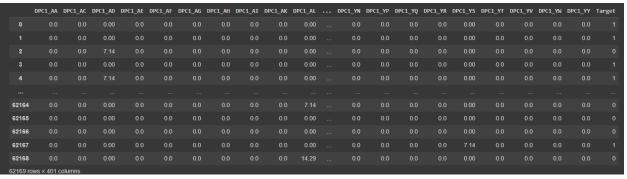
Thus, I have worked with 6 different set of datasets and their combination.



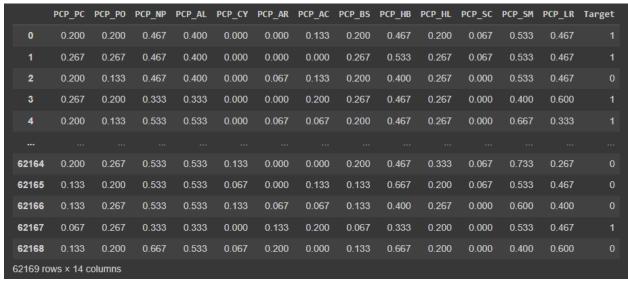
Main dataset(phospho\_dataset.csv)



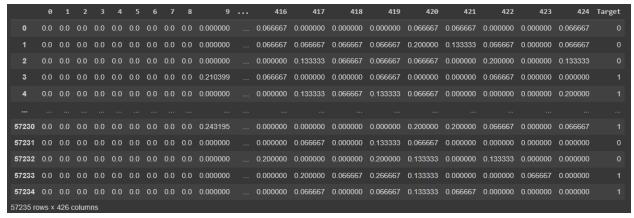
pFeature dataset1 (amino acid composition)



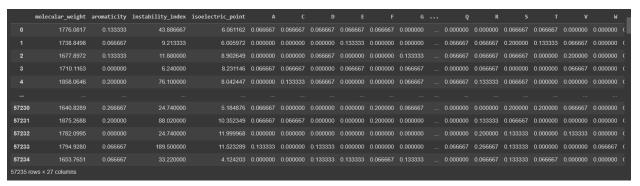
pFeature dataset2 (dipeptide composition)



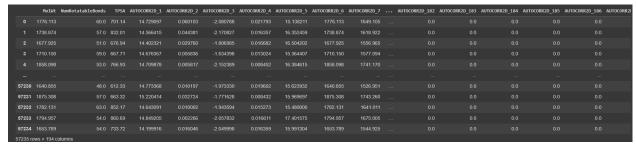
pFeature dataset3 (physio-chemical)



Protparam dataset1



Protparam dataset2



Rdkit dataset

#### **Models**

#### **Machine Learning models**

**Random Forest:** It is an ensemble technique that takes the output of multiple decision trees and votes for the final output. The decision tree technique is applied to different subsamples of the dataset and makes its individual decisions.

**SVM:** Support Vector Machine is a supervised learning method that works on the principle of classification using hyperplanes to separate classes in multi-dimensional space. It generates optimal hyperplane iteratively for minimizing the error. SVM aims to maximize the marginal hyperplane to divide the classes such that the optimal hyperplane has the maximum margin with the support vectors.

**KNN:** K-Nearest Neighbour is a non-parametric supervised learning method that stores the dataset and uses the similarity (distance) with the k nearest neighbors for the classification.

**Naive Bayes:** It is a probabilistic supervised learning algorithm that uses the Bayes theorem for classification. The naive means that this approach assumes that the input features are conditionally independent of each other.

**MLP:** Multi-Layer Perceptron is a type of artificial neural network having multiple hidden layers and connected nodes that are used for the task of classification.

**Catboost:** Categorical Boosting is a type of gradient boosting algorithm that is used for handling both numerical as well as categorical type of data for the task of classification. It is an ensemble technique where every decision tree is trained to minimize the error of the previous tree.

**SGD:** It stands for stochastic gradient descent. It is a classifier based on a linear model with an assumption that classes are linearly separable. It uses the gradient of the loss function to update the weights of the data sample. It is computationally efficient to work with large datasets.

**Logistic Regression:** It is a statistical supervised learning method used for binary classification. It uses the sigmoid function for modeling the probability of input belonging to a specific class. It uses a certain threshold (0.5) to classify the input into positive and negative classes.

**Voting Classifier:** It is an ensemble technique that takes various estimators as input to predict the final output. Various estimators are trained individually and classify the input based on the highest majority of voting.

**BaggingClassifier:** It is an ensemble based classifier that trains same classifiers on the different random subsets of the original dataset and the final output is obtained by taking the vote of each estimator.

**Adaboot:** It is also an meta-estimator that trains the classifier on the dataset and then the copies of the classifier are trained on the same dataset in order to prioritize the weight of the misclassified instances.

**LightGBM:** It is a gradient boosting algorithm that uses decision trees called LightGBM trees. These trees works as an ensemble technique and the aggregation of these collective output predicts the final output.

**Histogram-based Gradient Boosting Classification Tree:** It is a boosting algorithm that ensembles various decision trees known as weak learners to predict the final output. This algorithm uses approach based on histogram to find the split best suited for the feature.

# **Deep Learning models**

Deep learning is a sub domain of machine learning that is based on the artificial neural networks. It architecture consists of a multi-layered structure of nodes interconnected to each other that helps to extract complex pattern from the input.

**RNN:** Recurrent neural networks is a variation of neural networks where each current step takes the output of previous step as an input. It is helpful with the classification of sequences as it has a memory and stores the previous information.

**LSTM:** It stands for Long Short Term Memory and is an extension of RNN. RNNs are capable of storing information of shorter range and fails to store longer dependencies and LSTM resolves this issue using gated units.

**Bidirectional LSTM:** It is a combination of two LSTMs one for the processing in forward direction and the other in backward direction. It helps to understand and establish the relationship between the sequence in a more contextual manner.

# **Optimizers**

Optimizers are the algorithm that helps the deep learning models to minimize the loss function by adjusting and updating the weights of the network.

The following optimizers have been used:

**ADAM:** It stands for Adaptive Moment Estimation and is a combination of RMSProp and momentum techniques. It accelerates convergence by accumulating the gradients and adjusts the learning rate dynamically at the time of training.

**RMSProp:** Root Mean Square Propogation is an adaptive learning algorithm for optimization. It uses exponential moving average for scaling the learning rate of parameters.

**AdaDelta:** It is a stochastic gradient descent method based optimization technique that uses adaptive learning rate optimization. It is the improvement of other adaptive learning techniques like RMSProp.

## **Active Function**

Active functions are used to introduce the non-linearity in the network by deciding if a neuron should be fired or not based on the computation of the weighted sum of the network and the bias. The active functions used are:

**Sigmoid:** It is an S shaped curve that ranges from 0 to 1 and is used for binary classification based on the threshold (0.5) and the equation is given by:

$$A = \frac{1}{1 + e^{-x}}$$

**Tanh:** It is a shifted version of sigmoid function that ranges from -1 to 1. The equation of tanh is given by:

$$A = \frac{2}{1 + e^{-2x}} - 1$$

**Relu:** It stands for rectified linear unit and is given by the equation:

$$A = max(x, 0)$$

#### Cross validation

It is a technique used to evaluate the performance of the machine learning models on the unseen data. It splits the data into train and testing sets which are further used to train and test the data separately. The cross validation techniques used are:

**Test-Train Split:** In this method the dataset is simply divided into train and test data in a fixed ratio. The model is trained on this training data and testing on the testing data.

**K-fold:** In this method the dataset is divided into k subsets out of which k-1 subsets are used for training and 1 subset for testing and this is done for all the subsets.

**Stratified k-fold:** In this method the dataset is divided into k-folds while maintaining the proportion of the positive and negative samples.

# Result

RF = Random Forest, NB = Naive Bayes , MLP = Multi layer perceptron, SGD = Stochastic Gradient Descent, LR = Logistic Regression, VC = Voting Classifier

Model	CV	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
RF	10 fold	83.28	82.07	86.81	84.37
RF	Stratified 10 fold	83.12	81.75	86.66	84.11
NB	10 fold	67.33	67.79	70.62	69.17
NB	Stratified 10 fold	67.48	67.97	7.67	69.29
MLP	10 fold	76.68	77.57	82.82	80.10
SGD	10 fold	63.89	65.68	64.68	65.12
LR	10 fold	67.89	67.85	72.75	70.21
LR	Stratified 10 fold	67.75	67.77	72.47	70.04
DL model	-	80.97	-	-	-

Machine learning and deep learning results on dipeptide composition dataset.

Epochs	Optimizer	Batch size	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
100	adam	64	77.03	75.97	81.68	78.72
100	adam	128	77.10	77.43	79.00	78.21
100	adam	256	76.99	78.00	77.67	77.83
200	adam	64	74.97	74.23	74.46	76.67
200	adam	128	75.84	75.91	78.45	77.16
200	adam	256	76.90	77.51	78.31	77.90
200	adam	512	76.98	77.48	78.58	78.03
100	sgd	64	83.43	83.39	85.09	84.23
100	sgd	128	83.45	83.13	85.55	84.32
100	sgd	256	83.46	83.18	85.49	84.32
200	sgd	64	83.77	84.95	83.59	84.27
200	sgd	128	84.40	84.58	85.62	85.09
200	sgd	256	84.28	84.59	85.31	84.95
200	sgd	512	84.30	84.48	85.51	85.00

Deep learning model on amino acid composition.

Epochs	Optimizer	Batch size	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
100	adam	64	66.43	64.46	79.05	71.01
100	adam	128	67.11	65.48	77.75	71.09
100	adam	256	67.35	65.95	76.97	71.03
200	adam	64	67.44	65.69	78.30	71.44
200	adam	128	68.12	67.10	75.95	71.25
200	adam	256	68.88	68.10	75.33	71.53

200	adam	512	67.91	67.85	72.79	70.23
100	sgd	64	66.06	66.97	68.58	67.76
100	sgd	128	66.72	67.05	70.81	68.88
100	sgd	256	66.45	63.51	83.42	72.12
200	sgd	64	66.90	65.85	75.54	70.36
200	sgd	128	66.07	66.97	68.58	67.77
200	sgd	256	66.28	66.68	70.27	68.43
200	sgd	512	66.30	64.85	76.87	70.35

Deep learning model on physio-chemical properties.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
NB	65.76	62.5	60.44	61.45
RF	74.80	74.37	71.13	72.71
LR	67.77	69.32	65.63	67.43
MLP	68.77	68.59	64.01	66.22
SVM	55.80	71.80	19.12	30.20
SGD	58.99	53.86	94.29	68.56
VC	73.62	74.17	73.79	73.98
DL	64.58	61.06	72.33	66.22

Machine learning and deep learning model on protparam dataset1.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
RF	75.60	74.09	75.59	74.84
LR	63.72	62.49	31.12	61.80
MLP	67.30	66.13	65.35	65.79
SGD	62.04	59.84	63.58	61.66

Machine learning models on protparam dataset2.

Epochs	Optimizer	Batch size	Accuracy	Precision	Recall	F1-Score
			(%)	(%)	(%)	(%)
100	adam	64	70.13	68.16	70.88	69.49
100	adam	128	70.21	68.88	69.24	69.06
100	adam	256	70.27	68.64	70.08	69.35
100	adam	512	71.07	70.30	68.80	69.54
200	adam	64	69.89	69.20	67.18	68.18
200	adam	128	69.92	67.98	70.57	69.25
200	adam	256	70.30	68.63	70.24	69.43
200	adam	512	70.35	68.18	71.68	69.88

100	sgd	64	69.33	67.92	68.44	68.18
100	sgd	128	68.59	66.10	70.95	68.44
100	sgd	256	68.07	68.44	62.20	65.16
100	sgd	512	67.03	68.49	57.98	62.81
200	sgd	64	69.63	65.87	76.25	70.68
200	sgd	128	68.64	64.37	77.65	70.39
200	sgd	256	68.27	66.31	68.89	67.58
200	sgd	512	66.93	91.90	80.92	70.14

Deep learning model on protparam dataset2.

Algorithm	CV Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
KNN	10 fold	71.78	70.54	78.97	74.51
KNN	Stratified 10 fold	71.44	70.20	78.79	74.25
Cataboost	10 fold	68.04	67.70	73.94	70.68
Cataboost	Stratified 10 fold	67.98	67.64	73.93	70.64
Hist Gradient Boosting	10 fold	68.95	68.47	75.05	71.60
Hist Gradient Boosting	Stratified 10 fold	68.97	68.41	74.76	71.44

KNN	Test-train split	70.85	69.13	77.83	73.24
Cataboost	Test-train split	68.44	67.33	74.52	70.74
Bagging Classifier	Test-train split	77.01	74.89	82.31	78.69
Hist Gradient Boosting	Test-train split	69.20	68.00	75.29	71.46
Adaboost	Test-train split	78.11	76.19	83.28	79.58
LGBM	Test-train split	69.10	67.86	75.32	71.40

Machine learning models on combined features of amino acid composition, dipeptide composition, physio-chemical properties.

Batch size	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
64	70.91	69.36	70.41	69.87
128	72.23	70.43	72.44	71.42
256	72.87	71.69	71.65	71.67
512	73.04	71.32	73.12	72.21

LSTM model on Protparam dataset1.

Batch size	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
64	68.04	65.62	69.89	67.69

128	68.28	66.94	66.75	66.84
256	68.51	67.15	67.05	67.10
512	68.42	68.06	64.21	66.08

LSTM model on Protparam dataset2.

Batch size	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
64	70.50	69.05	78.42	73.44
128	70.18	69.23	76.80	72.82
256	70.45	69.11	78.09	73.32
512	70.49	69.50	77.09	73.10

LSTM model on Pfeatures dataset 1 (amino-acid composition).

Batch size	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
64	73.99	72.82	79.74	76.13
128	74.51	74.01	78.58	76.23
256	75.03	74.26	79.56	76.82
512	75.35	76.10	76.69	76.39

LSTM model on Pfeatures dataset 2 (dipeptide composition).

Batch size	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
64	66.53	64.54	73.12	71.09
128	66.69	64.93	72.19	70.95
256	66.84	65.22	77.68	70.91
512	66.72	65.08	77.91	70.89

LSTM model on Pfeatures dataset 3 (physio-chemical structure).

Batch size	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
64	71.82	68.98	75.02	71.87
128	73.04	71.44	72.83	72.13
256	73.09	71.03	73.33	72.48
512	73.65	71.57	74.63	73.07

Bi-LSTM model on Protparam dataset1.

Batch size	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
64	68.35	66.10	69.63	67.82
128	68.64	66.37	69.97	68.13
256	68.53	65.56	72.28	68.76

512	68.45	65.87	70.85	68.27

Bi-LSTM model on Protparam dataset2.

Batch size	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
64	74.24	73.90	78.02	75.91
128	75.71	75.73	78.44	77.06
256	75.92	75.32	79.87	77.53
512	76.46	76.22	79.57	77.86

Bi-LSTM model on Pfeatures dataset 1 (amino-acid composition).

Batch size	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
64	74.24	79.90	78.02	75.91
128	75.71	75.73	78.44	77.06
256	75.92	75.32	79.87	77.53
512	76.46	76.22	79.57	77.86

Bi-LSTM model on Pfeatures dataset 2 (dipeptide composition).

Batch size	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
64	66.69	65.03	77.77	70.83
128	67.39	65.63	78.30	71.41

256	67.40	66.23	76.14	70.84
512	67.59	66.26	76.80	71.14

Bi-LSTM model on Pfeatures dataset 3 (physio-chemical).

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
RF	73.60	72.97	72.07	72.52
MLP	67.34	64.60	71.14	67.98
SGD	64.70	63.91	61.93	62.90
SVC	66.60	65.02	66.84	65.92
LR	64.58	63.88	61.44	62.64
KNN	67.9	66.09	69.01	67.52
Cataboost	72.82	71.78	72.11	71.94
Hist Gradient Boosting	68.25	66.50	69.16	67.80
Bagging Classifier	73.33	71.14	74.05	72.88
Adaboost	74.10	72.94	73.77	73.35
LGBM	68.85	67.14	69.61	68.35

Machine learning models on rdkit dataset.

Approach	Optimizer	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
DL	adam	67.12	65.26	65.34	65.30

	adadelta	67.00	64.64	66.84	65.72
	rmsprop	67.07	64.46	67.82	66.10
LSTM	adam	67.40	64.66	68.66	66.06
	adadelta	67.50	64.73	68.88	66.74
	rmsprop	67.58	64.93	68.52	66.68
Bi-LSTM	adam	66.97	64.40	67.58	65.95
	adadelta	66.90	64.22	67.91	66.01
	rmsprop	67.17	64.57	67.91	66.20

Deep learning models on rdkit dataset.

Optimizer	Dataset	Batch size	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
Adadelta	Protparam Dataset 1	512	63.93	63.66	57.53	60.44
	Protparam Dataset 2	512	62.32	60.99	59.18	60.07
	Pfeature Dataset 1	512	66.40	66.28	72.09	69.06
	Pfeature Dataset 2	512	66.47	67.11	69.69	68.37
	Pfeature Dataset 3	512	64.22	63.25	74.47	68.40
	Rdkit Dataset	64	52.10	0.0	0.0	0.0
RMSProp	Protparam	128	67.69	65.78	67.82	66.79

Dataset 1					
Protparam Dataset 2	512	70.64	69.94	67.86	68.88
Pfeature Dataset 1	256	69.63	68.74	76.31	72.33
Pfeature Dataset 2	512	72.91	73.56	74.79	74.17
Pfeature Dataset 3	512	66.58	65.11	77.00	70.56
Rdkit Dataset	64	52.00	0.0	0.0	0.0

RNN model on all the datasets with adadelta and rmsprop as optimizers and best batch size.

Submitted By: Chitransh Bose

Submitted To: Dr. N. Arul Murugan