

# Accurately Predicting Microbial Phosphorylation Sites

## Using Evolutionary and Structural Features



Faisal Ahmed

Department of Computer Science and Engineering

United International University

A thesis submitted for the degree of

*Master of Science in Computer Science & Engineering*

October 2022

## Declaration

I, Faisal Ahmed, declare that this thesis titled Accurately Predicting Microbial Phosphorylation Sites Using Evolutionary and Structural Features and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a MSc/ BSc degree at United International University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at United International University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

---

Date:

---

Faisal Ahmed

## Certificate

I do hereby declare that the research works embodied in this thesis entitled Accurately Predicting Microbial Phosphorylation Sites Using Evolutionary and Structural Features is the outcome of an original work carried out by Faisal Ahmed under my supervision.

I further certify that the dissertation meets the requirements and the standard for the degree of MSc in Computer Science and Engineering.

Signed:

---

Date:

---

Dr. Swakkhar Shatabda

Professor, Department of Computer Science and Engineering

Director, IQAC

United International University

Dhaka-1209, Bangladesh.

## Abstract

Post-translational modification (PTM) is a biological process involving a protein's enzymatic changes after its translation by the ribosome. Phosphorylation is one of the most critical PTMs that occurs when a phosphate group interacts with an amino acid residue along the protein sequence. It contributes to cell communication, DNA repair, and gene regulation. Predicting microbial phosphorylation sites can help better understand host-pathogen interaction and develop anti-microbial agents. Experimental methods such as radioactive chemical methods, chromatin immunoprecipitation, and mass spectrometry are time-consuming, laborious, and expensive. Therefore, finding fast and effective computational approaches to detect phosphorylation sites accurately is drawing immense attention.

In this thesis, we propose two new computational tools, RotPhoPred and DeepPhoPred, for predicting phospho-serine (pS), phospho-threonine (pT), and phospho-tyrosine (pY) sites in the microbial organism. In RotPhoPred, we integrate the evolutionary bigram profile with structural information and use Rotation Forest as the classification technique. To the best of our knowledge, our extracted features and employed classifier have never been utilized for this task. Comparative results demonstrate that the RotPhoPred surpasses its peers in terms of different metrics such as sensitivity (90.0%, 75.4% and 78.2%), specificity (92.1%, 97.2% and 94.7%), accuracy (91.0%, 86.3%, 86.4%), and MCC (0.82, 0.74 and 0.74) for pS, pT, and pY sites predictions, respectively. RotPhoPred as a standalone predictor and all its source codes are publicly available at : <https://github.com/faisalahr3d/RotPredPho>

In DeepPhosPred, we design a lightweight two-headed CNN architecture blending squeeze and excitation blocks that can automatically discover compelling features from the peptide's structural and evolutionary properties

without human intervention to detect phosphorylation sites. To the best of our knowledge, it is the first tool that integrates two information sources and applies deep learning technique to tackle this problem. DeepPhosPred outperforms RotPhoPred and current predictors with significantly higher sensitivity (100%, 99.2% and 99.3%), specificity (98.9%, 99% and 98.7%), accuracy (99.0%, 99.0%, 98.7%), and MCC (0.93, 0.94 and 0.94) for pS, pT, and pY sites predictions, respectively.

My genuine gratitude and warmest regard are that

I dedicate this work to my beloved

**Mother**

## Published Papers

Work relating to the research presented in this thesis has been published/ submitted by the author in the following peer-reviewed journals and conferences:

1. Faisal Ahmed, Iman Dehzangi, Md. Mehedi Hasan, Swakkhar Shatabda, “Accurately predicting microbial phosphorylation sites using evolutionary and structural features”, Gene, Elsevier.
2. Faisal Ahmed, Iman Dehzangi, Md. Mehedi Hasan, Swakkhar Shatabda, “DeepPhoPred: An accurate two-headed deep convolutional neural network to predict microbial phosphorylation sites using evolutionary and structural information”, Computational and Structural Biotechnology Journal, Elsevier. [**Submitted**]

## Acknowledgements

First, I would like to express sincere gratitude and appreciation to my academic supervisor Dr. Swakkhar Shatabda, Professor, Department of Computer Science & Engineering and Director, IQAC, United International University, Dhaka, Bangladesh, for his guidance, motivation, and constant support throughout my MSc journey. His opinions and wisdom have shaped how I think about research and will surely accompany me in the future. Without his instructive supervision, this thesis would not have seen the light of day.

I am highly grateful to Dr. Iman Dehzangi, Assistant Professor, Department of Computer Science, Center for Computational and Integrative Biology (CCIB), Rutgers University, Camden, NJ 08102, USA, for being part of my research as an unofficial supervisor. He has accelerated my research by giving scholarly criticisms and corrections during the weekly meetings of the entire thesis period.

I thank the honorable members of the examination board for their invaluable suggestions. I also acknowledge the support of all faculty members and staff of the department of CSE, UIU.

My family, including my parents, siblings, and wife, deserve special recognition and appreciation for their unconditional love and immense emotional support.

Finally, I praise the Almighty Allah for giving me the strength and courage to finish the task and achieve the goal.

# Contents

|                                                  |            |
|--------------------------------------------------|------------|
| <b>List of Figures</b>                           | <b>xii</b> |
| <b>List of Tables</b>                            | <b>xiv</b> |
| <b>1 Introduction</b>                            | <b>1</b>   |
| 1.1 Motivation . . . . .                         | 1          |
| 1.2 Objectives of the Thesis . . . . .           | 2          |
| 1.3 Thesis Contributions . . . . .               | 3          |
| 1.4 Organization of the Thesis . . . . .         | 4          |
| <b>2 Background &amp; Related Work</b>           | <b>5</b>   |
| 2.1 Post Translational Modification . . . . .    | 5          |
| 2.2 Machine Learning . . . . .                   | 7          |
| 2.3 Deep Learning . . . . .                      | 9          |
| 2.4 Convolutional Neural Network (CNN) . . . . . | 10         |
| 2.4.1 Convolution Layer . . . . .                | 10         |
| 2.4.2 Pooling Layer . . . . .                    | 11         |
| 2.4.3 Fully Connected Layer . . . . .            | 11         |
| 2.5 Related works . . . . .                      | 12         |
| 2.6 Research Gaps . . . . .                      | 14         |
| 2.7 Summary . . . . .                            | 15         |
| <b>3 RotPhoPred: An ensemble based method</b>    | <b>16</b>  |
| 3.1 Material and Methods . . . . .               | 16         |
| 3.1.1 Benchmark datasets . . . . .               | 16         |
| 3.1.2 Features . . . . .                         | 18         |

|          |                                                                    |           |
|----------|--------------------------------------------------------------------|-----------|
| 3.1.3    | Formulation of peptide fragments for each site . . . . .           | 18        |
| 3.1.4    | Evolutionary feature extraction . . . . .                          | 19        |
| 3.1.5    | Structural feature extraction . . . . .                            | 20        |
| 3.1.6    | Formation of feature vector . . . . .                              | 21        |
| 3.1.7    | Balancing dataset . . . . .                                        | 22        |
| 3.1.8    | Classification model . . . . .                                     | 22        |
| 3.1.9    | Validation scheme . . . . .                                        | 23        |
| 3.1.10   | Evaluation metrics . . . . .                                       | 23        |
| 3.2      | Results and Discussion . . . . .                                   | 25        |
| 3.2.1    | Feature significance analysis . . . . .                            | 26        |
| 3.2.2    | Comparison with different classifiers . . . . .                    | 28        |
| 3.2.3    | Comparison with current state-of-the-art predictors . . . . .      | 30        |
| 3.3      | Summary . . . . .                                                  | 34        |
| <b>4</b> | <b>DeepPhoPred: A two-headed CNN model</b>                         | <b>35</b> |
| 4.1      | Methodology . . . . .                                              | 35        |
| 4.1.1    | Encoding protein segment . . . . .                                 | 35        |
| 4.1.2    | Balancing the dataset . . . . .                                    | 36        |
| 4.1.3    | Architecture of the proposed deep learning model . . . . .         | 37        |
| 4.1.4    | Model training . . . . .                                           | 40        |
| 4.2      | Results and Discussion . . . . .                                   | 41        |
| 4.2.1    | Contribution analysis of Squeeze and Excitation (SE) block . . .   | 41        |
| 4.2.2    | Contribution analysis for different sequence representations . . . | 41        |
| 4.2.3    | Comparison with traditional machine learning-based classifiers .   | 43        |
| 4.2.4    | Comparing data balancing techniques . . . . .                      | 44        |
| 4.2.5    | Selection of optimal window size . . . . .                         | 46        |
| 4.2.6    | Visualization of the features . . . . .                            | 48        |
| 4.2.7    | Comparison with existing predictors . . . . .                      | 48        |
| 4.3      | Summary . . . . .                                                  | 51        |
| <b>5</b> | <b>Summary, Conclusions, and Future Work</b>                       | <b>53</b> |
| 5.1      | Summary . . . . .                                                  | 53        |
| 5.2      | Conclusions . . . . .                                              | 54        |
| 5.3      | Future Work . . . . .                                              | 55        |

|                                                                            |           |
|----------------------------------------------------------------------------|-----------|
| <b>Bibliography</b>                                                        | <b>57</b> |
| <br>                                                                       |           |
| <b>A Supplementary Data for RotPhoPred</b>                                 | <b>66</b> |
| A.1 Comparing different window sizes . . . . .                             | 66        |
| A.2 Comparing data balancing techniques . . . . .                          | 67        |
| <br>                                                                       |           |
| <b>B Supplementary Data for DeepPhoPred</b>                                | <b>68</b> |
| B.1 Cross-validation results of DeepPhoPred with and without SE blocks . . | 68        |
| B.2 Comparing different encoding methods on independent test . . . . .     | 68        |
| B.3 Loss and Accuracy Curves . . . . .                                     | 68        |

# List of Figures

|     |                                                                                                                                                                                                                                                                 |    |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | Phosphorylation on Serine, Threonine and Tyrosine residues . . . . .                                                                                                                                                                                            | 7  |
| 2.2 | Supervised Machine Learning . . . . .                                                                                                                                                                                                                           | 8  |
| 2.3 | Unsupervised Machine Learning . . . . .                                                                                                                                                                                                                         | 8  |
| 2.4 | Reinforcement Learning . . . . .                                                                                                                                                                                                                                | 9  |
| 2.5 | Semi-supervised Learning . . . . .                                                                                                                                                                                                                              | 9  |
| 2.6 | Architecture of a Convolutional Neural Network. Figure adapted from<br>Analytical Vidhya blog . . . . .                                                                                                                                                         | 10 |
| 2.7 | Convolution operation using 3 x 3 filter . . . . .                                                                                                                                                                                                              | 11 |
| 2.8 | Pooling operation (a) max pooling, (b) average pooling. . . . .                                                                                                                                                                                                 | 12 |
| 3.1 | The overall flowchart of the RotPhoPred. . . . .                                                                                                                                                                                                                | 17 |
| 3.2 | The representation of serine residue with its upstream and downstream<br>amino acids. (a) Serine residue with enough neighbors on the upstream<br>and downstream sides. (b) Serine residue with inadequate neighbors,<br>either upstream or downstream. . . . . | 19 |
| 3.3 | Schematic overview of five folds cross-validation scheme. . . . .                                                                                                                                                                                               | 24 |
| 3.4 | ROC curves for different feature groups for (a) pS, (b) pT, and (c) pT<br>sites identifications on independent test . . . . .                                                                                                                                   | 27 |
| 3.5 | ROC curves for different classifiers for (a) pS, (b) pT, and (c) pY sites<br>identifications on independent test . . . . .                                                                                                                                      | 31 |
| 4.1 | Working process of SMOTE . . . . .                                                                                                                                                                                                                              | 37 |
| 4.2 | Architecture of the proposed deep learning model DeepPhoPred . . . . .                                                                                                                                                                                          | 38 |
| 4.3 | Overall operation of squeeze and excitation block . . . . .                                                                                                                                                                                                     | 39 |

## LIST OF FIGURES

---

|     |                                                                                                                                                                                                                                           |    |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 4.4 | Comparative results for different encoding techniques. (a)ROC curves for pS site, (b)PR curves for pS site, (c)ROC curves for pT site (d) PR curves for pT site, (e) ROC curves for pY site and (f) PR curves for pY site . . . . .       | 43 |
| 4.5 | Comparative results for different classifiers. (a) ROC curves for pS site, (b)PR curves for pS site, (c)ROC curves for pT site (d) PR curves for pT site, (e) ROC curves for pY site and (f) PR curves for pY site . . . .                | 45 |
| 4.6 | Comparative results for different data balancing technique. (a) ROC curves for pS site, (b)PR curves for pS site, (c)ROC curves for pT site (d) PR curves for pT site, (e) ROC curves for pY site and (f) PR curves for pY site . . . . . | 47 |
| 4.7 | MCC scores for different window sizes for predicting phosphorylation on (a) Serine, (b) Threonine and (c) Tyrosine residues. . . . .                                                                                                      | 49 |
| 4.8 | Two-dimensional projection of row input and features learned by DeepPhoPred (left to right) using t-SNE for pS, pT and pT tasks (top to bottom); blue and orange points are negative and positive samples respectively. . . . .           | 50 |
| B.1 | Loss and accuracy curves (left to right) for training DeepPhoPred to predict pS, pT, and pY sites (top to bottom) . . . . .                                                                                                               | 69 |

# List of Tables

|     |                                                                                                                                                |    |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.1 | The summary of the datasets for pS, pT, and pY identification problems.                                                                        | 18 |
| 3.2 | Impact of different features on the prediction performance using 5-folds cross validation . . . . .                                            | 26 |
| 3.3 | Impact of structural bigram profile in the prediction performance on the independent test set. Bold items indicate the highest values. . . . . | 28 |
| 3.4 | Hyper-parameters summary of different classifiers . . . . .                                                                                    | 29 |
| 3.5 | Comparative results using different machine learning algorithm on 5 folds cross-validation . . . . .                                           | 32 |
| 3.6 | Comparative results using different machine learning algorithm on independent test . . . . .                                                   | 33 |
| 3.7 | Comparison of our method and current predictors for pS site identification on the independent test set. . . . .                                | 34 |
| 3.8 | Comparison of our method and current predictors for pT site identification on the independent test set . . . . .                               | 34 |
| 4.1 | Summary of the hyperparameters' space and their optimal values . . . .                                                                         | 40 |
| 4.2 | Performance of DeepPhoPred with and without SE block on independent test . . . . .                                                             | 41 |
| 4.3 | Comparative results among different sequence representations on 5-folds cross-validation . . . . .                                             | 42 |
| 4.4 | The result achieved for CNN compared to traditional ML models on independent test data (using direct pssm and spd) . . . . .                   | 46 |
| 4.5 | Comparative results for data balancing methods on the independent test set . . . . .                                                           | 46 |

## LIST OF TABLES

---

|     |                                                                                                                                                                  |    |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 4.6 | Comparison of our methods and current predictors for pS site identification on the independent test set. . . . .                                                 | 51 |
| 4.7 | Comparison of our methods and current predictors for pT site identification on the independent test set . . . . .                                                | 51 |
| 4.8 | Comparison of our predictors for pY site identification on the independent test set. . . . .                                                                     | 52 |
| A.1 | Independent test results of RotPhosPred using different window sizes for feature extraction. . . . .                                                             | 66 |
| A.2 | 5-fold cross-validation results of RotPhoPred for predicting pS, pT, and pY when NearMiss and SMOTE methods are applied to balance the training dataset. . . . . | 67 |
| B.1 | Cross-validation results of DeepPhoPred with and without SE blocks . .                                                                                           | 68 |
| B.2 | Independent test results of different encoding methods. . . . .                                                                                                  | 69 |

# Chapter 1

## Introduction

Post-translational modification is a biological mechanism in which one or more amino acids of a protein interact with a specific molecular group after its translation process by the ribosome [1]. Phosphorylation is one of the most critical and common PTMs. It occurs when a phosphate group is added to an amino acid residue. It most commonly appears in serine (S), threonine (T), and tyrosine (Y). It also happens in arginine, lysine, and histidine residues to a lesser extent [2]. Phosphorylation plays an essential role in a wide range of cellular functions, including cell communication, DNA repair, and gene regulation in eukaryote and microbial organisms [3], [4]. Phosphorylation causes dysregulation of cell signalling mechanisms, which results in the development and progress of complex diseases like cancer [5]. For example, p53 is a protein where multiple phosphorylation sites are observed to be responsible for tumor development [6]. Identification of Phosphorylation in prokaryotes cells can provide crucial information for a better understanding of host-pathogen interactions and the development of antimicrobial agents [7].

### 1.1 Motivation

Liquid chromatography-tandem mass spectrometry (LC-MS/MS), radioactive chemical labelling, and western blotting are the most common experimental methods for identifying PTMs, including phosphorylation. However, experimental approaches for detecting PTMs are time-consuming, tedious, expensive, and require a skilled workforce. Moreover, the number of protein sequences is increasing exponentially due to

advanced sequencing technologies. Therefore, it is unfeasible to identify phosphorylation sites using experimental methods in the wet lab from such a massive protein database. Hence, there is a crucial demand for developing fast and accurate computational tools to identify phosphorylation sites.

Machine Learning and Deep learning have become one of the most popular methods for rapidly evaluating massive, complicated scientific datasets to retrieve valuable insights because of the recent tremendous rise in computer capacity and algorithmic breakthroughs. They have also exhibited promising results in bioinformatics and computational biology, including sequence analysis, structure prediction, interaction and reconstruction, biomolecular property and function prediction, biomedical image processing, diagnosis, and systems biology. Consequently, molecular biology has evolved into a data-intensive and -centric field in the twenty-first century.

In recent years, scientists have successfully utilized machine learning and deep learning models to predict post-translational modifications such as acetylation [8][9], methylation [10][11], AMPylation [12], glycation [13][14], malonylation [15][16], sumoylation [17], etc, and achieved promising results.

In this thesis, we applied machine learning and deep learning to identify phosphorylation sites, a vital post-translation modification, in microorganisms.

## 1.2 Objectives of the Thesis

The thesis aims to develop novel computational methods for predicting microbial phosphorylation sites based on machine learning/deep learning, which are relatively more accurate and faster. The followings are the objectives to achieve the aim.

- To find the optimal number of upstream and downstream amino acid residues to represent each phosphorylated and non-phosphorylated site for feature extraction.
- To investigate the impact of different features extracted from protein's structural and evolutionary information on prediction performance.
- To develop and improve the current machine learning methods for microbial phosphorylation site prediction, leveraging structural and evolutionary features.

- To design a deep learning architecture based on CNN that can simultaneously learn significant features from protein’s structural and evolutionary information for accurately predicting microbial phosphorylation sites.

### 1.3 Thesis Contributions

The significant contributions of this paper are as follows:

1. The thesis first proposes the fusion of the evolutionary bigram profile with structural information as features and the utilization of Rotation Forest as the classifier for the first time for microbial phosphorylation prediction.
2. The thesis also proposes a highly optimized two-headed CNN architecture incorporating squeeze and excitation blocks that can jointly learn critical features from the protein’s structural and evolutionary information to enhance prediction accuracy.
3. It is the first study for predicting microbial phospho-tyrosine (pY) sites. Previous studies focused only on phosphorylation prediction on serine (S) and threonine (T) residues.
4. Our proposed predictors are generic, which can predict phospho-serine (pS), phospho-threonine (pT), and phospho-tyrosine (pY) sites applying the same features and classifiers.
5. We have conducted extensive experiments on the benchmark datasets of laboratory-verified phosphorylated sites to validate the effectiveness and applicability of the proposed predictors.
6. The predictors can maintain an excellent balance between sensitivity and specificity in a highly imbalanced dataset, as apparent in the experimental results.
7. We have publicly shared our datasets and models so that researchers can quickly reproduce the results for further experiments and biologists can easily access the predictor for the initial screening of phosphorylation sites.

## 1.4 Organization of the Thesis

The rest of the thesis is organised as follows:

**Chapter 2** first discusses some preliminary concepts related to this thesis and then comprehensively reviews current works on microbial phosphorylation site prediction, highlights their limitations, and identifies the research gaps.

**Chapter 3** describes our proposed microbial phosphorylation site prediction method based on traditional machine learning leveraging rotation forest as a classifier with structural and evolutionary features. It then presents the experimental results for the proposed predictor and discusses them.

**Chapter 4** introduces a deep learning framework founded on CNN for predicting microbial phosphorylation sites and compares it with the machine learning-based method and state-of-the-art predictors.

**Chapter 5** concludes the thesis by summarising the main findings and discussing future research directions.

## Chapter 2

# Background & Related Work

This chapter first explains some preliminary concepts to comprehend the overall thesis, including protein's post-translational modification, machine learning, and deep learning. Finally, the state-of-the-art methods for phosphorylation site prediction and their limitations are highlighted, along with the stress on the research gaps we have addressed.

### 2.1 Post Translational Modification

Protein posttranslational modification (PTM) is a critical regulatory mechanism that refers to proteins' enzymatic changes after their biosynthesis [18]. It is estimated that 5% of the proteome incorporates enzymes that experience more than 200 posttranslational modifications [19]. PTMs can occur in one or more types of amino acids at any phase of the lifespan and alter the chemical characteristics of the modified sites. PTMs appear in diverse cellular organelles, including the nucleus, cytoplasm, endoplasmic reticulum, and Golgi body. They typically occur in proteins with critical structures or activities, such as histones, membrane, and secretory proteins[20].

PTM can be of two types - reversible or irreversible. While the former contains covalent changes, the latter includes proteolytic changes and proceeds in one direction. Research has shown that most cell proteins are controlled by reversible posttranslational modification [21].

PTMs affect numerous protein behaviors and features, including protein solubility, folding, localization, protein longevity, protein-protein interactions, cell-cell and

cell-matrix interactions, molecular trafficking, and receptor activation [20]. Therefore, PTMs contribute to biological processes, including signal transmission, gene expression regulation, gene activation, DNA repair, and cell cycle control. [19].

PTMs also contribute significantly to normal and pathological cell physiology. They enable cells to respond immediately to external environment changes and signal stimulation [22]. Therefore, studying and understanding proteins and their PTMs are fundamental to comprehending their functioning, especially for investigating heart disease, cancer, neurodegenerative diseases, and diabetes [23].

High throughput experimental PTM detection methods in the wet lab include the radioactive chemical method, chromatin immunoprecipitation, and mass spectrometry [24], which require huge time, cost, labor, and human resources [15]. Hence it is necessary to find alternative computational methods to detect PTM sites that are fast, easy, and economical.

Researchers have identified many PTMs, including phosphorylation, prenylation, ubiquitination, methylation, crotonylation, and acetylation [20]. Among these PTMs, Phosphorylation is one of the most critical and common PTMs. It occurs when a phosphate group is added to an amino acid residue. It most commonly appears in serine (S), threonine(T), and tyrosine(Y) (Figure 2.1). It also happens in arginine, lysine, and histidine residues to a lesser extent [2]. The phosphorylation of the side chains of the amino acids tyrosine, serine, and threonine in proteins is the most significant chemical alteration among the intracellular signal transmission pathways. Due to the phosphate group's size and negative charge, phosphorylation is one of the most efficient ways to alter proteins' structure. For the same reason, it works well as a recognition signal for other proteins. Phosphorylation plays an essential role in a wide range of cellular functions, including cell communication, DNA repair, and gene regulation in eukaryote and microbial organisms [3], [4].

Phosphorylation causes dysregulation of cell signalling mechanisms, which results in the development and progress of complex diseases like cancer [5]. For example, p53 is a protein where multiple phosphorylation sites are observed to be responsible for tumor development [6]. Identification of Phosphorylation in prokaryotes cells can provide crucial information for a better understanding of host-pathogen interactions and the development of antimicrobial agents [7].

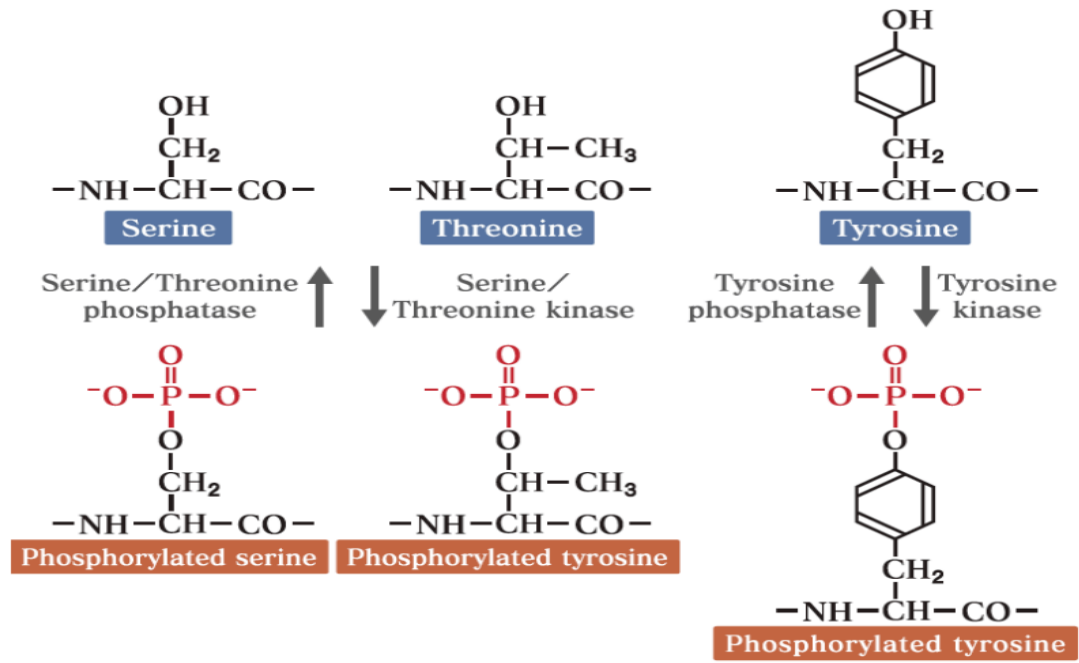


**Figure 2.1:** Phosphorylation on Serine, Threonine and Tyrosine residues

## 2.2 Machine Learning

Machine learning is a branch of Artificial Intelligence that depends on data and algorithms to imitate the human learning process to improve its performance. The rapidly expanding discipline of data science includes machine learning as a key element. Algorithms are taught to create classifications or predictions and to find significant insights in data mining projects via the use of statistical approaches. The decisions made as a result of these insights influence key growth indicators in applications and enterprises, ideally. Based on the learning strategy, there are four types of machine learning approaches: supervised, unsupervised, semi-supervised, and reinforcement.

In supervised learning, the algorithms are trained with a labeled dataset to learn functions and draw general rules to accurately predict outcomes and categorize data (Figure 2.2). This approach is analogous to how people learn in the classroom, where the teacher gives students some good instances, and the students infer the general principles from them.

In unsupervised learning, the algorithms automatically discover hidden patterns based on similarities and differences in unlabelled training data and split them into dif-

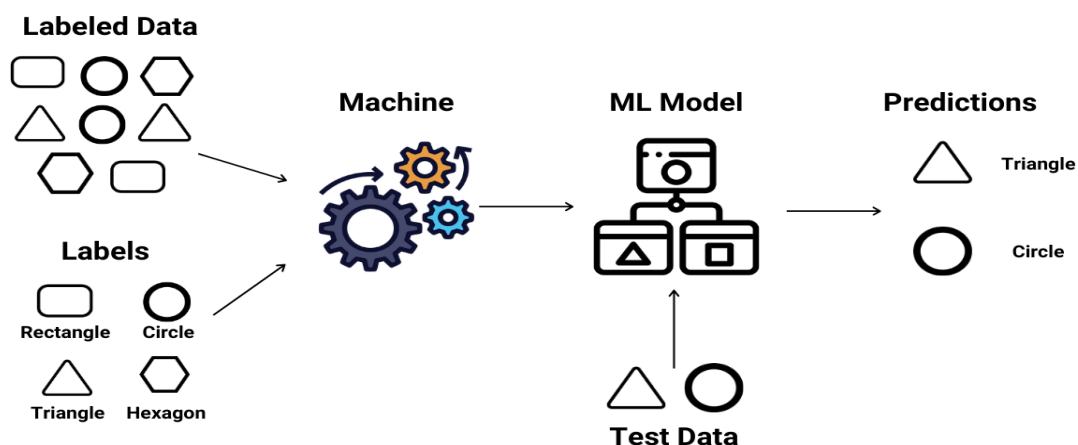


Figure 2.2: Supervised Machine Learning

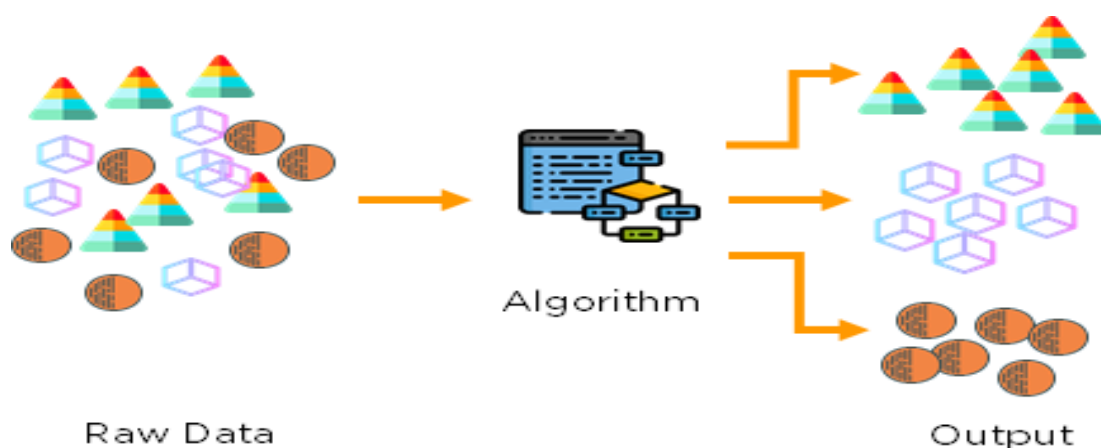


Figure 2.3: Unsupervised Machine Learning

ferent groups (Figure 2.3). Unsupervised learning is used for clustering, dimensionality reduction, and association rule mining.

Reinforcement Learning is a feedback-based Machine learning technique in which an agent (computer program or algorithm) learns to behave in an environment without any label data by directly interacting with it to execute the actions and see the results. The agent gets a reward for each right action and a penalty for each wrong action. While interacting with the environment, the agent aims to get the maximum positive rewards and thus improve performance (Figure 2.4).

Semi-supervised learning falls between unsupervised and supervised learning, where

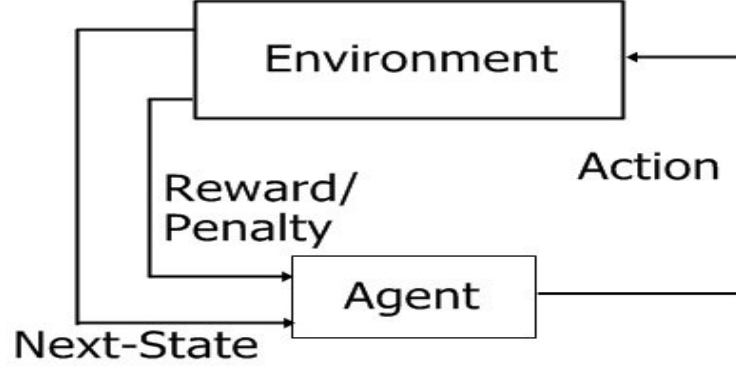


Figure 2.4: Reinforcement Learning

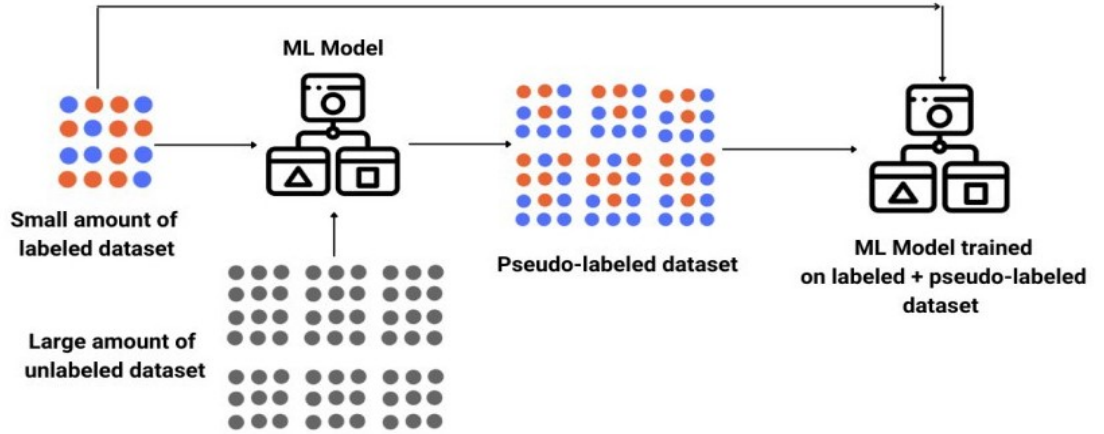


Figure 2.5: Semi-supervised Learning

the algorithms' training data includes small amounts of labeled samples and many unlabeled samples (Figure 2.5).

## 2.3 Deep Learning

Deep learning is a subset of machine learning based on artificial neural networks with multiple processing layers that automatically extract higher-level features from data. Unlike machine learning, deep learning can consume and process unstructured data like text and images, automatically extracting and selecting features without human intervention. Convolutional Neural Network(CNN), Recurrent Neural Network(RNN), Gated Recurrent Unit (GRU), and Long Short Term Memory(LSTM) are some most

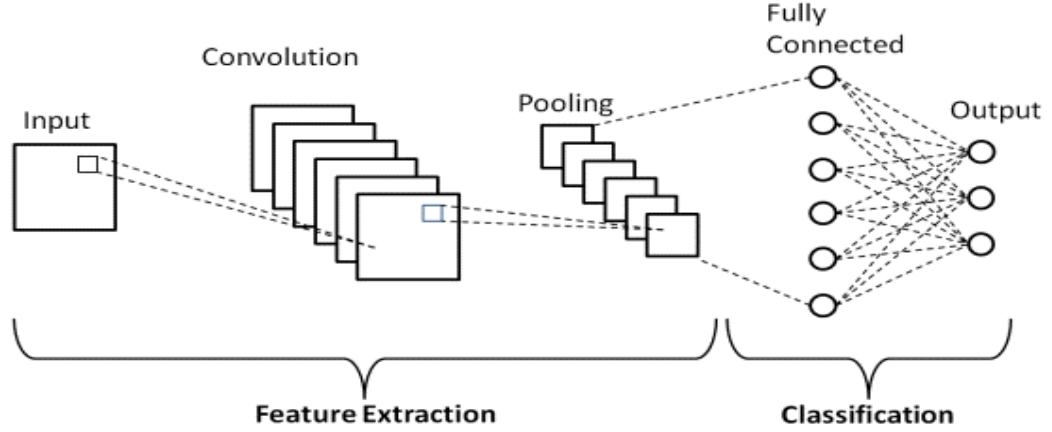


**Figure 2.6:** Architecture of a Convolutional Neural Network. Figure adapted from Analytical Vidhya blog

popular and basic Deep Learning techniques.

## 2.4 Convolutional Neural Network (CNN)

A convolution Neural Network is a particular neural network initially designed for two-dimensional data like image processing. The basic architecture of CNN consists of convolution, pooling, activation, and fully connected layers, as shown in Figure 2.6.

### 2.4.1 Convolution Layer

The convolution layer is CNN's most critical building block that serves as a feature extractor. In the convolution layer, a two-dimensional weight matrix that works as a feature detector and is known as a kernel or filter is applied to an input image, and the dot product between the image pixels covered by the filter and the filter's weights are calculated. The result of the dot product is then stored in an output array, and a stride (number of pixels that the filter moves over the image) shifts the filter. The process is repeated until the filter move across the entire image. While moving across the image, the filter's weights remain unchanged. The sequence of the dot products produces a final output array known as a feature map or activation map. The number of filters in a convolution layer determined the feature map's depth; two different filters generated a feature map of depth two. The padding operation is performed on the input image

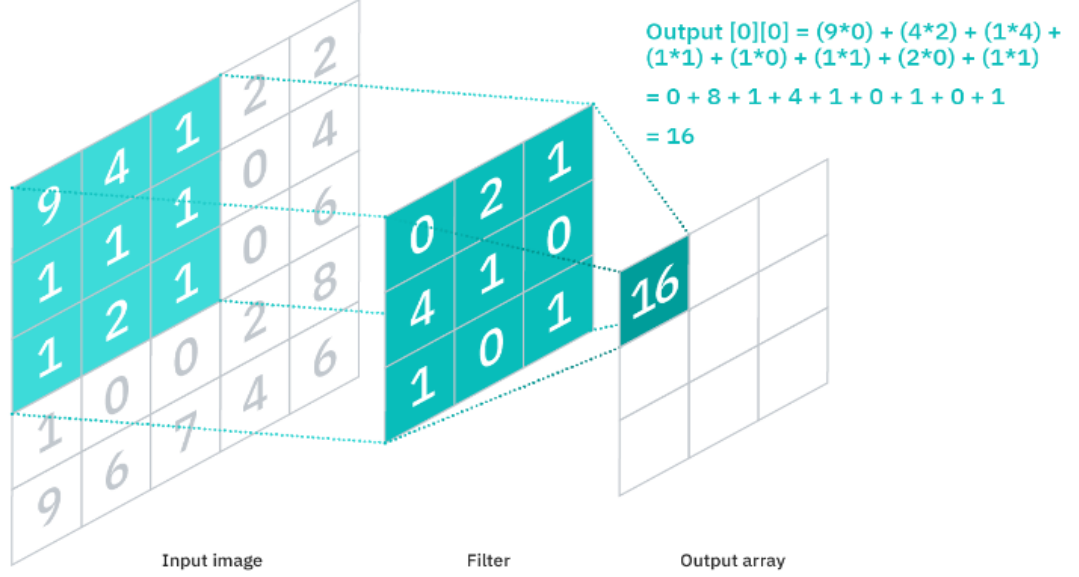


**Figure 2.7:** Convolution operation using 3 x 3 filter

in case the filter does not fit. A convolution operation using a 3x3 kernel is shown in Figure 2.7.

### 2.4.2 Pooling Layer

The pooling layer reduces the feature map's spatial dimension by extracting only dominant rotational and positional invariant features and thus minimizes the computational overhead of the training process. There are two types of Pooling: Max Pooling and Average Pooling. They respectively return the maximum value and average of all the values from the image/feature map region covered by the Kernel. The operation of max and average pooling with filter size ( $2 \times 2$ ) and stride 2 is depicted in Figure 2.8. Max pooling performs better than average pooling because it helps with dimension reduction and works as a noise suppressant by discarding noisy activation.

### 2.4.3 Fully Connected Layer

The fully connected layer is the final component of CNN. It receives features extracted in the previous convolution or pooling layers as one dimension vector and performs

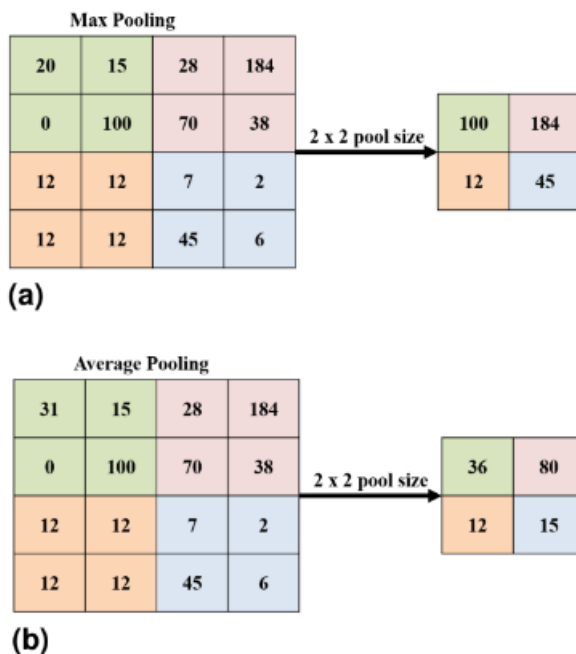


**Figure 2.8:** Pooling operation (a) max pooling, (b) average pooling.

classification. It exploits a softmax activation function that returns a probability distribution over the target variable to classify the input accurately.

Recently convolutional neural networks (CNN) have been widely exploited in computational biology to predict different biological and chemical functionalities and entities from multi-omics datasets. It has demonstrated incredible success in predicting different PTMs, cancer cell types, replication origins, anti-microbial peptides, and many more [25], [26], [27]. The main advantage of applying CNN in computational biology is that it can learn and extract relevant and intrinsic features from the protein’s row sequence or evolutionary or structural information without human intervention.

## 2.5 Related works

During the past few years, several machine learning-based predictors have been proposed to predict phosphorylation sites. The most promising predictors are PhosPred-RF [28], PhosphoSVM [29], NetPhos [30], PRED [31], Musite [32]. These methods have used features extracted from proteins’ primary sequence or secondary structure. For ex-

ample, PhosPred-RF [28] and PhosphoSVM [29] use sequence-based features, whereas PPRED [31] uses evolutionary information to identify phosphorylation sites. Sequence and structural features are combined in NetPhos [30] for independent and kinase-specific phosphorylation site prediction. PhosphoPredict integrates sequence-based and functional features to identify kinase-specific substrates and their corresponding phosphorylation sites. They also used different classification techniques including support vector machines (SVM), Random Forests (RF), gradient boosting trees (GBT), and AdaBoost to build their models.

More recently, several deep learning-based predictors have been proposed to predict Phosphorylation sites. Manual feature extraction is unnecessary for the deep learning-based approaches since they can automatically retrieve complicated patterns from protein sequences. MusiteDeep [33], DeepPPSite [34], DeepPhos [35], and Chlamy-EnPhosSite [36] are notable deep learning-based Phosphorylation site predictors. MusiteDeep uses one-hot encoding of protein sequence and convolutional neural network (CNN) with attention layer [33]. DeepPhos utilizes multi-layer CNN architecture consisting of densely connected convolutional blocks with different window and filter sizes [35]. DeepPPSite is constructed using a stacked long-short-term memory recurrent network [34], whereas Chlamy-EnPhosSite is an ensemble-based organism-specific predictor developed by combining CNN and LSTM [36]. DeepPPSite combines five distinct sequence-encoding approaches namely, sequence location information, amino acid composition descriptors, grouped-based features, and physicochemical property-based features. Unlike MisiteDeep and DeepPhos, where binary encoding is used, the embedding layer is employed in Chlamy-EnPhosSite to encode protein sequences.

Among all these approaches, only four computational methods for predicting phosphorylation sites in microbial organisms are available to date. The initial two methods NetPhosBac [37] and cPhosBac [38], are bacteria-specific protein phosphorylation site predictors. The former is created by implementing an artificial neural network algorithm. The latter utilizes k-spaced amino acid pairs (KSAAP) composition for sequence encoding and SVM for classification. The predictors are trained on the same dataset, consisting of 152 experimentally confirmed phosphorylated serine/threonine sites in 119 substrates. The cPhosBac outperforms the NetPhosBac. On the other hand, prkC-PSP was proposed by Zhang et al. as a prkC-specific phosphorylation site predictor [39]. It extracts amino acid location information-based features from the protein sequence and

use SVM as the classification technique to distinguish probable prkC-specific phosphorylation sites. The dataset contains experimentally identified 36 phosphorylation and 512 non-phosphorylation sites curated manually from the literature. In 2019, Mamun et al., developed a general microbial phosphorylation site predictor named MPsite by using enhanced characteristics of sequence as features and Random Forest as the classification technique [40]. To build this model, they used Wilcoxon rank-sum test (WR) to select the optimal set of features. The dataset used in this study was collected from the dbPSP, consisting of 2045 pS sites in 1940 proteins and 2174 pT sites in 1534 proteins. MPsite shows more promising performance than the existing microbial phosphorylation site predictors.

## 2.6 Research Gaps

It is important to note that despite all attempts that have improved microbial phosphorylation site prediction and established good benchmarks for forthcoming studies in recent years, some research gap exists in this domain that encourages future investigations.

First, the overall performance of the aforementioned microbial phosphorylation site predictors is not satisfying. We observed that only amino acid composition-based features were used in these predictors, ignoring the structural and evolutionary information, which may be the possible cause of such inadequate performance. However, previous research on protein subcellular localization [41], bacteriophage protein identification [42], and protein succinylation and malonylation prediction [15], [43] have shown that extracting features from the protein’s structural and evolutionary information improves prediction performance.

Second, all prior predictors relied on manual feature extraction mechanisms and used typical machine learning methods. However, formulating problem-specific appropriate feature representations for protein sequences is challenging and requires many cycles of try and error. Therefore, looking at automated feature extraction using deep learning approaches is necessary.

Third, phosphorylation on tyrosine (Y) residue has not been explored yet. All the previous predictors focused only on detecting phosphorylation on serine (S) and threonine (T) sites.

This study proposes two predictors that address all the abovementioned shortcomings. First, we developed a traditional machine learning-based predictor using Rotation Forest that exploits manual structural and evolutionary features. Second, we propose a two-headed CNN incorporating squeeze and excitation blocks that simultaneously learn compelling features from the peptide’s evolutionary and structural information and predict phosphorylation sites. Finally, our proposed predictors can predict phosphorylation on serine (S), threonine (T), and tyrosine(Y) residues.

## 2.7 Summary

This chapter has reviewed different computational methods available in the literature on phosphorylation sites prediction and highlighted their shortcomings, generating potential research gaps. The next chapter describes our proposed solution based on machine learning, presents experimental results, and discusses them.

## Chapter 3

# RotPhoPred: An ensemble based method

In this chapter, we propose a new machine learning-based predictor called RotPhoPred to accurately predict phospho-serine (pS), phospho-threonine (pT), and phospho-tyrosine (pY) in the protein sequence of microbial organisms, which integrates both structural and evolutionary information. Our predictors coalesce predicted structural features and evolutionary bigram profiles to describe each peptide fragment in the dataset. We also use the NearMiss-3 undersampling technique to balance the dataset to avoid bias towards larger class set. Subsequently, we use Rotation Forest classifier which is an ensemble-based machine learning classifier to predict microbial phosphorylation on serine (S), threonine (T), and tyrosine (Y) residues. We then use 5-folds cross-validation and independent test set to assess the prediction performance of the predictors. The overall flowchart of the proposed method is shown in Figure 3.1.

### 3.1 Material and Methods

In this section, we describe the benchmark dataset that is used in this study and present our proposed methodology to build RotPhoPred.

#### 3.1.1 Benchmark datasets

We have collected the protein sequences with experimentally verified pS, pT, and pY PTMs from the dbPSP database [7]. The assembled dataset contains redundant pro-

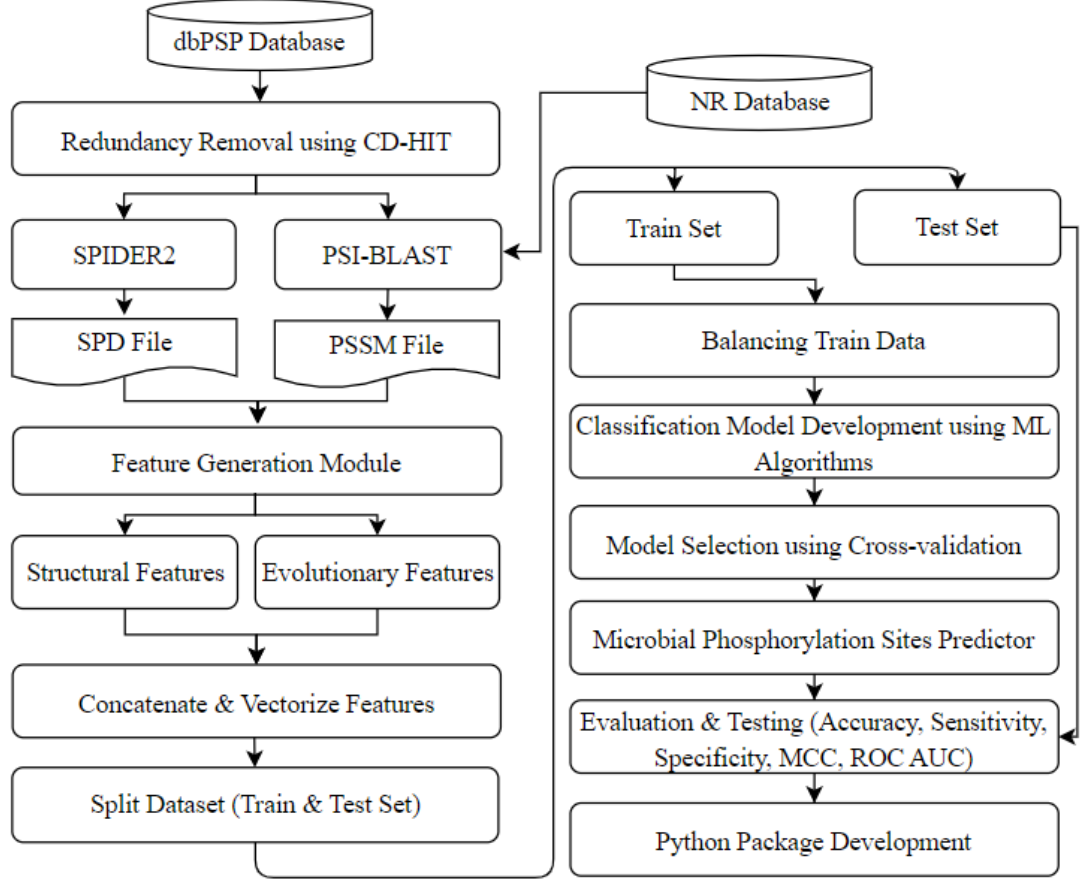


**Figure 3.1:** The overall flowchart of the RotPhoPred.

teins. To remove redundancy, we use CD-HIT to remove those proteins with over 40% sequential similarity [44][45][46]. The final datasets contain 1483, 1220, and 1161 protein sequences for pS, pT, and pY PTMs, respectively. The pS dataset consists of a total 36,513 instances with 2024 phosphorylated (positive) and 34489 no-phosphorylated (negative) sites. There are 26239 samples in the pT dataset, with 1647 positive and 24592 negative sites. The pY dataset comprises of total 17476 instances where 1644 are positive and 15832 are negative. A summary of the datasets used in this study is given in Table 3.1.

To avoid overfitting and assess our model’s generality, 15% of sequences consisting of 223, 183, and 174 proteins for pS, pT, and pY sites prediction tasks are arbitrarily separated to form independent test sets. The remaining proteins in the datasets are used to train the classifiers (training dataset).

### 3.1 Material and Methods

| Tasks | Total Sequences | Positive Sites | Negative Sites | Positive: Negative |
|-------|-----------------|----------------|----------------|--------------------|
| pS    | 1483            | 2040           | 34489          | 1:17               |
| pT    | 1220            | 1647           | 24952          | 1:15               |
| pY    | 1161            | 1644           | 15832          | 1:10               |

**Table 3.1:** The summary of the datasets for pS, pT, and pY identification problems.

#### 3.1.2 Features

To extract the evolutionary and structural information, the protein sequences from the benchmark dataset are fed to PSI-BLAST [47] and SPIDER2 [48][49]. Using PSI-BLAST, we generate position-specific scoring matrix (PSSM) file, and using SPIDER2, we generate an SPD file. The PSSM calculates the likelihood of replacing each protein’s amino acid with the other 20 amino acids based on their location. On the other hand, the predicted secondary structure probabilities, accessible surface area (ASA), and torsion angles for each amino acid residue are described in the SPD file in matrix format. The evolutionary bigram profile and structural features are then constructed from the PSSM and SPD files. Later, the different feature groups extracted from PSSM and SPD files are combined to form a feature vector. After feature vectorization, we split the data into training and test sets. A NearMiss-3 under-sampling technique is then applied to the training set to address the imbalance issue while keeping the test data untouched [50].

#### 3.1.3 Formulation of peptide fragments for each site

In this study, we used a window-based approach to represent each positive or negative site. Each phosphorylation or non-phosphorylation site is described by a peptide fragment  $P$  of  $2n+1$  residues with a S/T/Y in the center,  $n$  upstream residues at the right, and  $n$  downstream residues at the left as follows:

$$P = \{A_{-n}, A_{-(n-1)}, \dots, A_{-2}, A_{-1}, S/T/Y, A_1, A_2, \dots, A_{+(n-1)}, A_{+n}\} \quad (3.1)$$

where  $A_{-i}$  and  $A_{+i}$  represent the upstream and downstream amino acids respectively, and S, T, Y represents serine, threonine, and tyrosine, respectively. If  $n$  upstream or downstream residues are not available in the protein to describe the S/T/Y sites, the

mirroring technique is used to fill the gap of missing amino acids as shown in Figure 3.2.



**Figure 3.2:** The representation of serine residue with its upstream and downstream amino acids. (a) Serine residue with enough neighbors on the upstream and downstream sides. (b) Serine residue with inadequate neighbors, either upstream or downstream.

After analyzing the performance of different window sizes, we choose  $n = 10$  since it exhibits the best performance. Figure 3.2 demonstrates the overall windowing process for serine residue.

### 3.1.4 Evolutionary feature extraction

As it was mentioned earlier, we use PSSM to extract evolutionary information. We generated the PSSM for each protein sequence in our benchmark dataset by running the PSI-BLAST algorithm for three iterations on the non-redundant (nr) database provided by NCBI with a cutoff (E) value of 0.001. The PSSM is an  $L \times 20$  matrix, where  $L$  is the protein sequence's length, and the 20 columns denote different amino acids of the genetic code.

It was shown in previous studies that using bigram, we can extract important discriminatory information for the classification task from PSSM for similar problems

[15, 43, 51–53]. Moreover, bigram feature size is independent from the window. It extracts a 400-dimensional feature vector to capture evolutionary information regardless of the number of upstream and downstream residues. As a result, we may expand the number of residues surrounding the S/T/Y site without increasing the number of features. This study generates bigram probabilities for each protein segment to apprehend its evolutionary profile. To generate the bigram profile from the evolutionary information, the submatrix  $M$  for the peptide fragment  $P$  that describes a phosphorylated or non-phosphorylated site is segmented from the PSSM matrix.  $M$  is a  $W * 20$  dimensional matrix where  $W$  is the window size ( $W = 21$  consisting of 10 upstream, 10 downstream, and one central S/T/Y residue), as mentioned in section 3.1.3. Each element  $m_{i,j}$  of the matrix  $M$  represents the transitional probability of  $j$ th amino acid at the  $i$ th position in the peptide fragment  $P$ . Then the bigram profile of the submatrix is calculated using the following equation:

$$B_{p,q} = \sum_{k=1}^{20} m_{k,p} m_{k+1,q}, \text{ where } 1 \leq p \leq 20 \text{ and } 1 \leq q \leq 20 \quad (3.2)$$

The resulting 20x20 dimensional matrix  $B$  represents the PSSM profile bigram of peptide fragment  $P$ . Subsequently, the matrix  $B$  is converted to a 400-dimensional row vector denoted by  $F1$  as shown in Equation 3.3.

$$F1 = [B_{1,1}, B_{1,2}, \dots, B_{1,20}, B_{2,1}, B_{2,2}, \dots, B_{2,20}, \dots, B_{20,1}, B_{20,2}, \dots, B_{20,20}] \quad (3.3)$$

#### 3.1.5 Structural feature extraction

Along with the evolutionary information, the structural properties of the protein have been shown to be effective to predict other PTMs [43][54][55][56][57][15]. The protein’s structural properties include secondary structures, accessible surface area (ASA), and torsion angles.

The secondary structure depicts each amino acid residue in a number of distinct configurations, the most frequent of which are helix, sheet, and coil. Local backbone angles also define the local protein structures through torsion angles between neighboring amino acids. Unlike secondary structure, which provides a coarse-grain description of local configuration in terms of 3 discrete shapes - coil, strand, or helix, local backbone angles give continuous information about the local structure concerning

four angles. The four angles include two backbone torsion angles,  $\psi$ , and  $\phi$ , which indicate the angles between atoms along the protein backbone, and dihedral angles  $\theta$  and  $\tau$ , which represent the rotation angles. The secondary structure and backbone angles describe which amino acids are more dissembled and prone to interact with other macromolecules.

ASA measures how much an amino acid residue area is exposed to solvent (water) in a protein. The amino acid residue on the protein’s surface area has a high chance of undergoing PTMs. Hence, ASA is an essential structural property for phosphorylation prediction. As it was mentioned earlier, we used SPIDER2 to predict the values of the parameters mentioned above for each amino acid residue in a protein sequence. SPIDER is a deep learning-based tool that achieves promising results for predicting secondary structure, backbone angles, and ASA from protein sequences [48] [49]. It produces a matrix of  $L \times 8$  dimensions that contain the predicted values of eight structural properties (coil, strand, helix, asa,  $\psi$ ,  $\phi$ ,  $\theta$ ,  $\tau$ ) for each amino acid residue in a protein of length  $L$ .

To capture the structural information of each peptide segment  $P$  in our dataset, we extract the submatrix  $A$  from the SPD file of the protein sequence that contains  $P$  and flattens it to form a row vector  $F2$ . The matrix  $A$  is  $W \times 8$  dimensional, where  $W$  is the window size. We investigated different values for window size which among them, using 21 for serine (S), threonine (T), and tyrosine (Y) demonstrates the best performance. Hence,  $A$  is a  $21 \times 8$  dimensional matrix. The resultant row vector is 168 dimensional denoted by  $F2$  as shown in Equation 3.4.

$$F2 = [B'_{1,1}, B'_{1,2}, \dots, B'_{1,8}, B'_{2,1}, B'_{2,2}, \dots, B'_{2,8}, \dots, B'_{21,1}, B'_{21,2}, \dots, B'_{21,8}] \quad (3.4)$$

#### 3.1.6 Formation of feature vector

After extracting the bigram profile from PSSM and structural features from SPD, we integrate both feature groups to form a feature vector to predict microbial phosphorylation. The resultant 568 ( $400 + 168$ ) dimensional feature vector captures the critical structural and evolutionary information essential to discriminating phosphorylated and non-phosphorylated sites.

$$F = [B_{1,1}, \dots, B_{1,20}, \dots, B_{20,1}, \dots, B_{20,20}, B'_{1,1}, \dots, B'_{1,8}, \dots, B'_{21,1}, \dots, B'_{21,8}] \quad (3.5)$$

### 3.1.7 Balancing dataset

The number of non-phosphorylated sites (negative samples) is greater than the number of phosphorylated sites (positive samples) in our benchmark datasets, as shown in Table 1. Such imbalance could influence any machine learning-based predictor to be biased towards the negative sample. Therefore, balancing the training dataset is critical for developing a bias-free predictor. Two main ways to balance the dataset are under-sampling and over-sampling. While the former keeps all samples in the rare class and reduces the abundant type, the latter increase the size of the infrequent category by generating artificial instances. This paper analyzes various balancing techniques from both under-sampling and over-sampling categories, including ADASYS [58], SMOTE [59], Tomek Links [60], and NearMiss [50]. Our results demonstrate that NearMiss-3 which is a down-sampling technique exhibited the best performance. Hence, we used the NearMiss-3 technique for balancing our training dataset.

NearMiss-3 selects the given number of closet samples from the majority class (negative) for each instance in the minority class based on the Euclidean distance. Consequently, it picks the more information-rich non-phosphorylated sites, which are vital for designing a powerful decision boundary to differentiate phosphorylated and non-phosphorylated sites [50]. Moreover, it does not produce artificial samples thus minimizing the computational cost when fitting the model. We implemented NearMiss-3 to select one closet non-phosphorylated site for each phosphorylated site. As a result, the transformed training dataset includes an equal number of phosphorylated and non-phosphorylated sites. Note that the balancing is not performed on the independent test set used to evaluate the performance of the model to avoid overfitting.

### 3.1.8 Classification model

In this study, to build RotPhoPred, we use Rotation Forest (RoF) algorithm since it exhibits encouraging performance in similar studies found in the literature [41][15][61][62][63][64][65]. Rotation forest is an ensemble learning technique that trains  $N$  base classifiers separately in parallel and predicts class labels based on the majority of soft voting [66]. Unlike Random Forest that uses a random subset of features, the rotation forest uses a transformed feature space to build the individual base learner. To build each base classifier, it randomly splits the feature set into  $K$  subsets, and for

each subset, a bootstrap sample of size 75% of the original dataset is drawn. Then the Principal Component Analysis (PCA) [67] is then performed on the selected samples to transform the feature vector linearly, to enhance diversity among the base learners. Later, the  $K$  transformed feature subsets are combined to form a feature vector to train the base classifier. RoF utilizes decision trees as the base learners since they are accurate and sensitive to the rotation of the feature axes. In this study, we used the `rotation_forest` package available in python3 with 100 decision trees as base estimators as it was shown the effective number in previous studies [41][15]. The `max_features` parameter was set to 'auto' and `n_jobs`, indicating the number of jobs to run parallel, was set to -1 to force the algorithm to use all the functional processors.

#### 3.1.9 Validation scheme

A wide range of validation schemes, including the k-folds cross-validation and jack-knife test, are reported in the literature to evaluate the efficacy of machine learning-based predictors. In this paper, we have utilized the stratified 5-folds cross-validation to avoid overfitting and assess our performance with different parameter settings. The cross-validation is performed as follows:

1. Divide the dataset into five disjoint folds of equal size by maintaining the percentage of instances from both classes in each fold.
2. Fit the predictor on 4-folds, and evaluate its performance on the remaining fold via different metrics such as sensitivity, specificity accuracy, and MCC.
3. Repeat step 2 five times and calculate each metric's average.

Figure 3 graphically demonstrates the overall process of the 5-fold cross-validation scheme.

#### 3.1.10 Evaluation metrics

To evaluate the performance of RotPhoPred, we use six metrics namely, Sensitivity (Sn), Specificity (Sp), Precision (Pr), Accuracy (Ac), F1 Measure (F1), Mathews' correlation coefficient (MCC), and area under the ROC curves (AUC) to evaluate the performance of the proposed method.

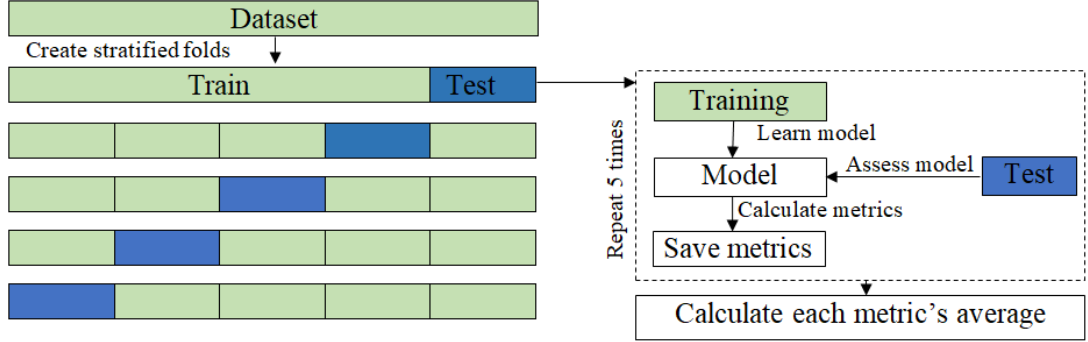


**Figure 3.3:** Schematic overview of five folds cross-validation scheme.

Sensitivity measures the predictor’s ability to identify phosphorylated sites accurately. It quantifies how many phosphorylated sites the predictor can accurately detect out of the total number of phosphorylated sites. Sensitivity ranges from 0 to 100 percent. The predictor detecting all phosphorylated sites will receive sensitivity of 100.

Specificity assesses how well the predictor performs in detecting non-phosphorylated sites. Hence, it is the ratio of the total number of successfully identified non-phosphorylated sites by the predictor to the actual number of non-phosphorylated sites. The value of specificity can be between 0 to 100 percent.

The predictor’s ability to discriminate between phosphorylated and non-phosphorylated sites is measured by accuracy. It summarizes the predictor’s overall performance with a single score. Accuracy ranges from 0% to 100%, with 100% indicating the most accurate prediction.

Precision is the ratio of correctly predicted phosphorylated sites to all predicted phosphorylated sites by the model.

F-Measure summarizes the sensitivity and precision through a single measure by computing their harmonic mean. The score reflects how well the predictor can balance the sensitivity and precision. It also ranges from 0 to 1, with 1 denoting perfect balance. It is the most common metric used to measure a model’s performance developed on an imbalanced dataset.

Matthews correlation coefficient (MCC) is the most reliable statistical metric for binary classifiers when both classes are of interest and the size differs. It considers the actual and predicted classes as two variables and calculates the correlation coefficient between them. It fluctuates from -1 to 1. The higher the correlation between actual

and predicted classes, the better the prediction. MCC is 1 for the perfect predictor, indicating a perfect positive correlation. Conversely, when the predictor consistently makes incorrect predictions, the MCC value drops to -1, representing the perfect negative correlation. Respectively, MCC of 0 represents no correlation. The Sn, Sp, Pr, Ac, F1, and MCC are calculated as follows:

$$Sn = \frac{TP}{TP + FN} \quad (3.6)$$

$$Sp = \frac{TN}{TN + FP} \quad (3.7)$$

$$Pr = \frac{TP}{TP + FP} \quad (3.8)$$

$$Ac = \frac{TP + TN}{TP + FN + FP + TN} \quad (3.9)$$

$$F1 = \frac{2 * Pr * Sn}{Pr + Sn} \quad (3.10)$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TN + FN) * (TP + FP) * (TN + FP) * (TP + FN)}} \quad (3.11)$$

where,  $TP$  (True Positive) indicates the number of correctly identified phosphorylation sites,  $TN$  (True Negative) means the number of correctly identified non-phosphorylation sites,  $FP$  (False Positive) represents the number of incorrectly identified non-phosphorylation sites as phosphorylated sites, and  $FN$  (False Negative) denotes the number of incorrectly identified phosphorylation sites as non-phosphorylated sites. We also used the Receiver Operating Characteristics (ROC) curve to evaluate the predictor's performance graphically. ROC curve plots the true positive rate against the false-positive rate for different classification thresholds. The area under the ROC curve can also quantify the predictor's performance. The higher the area under the ROC (AUC) value, the better the predictor. A perfect predictor will achieve the highest score in all of these metrics.

## 3.2 Results and Discussion

In this section we present our results, compare them with previous studies, and discuss their significance.

### 3.2.1 Feature significance analysis

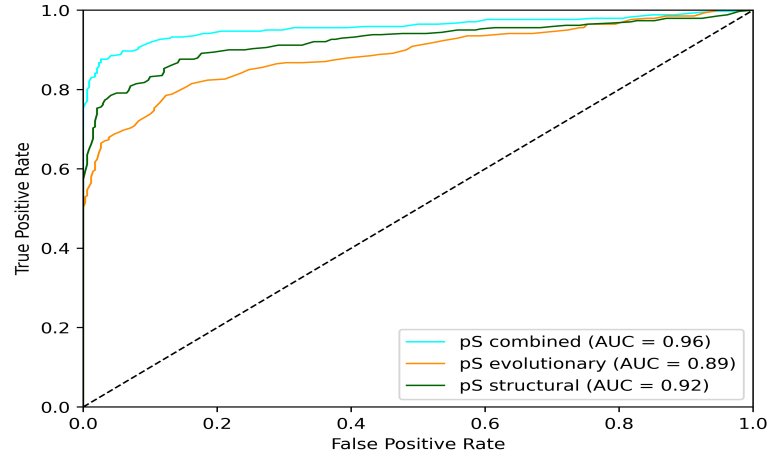
This section investigates the significance of extracted features in phosphorylation site prediction. Three sets of features namely structural, evolutionary, and combination of both have been compared using the Rotation Forest classifier. The results achieved using the 5-folds cross-validation are presented in Table 3.2 for pS, pT, and pY, prediction tasks. As shown in this table, the evolutionary feature extracted from the PSSM achieves better results for the pS site prediction. In contrast, the structural feature provides more discriminatory information for pT and pY site identification. However, the best prediction performance in terms of sensitivity, specificity, precision, accuracy, and MCC is reported with integrated structural and evolutionary features. This pattern is consistent for all three predictors to identify phosphorylation on serine, threonine, and tyrosine residue. Such consistency justifies the significance of evolutionary and structural features in microbial phosphorylation prediction.

We also plot the ROC curves for structural, evolutionary, and combined features in predicting pS, pT, and pY sites using the independent test set. The plots are shown in Figure 3.4. The curves illustrate that the best AUC values of 0.96, 0.90, and 0.91 are achieved respectively for pS, pT, and pY site prediction tasks when structural and evolutionary features are combined.

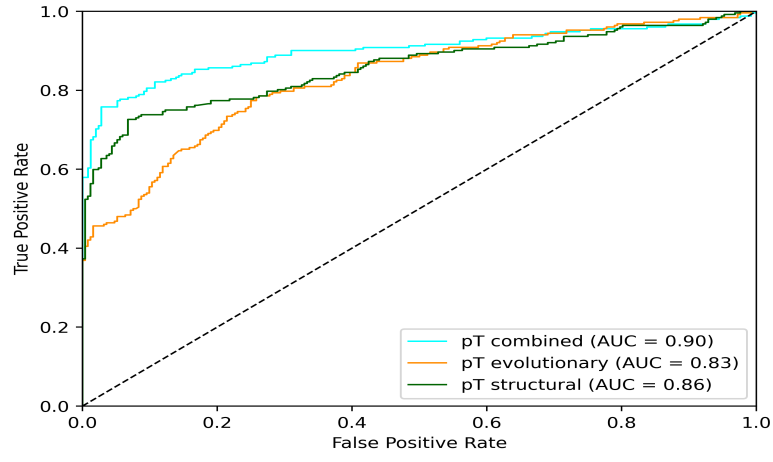
| Task | Features     | Sn (%)            | Sp(%)             | Pr                | Ac(%)             | F1                | MCC               | AUC               |
|------|--------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| pS   | Combined     | <b>87.7(0.03)</b> | <b>97.6(0.01)</b> | <b>0.93(0.01)</b> | <b>92.7(0.01)</b> | <b>0.92(0.01)</b> | <b>0.86(0.02)</b> | <b>0.93(0.01)</b> |
|      | Evolutionary | 81.1(0.03)        | 93.0(0.02)        | 0.92(0.02)        | 87.0(0.01)        | 0.86(0.01)        | 0.75(0.02)        | 0.87(0.01)        |
|      | Structural   | 81.1(0.02)        | 91.6(0.02)        | 0.91(0.02)        | 86.4(0.01)        | 0.86(0.01)        | 0.73(0.02)        | 0.86(0.01)        |
| pT   | Combined     | <b>79.4(0.02)</b> | <b>99.3(0.01)</b> | <b>0.99(0.01)</b> | <b>89.3(0.01)</b> | <b>0.88(0.01)</b> | <b>0.80(0.02)</b> | <b>0.89(0.01)</b> |
|      | Evolutionary | 50.9(0.01)        | 98.9(0.01)        | 0.98(0.02)        | 74.9(0.01)        | 0.67(0.01)        | 0.57(0.02)        | 0.75(0.01)        |
|      | Structural   | 75.4(0.02)        | 97.8(0.01)        | 0.97(0.01)        | 86.6(0.01)        | 0.85(0.02)        | 0.75(0.02)        | 0.87(0.01)        |
| pY   | Combined     | <b>75.2(0.03)</b> | <b>99.3(0.01)</b> | <b>0.99(0.00)</b> | <b>87.3(0.02)</b> | <b>0.85(0.02)</b> | <b>0.77(0.03)</b> | <b>0.87(0.02)</b> |
|      | Evolutionary | 54.2(0.03)        | 97.1(0.01)        | 0.95(0.02)        | 75.6(0.01)        | 0.69(0.02)        | 0.57(0.02)        | 0.76(0.01)        |
|      | Structural   | 67.7(0.02)        | 95.7(0.01)        | 0.94(0.01)        | 81.7(0.01)        | 0.79(0.01)        | 0.66(0.02)        | 0.82(0.01)        |

**Table 3.2:** Impact of different features on the prediction performance using 5-folds cross validation

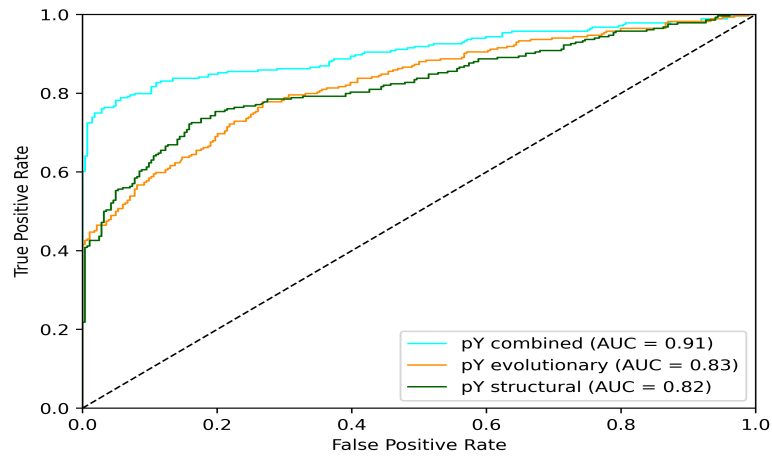
We created another predictor by extracting the bigram profile from the structural and evolutionary peptide matrixes and training a rotation forest algorithm to examine how the structural information’s bigram profile influences the prediction performance. We named it StrucBigram. Table 3.3 compares StrucBigram with RotPhoPred on the independent test set for predicting pS, pT, and pY sites. The table shows that the performance degrades for all prediction tasks when the bigram feature of structural



(a)



(b)



(c)

**Figure 3.4:** ROC curves for different feature groups for (a) pS, (b) pT, and (c) pT sites identifications on independent test

## 3.2 Results and Discussion

| Task | Model       | Sn(%)       | Sp(%)       | Pr          | Ac(%)       | F1          | MCC         | AUC         |
|------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| pS   | RotPhoPred  | <b>90.0</b> | <b>92.1</b> | <b>0.92</b> | <b>91.0</b> | <b>0.91</b> | <b>0.82</b> | <b>0.96</b> |
|      | StrucBigram | 88.5        | 82.5        | 0.29        | 83.0        | 0.44        | 0.45        | 0.86        |
| pT   | RotPhoPred  | 75.4        | <b>97.2</b> | <b>0.96</b> | <b>86.3</b> | <b>0.85</b> | <b>0.74</b> | <b>0.90</b> |
|      | StrucBigram | <b>88.9</b> | 82.4        | 0.33        | 83.0        | 0.48        | 0.48        | 0.86        |
| pY   | RotPhoPred  | 78.2        | <b>94.7</b> | <b>0.94</b> | <b>86.4</b> | <b>0.85</b> | <b>0.74</b> | <b>0.89</b> |
|      | StrucBigram | <b>82.0</b> | 74.2        | 0.38        | 75.5        | 0.52        | 0.43        | 0.78        |

**Table 3.3:** Impact of structural bigram profile in the prediction performance on the independent test set. Bold items indicate the highest values.

information is utilized, as indicated by the MCC scores of StrucBigram. RotPhosPred achieves the highest performance for the pS site prediction in all evaluation measures. While predicting pT and pY sites, although StrucBigram attains 13.5% and 3.8% better sensitivity, its performance in other metrics is significantly lower than RotPhosPred. StrucBigram fails to detect 17.5%, 17.6%, and 25.8% non-phosphorylated serine, threonine, and tyrosine sites, respectively. Besides, it misclassified many nonphosphorylated sites as phosphorylated sites, as apparent from the lower precisions of 0.29, 0.33, and 0.38 for the detection of pS, pT, and pY sites. The results also indicate that the StrucBigram predictor is biased towards the positive class. The possible cause behind such biases can be the minimization of features due to the bigram operation on structural information. While the bigram operation is performed, the structural features are reduced from 168 to 64. Consequently, RotPhosPred and StrucBigram are trained on the 568 and 464 features, respectively.

### 3.2.2 Comparison with different classifiers

In this section, we analyze the performance of different machine learning algorithms for phosphorylation prediction. To do this, we use five different classifiers namely, Rotation Forest (RoF), Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting Tree (GBT), and Adaptive Boosting (AdaBoost). The hyperparameters of all the classifiers are tuned using cross-validation. We created the Rotation Forest exploiting the *rotation-forest* package available in python3 with 100 decision trees as base estimators. The *max\_features* parameter was set to *auto*, and *n\_jobs*, indicating the number of jobs to run parallel, was set to -1 to force the algorithm to use all the

| Classifiers            | Hyper-parameters                                                                                                       |
|------------------------|------------------------------------------------------------------------------------------------------------------------|
| Rotation Forest        | Base estimator = Decision Tree<br>Number of tree = 100<br>Maximum features = ‘auto’<br>Number of jobs = -1             |
| Random Forest          | Number of tree: 100<br>Splitting criteria : ‘gini’                                                                     |
| Support Vector Machine | Kernel: ‘polynomial’<br>Regularization, C: 1.0                                                                         |
| Gradient Boosting      | Loss : ‘log_loss’<br>Learning rate = 0.1<br>Splitting criteria = <i>friedman_mse</i><br>Number of boosting stage = 100 |
| Adaptive Boosting      | Base estimator = Decision Tree<br>Learning rate = 0.1<br>Maximum number of estimator = 100                             |

**Table 3.4:** Hyper-parameters summary of different classifiers

functional processors. All other machine-learning algorithms are implemented using the *scikit-learn* package in Python. We used a polynomial kernel for SVM to make the samples linearly separable. Besides, set the regularization parameter  $c = 1$ , which optimizes the hyperplane’s margin and minimizes the misclassification of training data. The Gradient Boosting is executed for 100 boosting stages to minimize the log loss with a learning rate of 0.1. The *friedman\_mse* splitting criterion was used to measure a split’s quality. A decision tree was used as the base estimator for adaptive boosting, which was constructed to execute up to 100 boosting iterations. Each classifier’s weight at each boosting iteration was set to 0.1 via the hyperparameter *learning\_rate*. We built the random forest model with 100 decision trees and Gini-impurity splitting criteria by setting *n\_estimator* = 100 and *criterion* = *gini*, respectively. Other parameters of the classifiers are kept default as in the *rotation-forest* and *scikit-learn* packages. A summary of the hyperparameters settings of different machine learning classifiers used in this study for comparison is shown in Table 3.4.

For this experiment, we extracted the structural and evolutionary features from each site represented by 21 amino acid residue windows. The experiments using 5-folds

cross-validation are shown in Table 3.5, where the mean values of different performance metrics are reported.

Table 3.6 shows the independent test results for the various classifiers where the rotation forest(RoF) reporting the best MCC scores for all three predictors for recognizing pS, pT, and pY sites.

The results shows that the RoF beats all the classifiers in sensitivity, accuracy, f1-measure, MCC, and AUC in pS sites identification. The specificity and precision of RoF are also reasonable since they are not significantly lower than the highest values achieved by SVM.

For pT sites predictin task, RoF outperforms all other classifiers in all metrics except the sensitivity. While, Gradient boosting achieves the best sensitivity. However, the sensitivity (75.4%) of the RoF is still comparable to the best results.

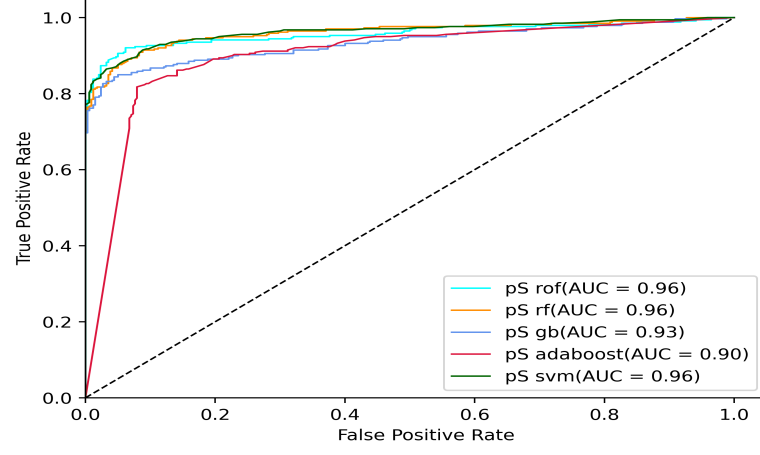
Among all the classifiers, RoF shows the highest specificity, precision, accuracy, f1-measure, and AUC values of 94.7%, 0.93, 86.4%, 0.85, and 0.86 in predicting pY sites. It is also competitive in terms of sensitivity. Hence, we use this classifier to build RotPhoPred.

Figure 3.5 illustrates the ROC curves of different classifiers for predicting pS, pT, and pY sites on the independent test set. It can be seen from this figure that the AUC values of RoF in predicting pS, pT and pY sites are 0.96, 0.90, and 0.89 respectively which are the highest among the classifiers.

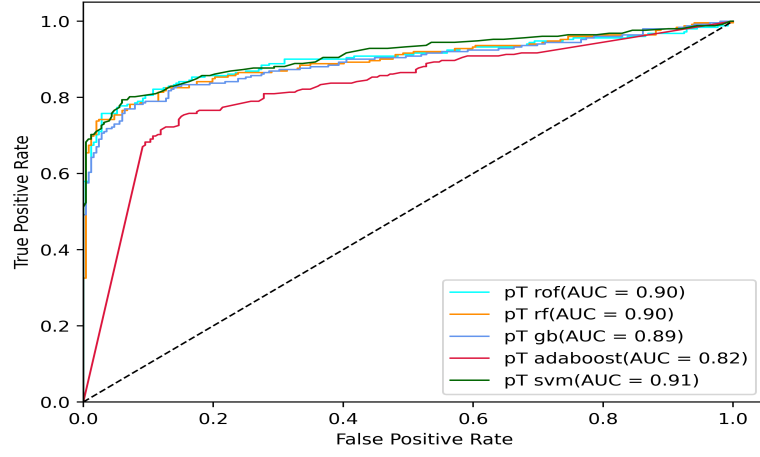
All these results demonstrate the effectiveness of RoF in predicting microbial phosphorylation sites. The secrete behind the superiority of RoF is its ability to do implicit feature selection and introduce diversity in each base classifier by feature transformation using PCA [66][67].

### 3.2.3 Comparison with current state-of-the-art predictors

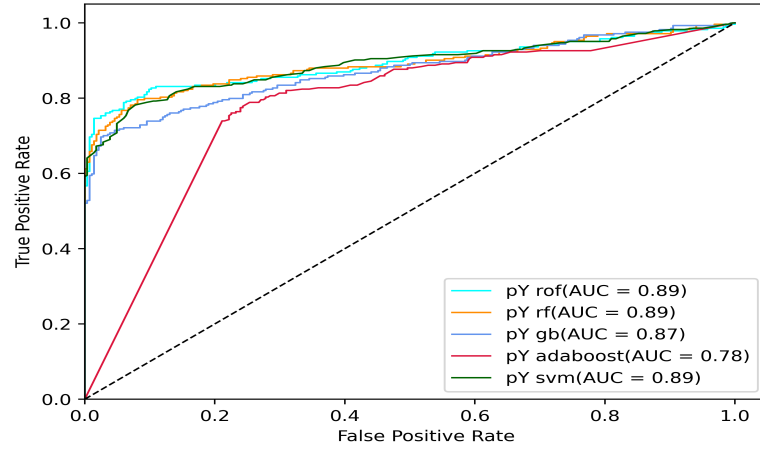
In this section, we compare RotPhoPred with MPsite and NetPhosBac on the independent test set as the two best microbial phosphorylation cite predictors. To do this, we fed the independent test as fasta files to MPSite and NetPhosBac servers and collected the predicted result for each site on the dataset. Then, we characterized the performances of these predictors in terms of sensitivity, specificity, accuracy, and MCC. The



(a)



(b)



(c)

**Figure 3.5:** ROC curves for different classifiers for (a) pS, (b) pT, and (c) pY sites identifications on independent test

### 3.2 Results and Discussion

| Task | Classifier | Sn(%)             | Sp(%)             | Pr                | Ac(%)             | F1                | MCC               | auROC             |
|------|------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| pS   | RoF        | <b>87.7(0.03)</b> | 97.6(0.01)        | 0.97(0.01)        | <b>92.7(0.01)</b> | <b>0.92(0.01)</b> | <b>0.86(0.02)</b> | <b>0.93(0.01)</b> |
|      | SVM        | 80.4(0.03)        | <b>99.6(0.01)</b> | <b>0.99(0.00)</b> | 90.0(0.02)        | 0.89(0.02)        | 0.82(0.03)        | 0.90(0.02)        |
|      | RF         | 87.2(0.02)        | 97.4(0.01)        | 0.97(0.01)        | 92.3(0.01)        | 0.92(0.01)        | 0.85(0.02)        | 0.92(0.01)        |
|      | AdaBoost   | 84.3(0.02)        | 85.7(0.01)        | 0.86(0.01)        | 85.0(0.00)        | 0.85(0.00)        | 0.70(0.00)        | 0.85(0.00)        |
|      | GB         | 87.5(0.02)        | 92.2(0.01)        | 0.92(0.01)        | 89.9(0.01)        | 0.88(0.02)        | 0.80(0.03)        | 0.90(0.01)        |
| pT   | RoF        | 79.4(0.02)        | <b>99.3(0.01)</b> | <b>0.99(0.01)</b> | 89.3(0.01)        | 0.88(0.01)        | 0.80(0.02)        | 0.89(0.01)        |
|      | SVM        | 84.3(0.02)        | 87.5(0.01)        | 0.87(0.01)        | 85.9(0.01)        | 0.86(0.01)        | 0.72(0.01)        | <b>0.93(0.01)</b> |
|      | RF         | 86.9(0.02)        | 97.0(0.01)        | 0.97(0.01)        | <b>91.9(0.01)</b> | <b>0.92(0.01)</b> | <b>0.84(0.02)</b> | 0.92(0.01)        |
|      | AdaBoost   | 84.3(0.02)        | 87.5(0.01)        | 0.87(0.01)        | 85.9(0.01)        | 0.86(0.01)        | 0.72(0.01)        | 0.86(0.01)        |
|      | GB         | <b>87.2(0.01)</b> | 91.3(0.02)        | 0.91(0.02)        | 89.2(0.02)        | 0.89(0.02)        | 0.76(0.04)        | 0.89(0.02)        |
| pY   | RoF        | 75.2(0.03)        | <b>99.3(0.00)</b> | <b>0.99(0.00)</b> | 87.3(0.02)        | 0.86(0.02)        | 0.77(0.03)        | 0.87(0.02)        |
|      | SVM        | <b>83.8(0.01)</b> | 95.7(0.01)        | 0.95(0.01)        | <b>89.7(0.01)</b> | <b>0.89(0.01)</b> | <b>0.80(0.02)</b> | <b>0.90(0.01)</b> |
|      | RF         | 83.4(0.01)        | 95.2(0.01)        | 0.95(0.01)        | 89.3(0.01)        | 0.87(0.01)        | 0.79(0.02)        | 0.89(0.01)        |
|      | AdaBoost   | 79.2(0.02)        | 80.1(0.01)        | 0.80(0.01)        | 79.6(0.01)        | 0.80(0.01)        | 0.59(0.02)        | 0.80(0.01)        |
|      | GB         | 83.1(0.03)        | 86.4(0.03)        | 0.86(0.03)        | 84.6(0.02)        | 0.84(0.02)        | 0.70(0.03)        | 0.85(0.02)        |

**Table 3.5:** Comparative results using different machine learning algorithm on 5 folds cross-validation

same metrics are calculated for our method on the independent test set for a fair comparison. The comparative results for the phospho-serine (pS) site are given in Table 3.7.

The results demonstrate that RotPhoPred outperforms both MPSite and NetPhosBac by achieving the highest sensitivity of 90.0%, specificity of 92.1%, accuracy of 91.0%, and MCC of 0.82. In the phospho-threonine (pT) site, our method also performs better than MPSite and NetPhosBac in terms of sensitivity, specificity, accuracy, and MCC to a large margin, as shown in Table 3.8. As shown in this table, we enhance the pT performance by 17.1%, 21.5%, 12.1%, and 0.52 in terms of sensitivity, specificity, accuracy, and MCC compared to MPSite as the current best predictor. Since no work has been done to predict the phospho-tyrosine(pY) site, there is no scope to compare the performance of the proposed predictor. Moreover, the independent test set results are consistent with the 5-fold cross-validation approach for both pS and pT classifiers, demonstrating our proposed method’s robustness and generality. It is important to note that the superiority of our proposed method comes from the integration of structural and evolutionary features and the use of the RoF classifier as it was discussed in prior sections.

We carefully analyzed the reason behind the poor performances of NetPhosBac and MPSite on the independent test set. We observed that the NetPhosBac was trained on only 152 positive sites, which might overfit or underfit the model. On the other hand, the benchmark training dataset of MPSite was imbalanced with five times higher negative samples than the positive sample for both pS and pT sites. MPSite did not

### 3.2 Results and Discussion

| Task | Classifier | Sn(%)       | Sp(%)       | Pr          | Ac(%)       | F1          | MCC         | auROC       |
|------|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| pS   | RoF        | <b>90.0</b> | 92.1        | 0.92        | <b>91.0</b> | <b>0.91</b> | <b>0.82</b> | <b>0.96</b> |
|      | SVM        | 79.4        | <b>99.4</b> | <b>0.99</b> | 89.4        | 0.88        | 0.80        | <b>0.96</b> |
|      | RF         | 88.8        | 92.4        | 0.92        | 90.6        | 0.90        | 0.81        | <b>0.96</b> |
|      | AdaBoost   | 87.6        | 76.8        | 0.79        | 82.2        | 0.83        | 0.65        | 0.90        |
|      | GB         | <b>90.6</b> | 79.7        | 0.82        | 85.1        | 0.86        | 0.71        | 0.93        |
| pT   | RoF        | 75.4        | <b>97.2</b> | <b>0.96</b> | <b>86.3</b> | <b>0.85</b> | <b>0.74</b> | <b>0.90</b> |
|      | SVM        | 82.9        | 86.5        | 0.86        | 84.7        | 0.84        | 0.69        | <b>0.91</b> |
|      | RF         | 82.5        | 84.1        | 0.84        | 83.3        | 0.83        | 0.67        | 0.90        |
|      | AdaBoost   | 79.8        | 72.2        | 0.74        | 76.0        | 0.77        | 0.52        | 0.82        |
|      | GB         | <b>83.7</b> | 78.6        | 0.80        | 81.2        | 0.82        | 0.62        | 0.89        |
| pY   | RoF        | 78.2        | <b>94.7</b> | <b>0.94</b> | <b>86.4</b> | <b>0.85</b> | <b>0.74</b> | <b>0.89</b> |
|      | SVM        | 85.6        | 73.2        | 0.76        | 79.4        | 0.81        | 0.59        | 0.89        |
|      | RF         | <b>87.3</b> | 70.8        | 0.75        | 79.0        | 0.81        | 0.59        | 0.89        |
|      | AdaBoost   | 83.8        | 65.5        | 0.71        | 74.6        | 0.77        | 0.50        | 0.78        |
|      | GB         | 86.6        | 66.9        | 0.72        | 76.8        | 0.79        | 0.55        | 0.87        |

**Table 3.6:** Comparative results using different machine learning algorithm on independent test

balance the training dataset like us using any imbalance treatment techniques, which may cause it to be biased toward the negative class. Besides, the test set of MPsite (the ratio of positive to negative sites is 1:5) was not highly imbalanced like ours (the ratio of positive to negative sites is 1: $\geq 15$ ). Maybe because of these reasons, NetPhosBac and MPsite show low MCC scores on our independent test but competitive scores on their test sets. However, though the models are trained on different training sets, we have evaluated all the predictors on the same independent test set; Hence the comparison is pretty fair. RotPhoPred as a standalone predictor and all its source codes are publicly available at : <https://github.com/faisalhm3d/RotPredPho>

| Predictors      | Sn(%)       | Sp(%)       | Ac(%)       | MCC         |
|-----------------|-------------|-------------|-------------|-------------|
| NetPhosBac [37] | 31.2        | 67.8        | 65.1        | -0.01       |
| MPSite [40]     | 39.1        | 77.9        | 75          | 0.11        |
| RotPhoPred      | <b>90.0</b> | <b>92.1</b> | <b>91.0</b> | <b>0.82</b> |

**Table 3.7:** Comparison of our method and current predictors for pS site identification on the independent test set.

| Predictors      | Sn(%)       | Sp(%)       | Ac(%)       | MCC         |
|-----------------|-------------|-------------|-------------|-------------|
| NetPhosBac [37] | 9.4         | 92.8        | 85.4        | 0.03        |
| MPSite [40]     | 58.3        | 75.7        | 74.2        | 0.22        |
| RotPhoPred      | <b>75.4</b> | <b>97.2</b> | <b>86.3</b> | <b>0.74</b> |

**Table 3.8:** Comparison of our method and current predictors for pT site identification on the independent test set

### 3.3 Summary

This chapter presents a new microbial phosphorylation site predictor, called RotPhoPred by integrating the structural information and evolutionary bigram profile. We also use Rotation Forest as our employed classifier, which to the best of our knowledge has never been used for this task, to build RotPhoPred. Experimental results on the independent test set demonstrate that RotPhoPred performs better than existing predictors found in the literature for both phospho-serin (pS) and phospho-threonine (pT). Such results indicate that the structural and evolutionary features provide significant discriminatory information to enhance the microbial phosphorylation site prediction task.

We also compared the performance of RoF with other state-of-the-art classifiers using same set of features. The results demonstrates the performance of using RoF over other classifiers for this task. In the future, we aim at using deep learning models to predict microbial phosphorylation sites more accurately. We also aim to develop a user-friendly and robust web server to provide data interpretation ability using graphical support. RotPhoPred as a standalone predictor and all its source codes are publicly available at : <https://github.com/faisalahr3d/RotPredPho>

## Chapter 4

# DeepPhoPred: A two-headed CNN model

In this chapter, we investigate how deep learning performs in predicting microbial phosphorylation sites. To do this, we design a CNN model incorporating a squeeze and excitation block that can handle structural and evolutionary data in parallel. Then, we perform experiments to select the optimal window size for sequence encoding and the best method for handling the data imbalance issue. We also analyze the contribution of the different sequence encoding schemes and the squeeze and excitation block. Finally, we visualize the features using t-distributed stochastic embedding and compare the model with state-of-the-art predictors reported in the literature.

### 4.1 Methodology

This section presents our proposed methodology to build DeepPhoPred, including encoding protein segments, balancing datasets, and designing and training the deep network.

#### 4.1.1 Encoding protein segment

Protein segments' structural and evolutionary properties are used as inputs for the proposed deep learning model since they exhibit promising results in other PTM sites prediction. Evolutionary property refers to the substitution probability of given amino

acid with other amino acids in a protein sequence. On the other hand, protein’s structural properties include secondary structures, accessible surface area (ASA), and torsion angles.

The evolutionary information of each protein sequence in our dataset is generated using the PSI-BLAST tool. PSI-BLAST was run for three iterations with a cutoff value of 0.001 on NCBI’s non-redundant (nr) database. It produces an  $L * 20$  dimensional position-specific scoring matrix (PSSM) for each sequence, where  $L$  signifies the protein’s length and 20 represents different amino acids which form the genetic code. To encode the evolutionary information for each positive or negative site described by the peptide  $P$  in our dataset, the affiliated submatrix  $M$  is pulled from the PSSM matrix of the protein sequence incorporating  $P$ .  $M$  is a  $W * 20$  shape matrix, where  $W$  is the window size, and 20 represents different amino acids.

Besides, each protein’s structural properties are predicted using the deep learning-based tool SPIDER3. It outputs an SPD file that holds a matrix of  $L * 8$  sizes that symbolises the predicted values of eight structural properties (coil, strand, helix, ASA,  $\psi$ ,  $\phi$ ,  $\theta$ ,  $\tau$ ) for each amino acid residue in a protein of length  $L$ . The submatrix  $N$  corresponds to the protein fragment  $p$  is segmented from the SPD file to encode the structural properties. Hence,  $N$  is a  $W * 8$  dimensional matrix, where  $W$  represents the window size, and 8 denotes the eight structural properties mentioned above.

The proposed deep learning model uses  $M$  and  $N$  matrices that capture evolutionary and structural information, respectively, to learn significant discriminatory features and predict phosphorylation sites

#### 4.1.2 Balancing the dataset

As discussed in the previous chapter, the benchmark datasets are imbalanced with more negative than positive samples. Earlier studies on PTM [68] [69] [70] show that the deep learning method performs better with more training data. Therefore, we have balanced the training dataset using the synthetic minority oversampling technique (SMOTE). SMOTE works as follows: it randomly selects a sample  $X_1$  from the minority class and discovers its  $k$  nearest minority instances based on some distance metric. The distance between  $X_1$  and another sample,  $X_{11}$ , which is arbitrarily chosen from the  $k$  instance, is then calculated. A synthetic instance  $r_1$  is then generated by adding the scaled distance value with  $X_1$ . The scaling of the distance value is performed by multiplying it with a

random number in  $(0,1]$ . The process is repeated until the dataset contains an equal number of positive and negative instances. The process is shown in Figure 4.1.

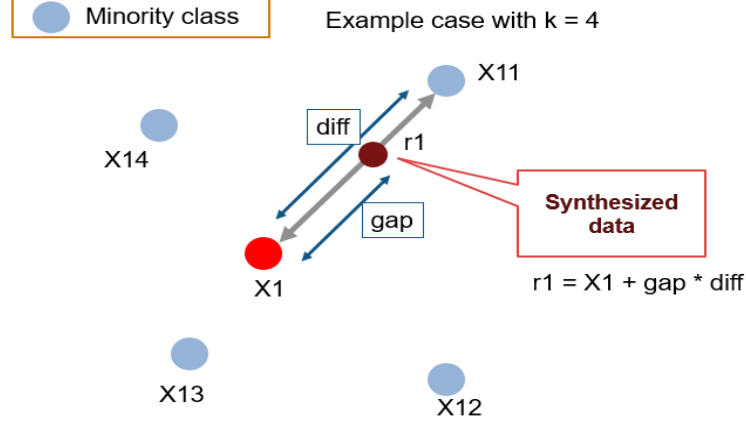


**Figure 4.1:** Working process of SMOTE

#### 4.1.3 Architecture of the proposed deep learning model

The proposed deep model's architecture is presented in Figure 4.2. Its input layer has two heads consisting of 21 (window size) \* 20 (number of amino acids) and 21 (window size) \* 8 (number of structural properties) dimensional matrices representing the evolutionary and structural information, respectively. Two parallel one-dimensional convolution layers are added next to these input channels, which can jointly learn and extract complex, ambiguous, discriminative features from the two information sources (structural and evolutionary profiles). Each convolution layer consists of 64 filters with kernel size 5, which is determined through cross-validation. Hence each convolution layer generates a feature map composed of 64 channels.

We do not utilise any pooling layer after the convolution layers for preserving spatial information and residue-related features extracted by the convolution layer. Moreover, our experiments show that it degrades the model's prediction performance. Our dataset is comparatively smaller, which may cause the overfitting problem. Hence, both l2 and dropout regularisation techniques are employed in the convolutional layers to alleviate this issue.

Next, each convolution layer has appended a squeeze and excitation (SE) block to introduce a channel attention mechanism by weighting each channel of the feature map

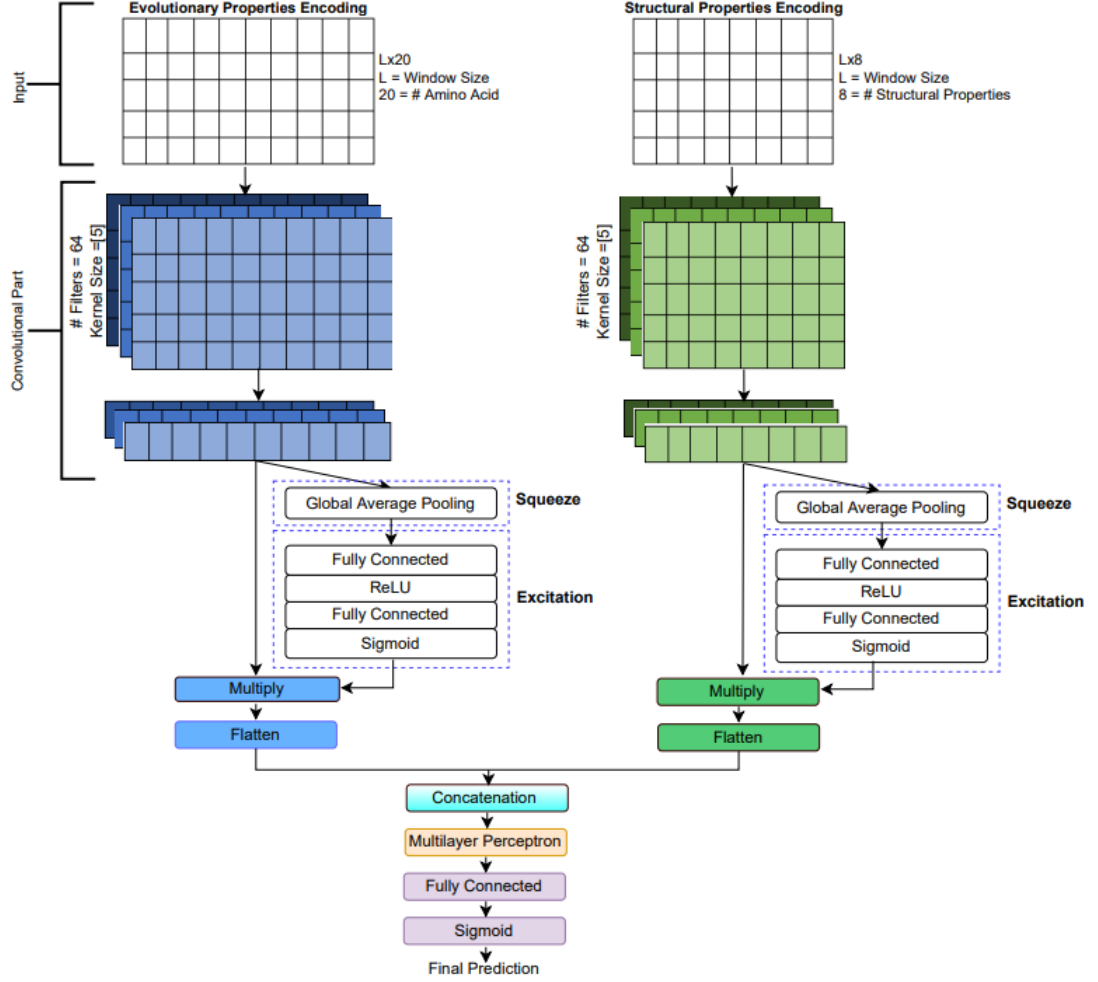


**Figure 4.2:** Architecture of the proposed deep learning model DeepPhoPred

adaptively based on their global information. The block consists of three modules: squeeze, excitation and scaling. The squeeze module gets a global understanding of each channel by employing a global average pooling layer, which reduces the spatial dimension of the whole channel to a singular value by taking the average of all values in that channel. The value indicates the relative importance of the channel. On the other hand, the excitation module learns the weights to scale each channel information of the feature map adaptively. Two fully connected layers operate in the excitation module to utilize the integrated information of the squeeze module and thoroughly capture channel-wise reliances [71]. The first layer with a relu activation function reduces the

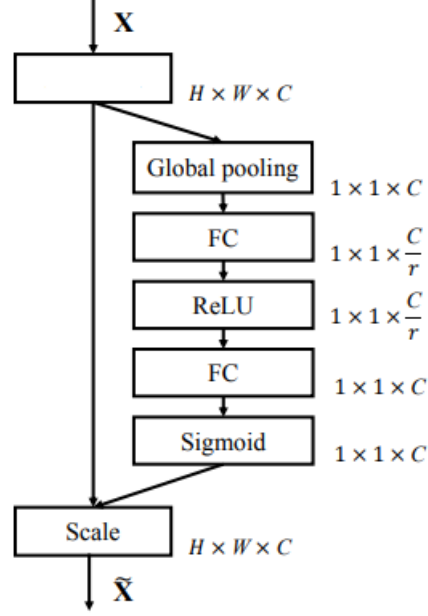


**Figure 4.3:** Overall operation of squeeze and excitation block

dimensionality of the input tensor by a factor of  $r$  (which is set at 16 in this study). This reduction is made by utilizing  $C/r$  neurons in this layer, where  $C$  is the number of squeezed block outputs. Since the number of neurons decreases, so do the number of training parameters (weight and bias) and, consequently, the computational complexity. The second fully connected layer restores the tensor dimension to the feature map's shape using the  $C$  number of neurons. This layer is followed by the sigmoid activation function, which converts the input tensor's values between 0 to 1. The overall operation of the squeeze and excitation block with the tensor dimension is shown in Figure 4.3.

Finally, the scaling module executes element-wise multiplication between the feature map generated by the convolution layer and the output tensor of the sigmoid function to scale the feature map. Thus the squeeze and excitation block reduces the non-relevant channel information without affecting the effective channels.

The scaled feature maps returned by the squeeze and excitation blocks are flattened into two one-dimensional vectors and later concatenated to form a feature vector. The resulting feature vector is then passed through a multi-layer perceptron consists of several densely connected neural network layers so that the model can successfully learn the importance of each feature and capture the complex relationship between the

| Hyperparameter | Hyperparameter Space          | Optimal Value |
|----------------|-------------------------------|---------------|
| Learning rate  | [0.1, 0.001, 0.0001, 0.00001] | 0.001         |
| Optimizer      | ['sgd', 'adam', 'RMSprop']    | 'adam'        |
| Batch size     | [16, 32, 64, 128]             | 64            |

**Table 4.1:** Summary of the hyperparameters' space and their optimal values

features in predicting the phosphorylation sites. The relu activation function is applied to each dense layer's neuron to draw a non-linear decision boundary in the complex feature space for separating phosphorylation sites from non-phosphorylated sites. The relu is chosen over other activation functions because of its simplicity, popularity and effectiveness. The dropout layer is added next to each dense layer to prevent the model from overfitting. The output layer of the network constitutes a dense layer of a single neuron with the sigmoid activation function for binary classification, which returns the probability of the site being phosphorylated.

#### 4.1.4 Model training

The binary cross-entropy loss function is used as the objective function, which measures the uncertainty associated with the predicted probability distribution as follows:

$$L_c = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (4.1)$$

where,  $y$  is the target value of the network, which is 1 for the phosphorylated site and 0 for the non-phosphorylated site,  $\hat{y}$  is the predicted probability that the site is positive, and  $N$  is the number of training instances. The model's parameters are trained using the Adam optimizer, which computes adaptive learning rates for each parameter based on the estimated first and second moments of the gradients, with an initial learning rate of 0.0001 and an exponential decay rate for the first and second moment estimates of  $\beta_1 = 0.5$ ,  $\beta_2 = 0.999$  and batch size of 64 samples. An early stopping strategy was used to reduce the training time. We used the model checkpoint function to choose the best model from all the epochs based on the highest accuracy and lowest loss on the validation dataset. All the hyperparameters are tuned through the cross-validation approach. A summary of the hyperparameters' space and their optimal values are presented in Table 4.1

## 4.2 Results and Discussion

This section presents our results, compares them with previous studies, and discusses their significance.

### 4.2.1 Contribution analysis of Squeeze and Excitation (SE) block

While generating the feature map, the convolution layer gives equal priority to each filter which might not be reasonable. Some channels in the feature map might be more critical. Therefore, in this study, we employed a squeeze and excitation (SE) block that introduced a channel attention mechanism and scaled the spatial information of different channels in the feature map based on their relative importance. This section analyses the SE block’s contribution to improving the proposed model’s prediction performance. To do this, we trained and evaluated the same model after removing the SE block only. The evaluation results reported in Tables B.1 and 4.2 shows that the model performance decreased in cross-validation and independent test when the SE block was removed. As shown in Table 4.2, the MCC scores of the model with the SE block are 0.18, 0.25 and 0.16 higher than those without the SE block for phosphorylation detecting in serine, threonine and tyrosine residues, respectively, on the independent test set. The model’s precision also decreased significantly from 0.88, 0.90 and 0.93 to 0.60, 0.53 and 0.68 for pS, pT and pY sites prediction. These results indicate that the SE block greatly minimizes the false positive rate and enhances the model’s prediction performance.

| Task | Model            | Ac (%) | MCC  | Pr   | auROC | F1   | Sn (%) | Sp (%) | auPR |
|------|------------------|--------|------|------|-------|------|--------|--------|------|
| pS   | With SE block    | 99.0   | 0.93 | 0.88 | 0.99  | 0.94 | 100    | 98.9   | 1.00 |
|      | Without SE block | 95.1   | 0.75 | 0.60 | 0.97  | 0.75 | 99.4   | 94.7   | 0.98 |
| pT   | With SE block    | 99.0   | 0.94 | 0.90 | 0.99  | 0.95 | 99.2   | 99.0   | 0.99 |
|      | Without SE block | 91.9   | 0.69 | 0.53 | 0.95  | 0.69 | 99.6   | 91.2   | 0.99 |
| pY   | With SE block    | 98.7   | 0.95 | 0.93 | 0.99  | 0.96 | 99.3   | 98.7   | 1.00 |
|      | Without SE block | 93.2   | 0.79 | 0.68 | 0.96  | 0.81 | 100    | 92.0   | 0.99 |

**Table 4.2:** Performance of DeepPhoPred with and without SE block on independent test

### 4.2.2 Contribution analysis for different sequence representations

In this section, we have examined the significance of different sequence representation techniques in phosphorylation prediction. We have trained two individual CNN models,

## 4.2 Results and Discussion

| Task | Encoding     | Ac (%) | MCC  | Pr   | auROC | F1   | Sn (%) | Sp(%) | auPR |
|------|--------------|--------|------|------|-------|------|--------|-------|------|
| pS   | Combined     | 99.5   | 0.99 | 0.99 | 0.99  | 1.00 | 100    | 99.0  | 1.00 |
|      | Structural   | 91.8   | 0.84 | 0.86 | 0.92  | 0.92 | 99.2   | 84.2  | 0.98 |
|      | Evolutionary | 92.5   | 0.86 | 0.88 | 0.92  | 0.93 | 98.6   | 86.4  | 0.99 |
| pT   | Combined     | 99.4   | 0.99 | 0.99 | 0.99  | 0.99 | 99.7   | 99.1  | 1.00 |
|      | Structural   | 91.1   | 0.83 | 0.86 | 0.91  | 0.92 | 99.0   | 83.3  | 0.97 |
|      | Evolutionary | 93.5   | 0.88 | 0.89 | 0.94  | 0.94 | 99.6   | 87.7  | 1.00 |
| pY   | Combined     | 98.8   | 0.98 | 0.98 | 0.99  | 0.99 | 99.5   | 98.1  | 1.00 |
|      | Structural   | 89.4   | 0.80 | 0.84 | 0.89  | 0.90 | 98.1   | 80.6  | 0.97 |
|      | Evolutionary | 89.5   | 0.80 | 0.84 | 0.89  | 0.90 | 97.9   | 81.0  | 0.98 |

**Table 4.3:** Comparative results among different sequence representations on 5-folds cross-validation

one with the structural and another with the evolutionary information and compared them with the proposed model (DeepPhoPred) that integrates both encoding schemes for this experiment. Structural and Evolutionary models are designed just by pruning the input head from the DeepPhoPred that captures the evolutionary and structural information, respectively. Other layers of the models were kept unchanged for a fair comparison. The comparative results on 5 folds cross-validation are presented in Table 4.3. The results show that the combined model outperforms the structural and evolutionary models on cross-validation for each measure in predicting three phosphorylation sites. Significant improvements are observed in accuracy, precision, specificity, MCC and F1 scores.

For pS prediction, the combined model achieves accuracy, specificity, precision, MCC and F1 values of 99.5%, 99%, 0.99, 0.99 and 1.00, respectively, which are 7.7%, 14.85%, 0.13, 0.15 and 0.8 higher than the structural model and 7%, 12.6%, 0.11, 0.13 and 0.7 higher than evolutionary model.

For pT, the combined model improves accuracy, specificity, precision, MCC, and F1 score by 8.3%, 15.8%, 0.13, 0.16 and 0.7 comparing the structural model and 5.9%, 11.4%, 0.10, 0.11 and 0.05 comparing the evolutionary model.

For pY, the combined model outperforms the structural model by 9.4%, 17.5%, 0.14, 0.18, and 0.9 in terms of accuracy, specificity, precision, MCC, and F1 score, while they are 9.3%, 17.1%, 0.14, 0.18, and 0.09 for the evolutionary model.

All the results demonstrate that both sequence encoding schemes significantly contribute to the model performance by reducing the false positive rate and increasing its ability to detect nonphosphorylation sites accurately.

The ROC and PR curves of these models on test data are plotted in Figure 4.4. The curves also show that the combined model surpasses the individual models since it attains higher AUC values for pS, pT and pY prediction.

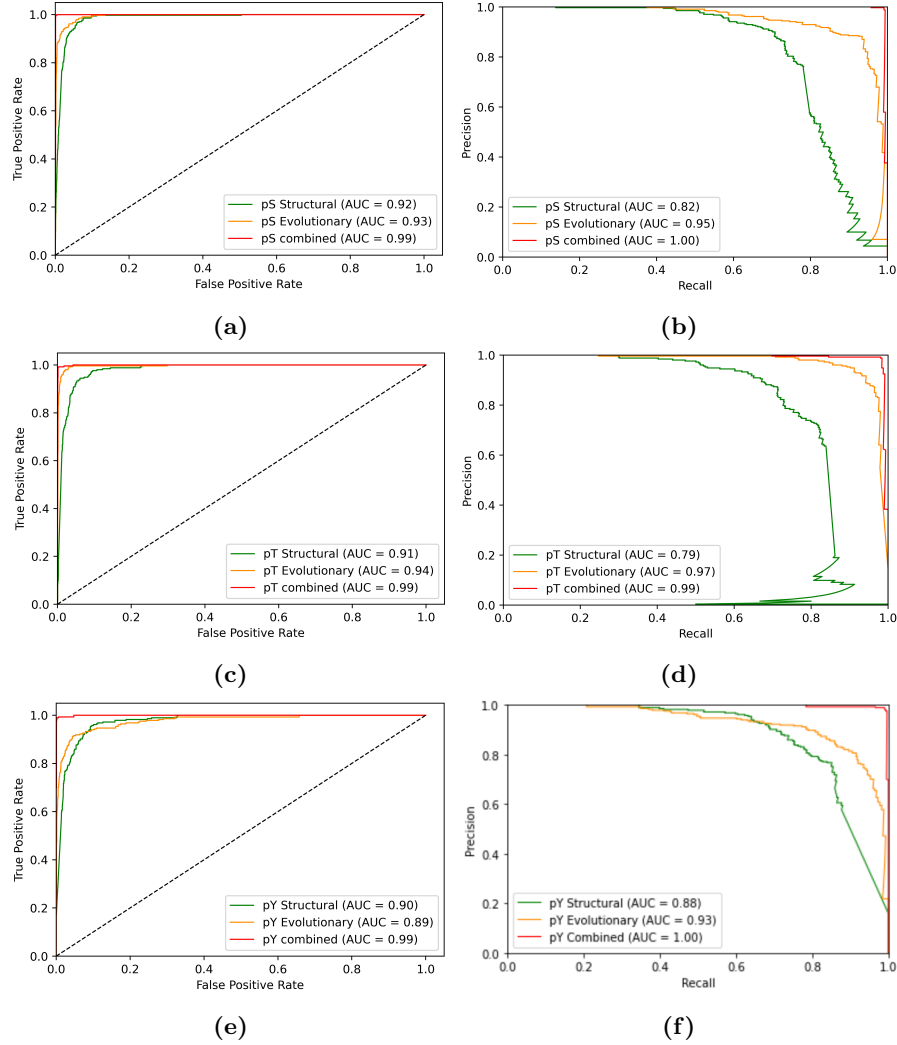


**Figure 4.4:** Comparative results for different encoding techniques. (a)ROC curves for pS site, (b)PR curves for pS site, (c)ROC curves for pT site (d) PR curves for pT site, (e) ROC curves for pY site and (f) PR curves for pY site

### 4.2.3 Comparison with traditional machine learning-based classifiers

In this section, we have compared the results attained by the proposed CNN model with the traditional machine learning models. The comparative results are presented

in Table 4.4. In the table, CNN shows superior performance over the ML models in all performance metrics in all classification tasks, which suggests the efficacy of the proposed CNN architecture. The sensitivity and specificity of the CNN models are not only higher than the ML models but also consistent. Consequently, the CNN model yields significantly higher MCC scores (0.93, 0.94 and 0.95) in predicting pS, pT and pY sites than ML models. In the case of the traditional ML models, we observe a notable imbalance between the sensitivity and specificity scores for all classification tasks, which indicates class bias and results in comparatively lower MCC scores. The ROC and PR curves for all classification task on the independent test is shown in Figure 4.5. The figure shows that the CNN model covers a substantially larger region in the PR and ROC curves than conventional machine learning classifiers. It achieves about 25% and 50% higher values in ROC and PR curves than the top ML models for predicting all types of phosphorylation sites. Specifically, it suppresses the best performing ml-classifier (Random Forest) for pS by achieving 0.26 and 0.46 higher areas in ROC and PR curves. It beats the Rotation Forest, which produces the highest AUC values in both ROC and PR curves for detecting pT sites among all ML models by 0.27 and 0.45. Finally, for pY sites, auROC and auPR are 0.32 and 0.46, respectively, higher than Rotation forest, which is the best machine learning-based predictor. Adaboosts shows the worst performance in all cases. Such results suggest that the traditional ML-based classifiers with the row structural and evolutionary information are insufficient to predict microbial phosphorylation sites accurately. Since the CNN model, with its optimal architecture, possess the ability to learn significant features from two information sources, it achieves outstanding performance in all prediction tasks.

#### 4.2.4 Comparing data balancing techniques

To compare the performance of under-sampling and oversampling techniques for data balancing, we have trained the DeepPhoPred using NearMiss and SMOTE methods.

Table 4.5 presents the comparative results on the independent test set. The SMOTE shows remarkably higher results in the table than NearMiss in all the performance measure metrics. On the other hand, NearMiss exhibits poor MCC values of 0.64, 0.64 and 0.72 for pS, pT and pY sites. It fails to detect 8.2%, 10.3%, and 12.7% phosphorylated serine, threonine, and tyrosine sites, respectively. The results are justified since deep learning needs a large dataset to train the model.

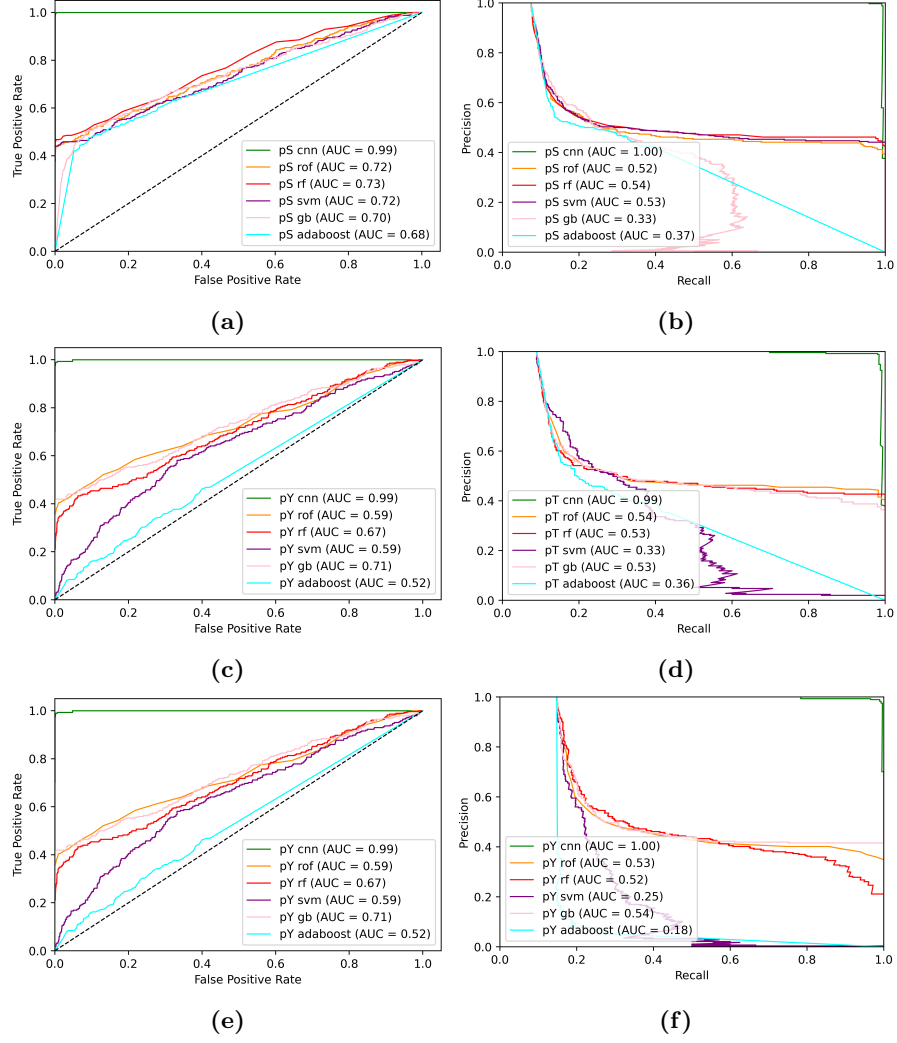


**Figure 4.5:** Comparative results for different classifiers. (a) ROC curves for pS site, (b) PR curves for pS site, (c) ROC curves for pT site (d) PR curves for pT site, (e) ROC curves for pY site and (f) PR curves for pY site

## 4.2 Results and Discussion

| Task | Classifier | Ac (%) | MCC  | Pr   | auROC | F1   | Sn (%) | Sp (%) | auPR |
|------|------------|--------|------|------|-------|------|--------|--------|------|
| pS   | CNN        | 99.0   | 0.93 | 0.88 | 0.99  | 0.94 | 100    | 98.9   | 1.00 |
|      | RoF        | 95.9   | 0.66 | 0.99 | 0.72  | 0.63 | 45.9   | 100    | 0.52 |
|      | RF         | 94.4   | 0.53 | 0.69 | 0.73  | 0.54 | 44.7   | 98.4   | 0.54 |
|      | SVM        | 95.8   | 0.65 | 1.00 | 0.72  | 0.61 | 44.1   | 100    | 0.53 |
|      | GB         | 84.3   | 0.29 | 0.25 | 0.70  | 0.34 | 53.2   | 86.9   | 0.33 |
|      | AdaBoost   | 80.8   | 0.24 | 0.20 | 0.68  | 0.29 | 52.4   | 83.1   | 0.37 |
| pT   | CNN        | 99.0   | 0.94 | 0.90 | 0.99  | 0.95 | 99.2   | 99.0   | 0.99 |
|      | RoF        | 94.9   | 0.64 | 0.98 | 0.72  | 0.61 | 43.9   | 99.9   | 0.54 |
|      | RF         | 94.8   | 0.63 | 0.99 | 0.71  | 0.60 | 42.7   | 100    | 0.53 |
|      | SVM        | 81.3   | 0.30 | 0.26 | 0.72  | 0.36 | 59.7   | 83.4   | 0.33 |
|      | GB         | 91.8   | 0.45 | 0.55 | 0.70  | 0.49 | 43.9   | 96.5   | 0.53 |
|      | AdaBoost   | 80.9   | 0.26 | 0.24 | 0.68  | 0.33 | 53.4   | 83.6   | 0.36 |
| pY   | CNN        | 98.7   | 0.95 | 0.93 | 0.99  | 0.96 | 99.3   | 98.7   | 1.00 |
|      | RoF        | 91.6   | 0.62 | 0.98 | 0.72  | 0.61 | 44.0   | 99.8   | 0.53 |
|      | RF         | 86.5   | 0.43 | 0.55 | 0.70  | 0.50 | 46.5   | 93.5   | 0.52 |
|      | SVM        | 76.8   | 0.31 | 0.34 | 0.69  | 0.42 | 57.7   | 80.1   | 0.25 |
|      | GB         | 91.3   | 0.61 | 1.00 | 0.71  | 0.59 | 41.5   | 100    | 0.54 |
|      | AdaBoost   | 73.9   | 0.23 | 0.29 | 0.64  | 0.37 | 50.7   | 77.9   | 0.18 |

**Table 4.4:** The result achieved for CNN compared to traditional ML models on independent test data (using direct pssm and spd)

| Task | Balancing Method | Sn (%) | Sp(%) | Pr   | Ac (%) | F1   | MCC  | auROC | auPR |
|------|------------------|--------|-------|------|--------|------|------|-------|------|
| pS   | SMOTE            | 100    | 98.9  | 0.88 | 99.0   | 0.94 | 0.93 | 0.99  | 1.00 |
|      | NearMiss         | 91.8   | 92.5  | 0.5  | 92.4   | 0.64 | 0.64 | 0.92  | 0.69 |
| pT   | SMOTE            | 99.2   | 99    | 0.90 | 99.0   | 0.95 | 0.94 | 0.99  | 0.99 |
|      | NearMiss         | 89.7   | 91.6  | 0.51 | 91.4   | 0.65 | 0.64 | 0.91  | 0.65 |
| pY   | SMOTE            | 99.3   | 98.7  | 0.93 | 98.7   | 0.96 | 0.95 | 0.99  | 1.00 |
|      | NearMiss         | 87.3   | 92.9  | 0.68 | 91.5   | 0.77 | 0.72 | 0.90  | 0.78 |

**Table 4.5:** Comparative results for data balancing methods on the independent test set

The ROC and PR curves for these two methods for predicting S, T, and Y phosphorylation sites are depicted in Figure 4.6. The figure shows that SMOTE achieves the perfect area under the ROC and PR curves for detecting three types of phosphorylation sites. It produces 0.07, 0.08 and 0.09 higher auROC values and 0.31, 0.34 and 0.22 higher auPR than the NearMiss for pS, pT and pY, respectively.

### 4.2.5 Selection of optimal window size

In this section, we have performed experiments to determine the optimal window size for feature extraction based on the MCC score. The MCC scores for different window lengths (11, 21, 31, 41 and 51) on 5 folds cross validation are graphically visualized on the bar plots in Figure 4.7 for pS, pT and pY sites prediction. The figure reveals

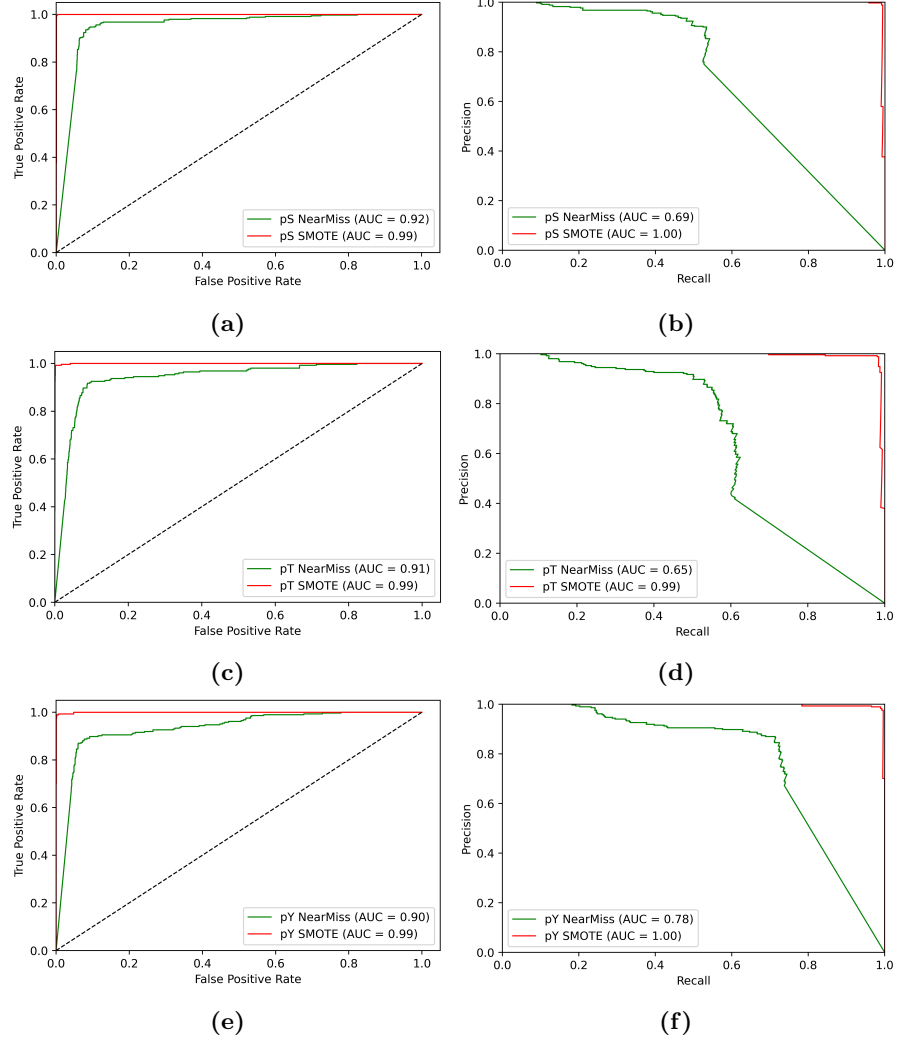


**Figure 4.6:** Comparative results for different data balancing technique. (a) ROC curves for pS site, (b) PR curves for pS site, (c) ROC curves for pT site (d) PR curves for pT site, (e) ROC curves for pY site and (f) PR curves for pY site

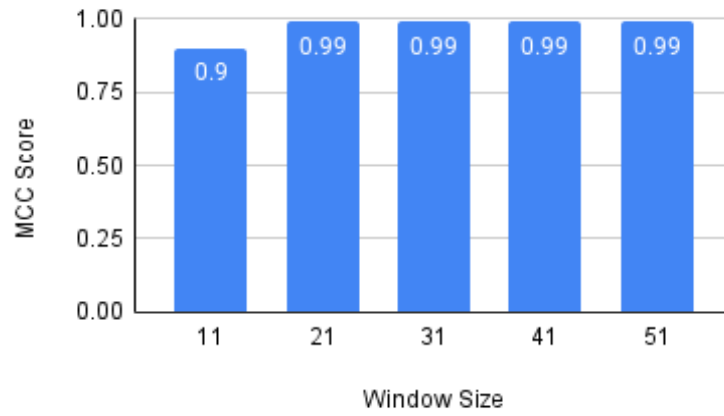
that window size 21 is optimal for predicting three types of phosphorylation sites (pS, pT and pY) with the highest MCC scores of 0.99, 0.99 and 0.98, respectively. With a window size of more than 21, the MCC score remains constant, whereas the learnable parameter increases, thus the model complexity. On the other hand, the necessary sequence information may be lost when the window size is less than 21, which causes the model’s performance degradation.

#### 4.2.6 Visualization of the features

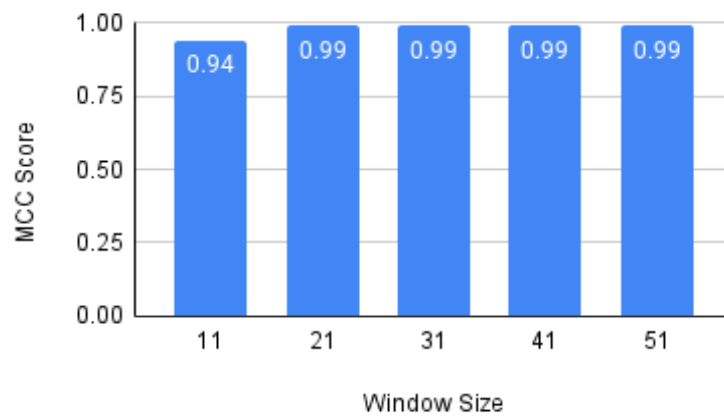
We have used the t-distributed stochastic neighbour embedding (t-SNE) to visualize the discriminating ability between the positive and negative samples by the row input features and learned features by our deep learning model. The scatter plots of two-dimensional components (tsne.1 and tsne.2) achieved by t-SNE for predicting pS, pT and pY are shown in Figure 4.8. The plots show that the row structural and evolutionary features distribution is distorted, chaotic and non-separable between the positive and negative samples. After passing through the convolution and multiple dense layers, the row input features are transformed into a representation such that there is a transparent partition between the distribution of the positive and negative sites. These results indicate that our model can effectively learn distinguishing features for separating the phosphorylated site from non-phosphorylated ones.

#### 4.2.7 Comparison with existing predictors

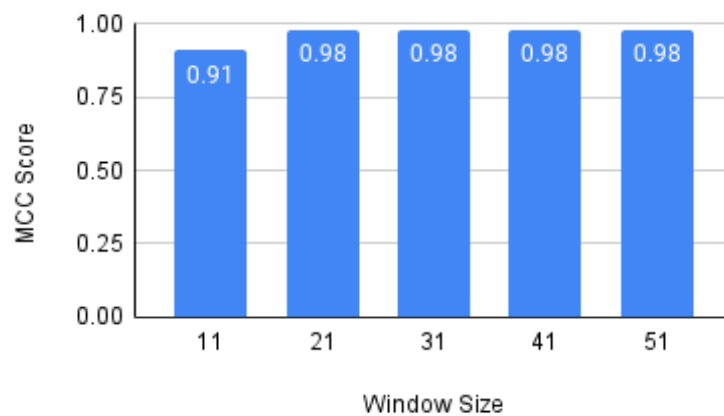
This section compares DeepPhosPred with RotPhosPred and publicly available existing predictors-MPsite and NetPhosBac. The comparative results for pS, pT and pY sites prediction on the independent test set in terms of Sn, Sp, Ac and MCC are presented in Tables 4.6-4.8. The Tables show that DeepPhosPred outperforms MPsite, NetPhosBac and RotPhosPred in all tasks, with significantly higher scores in all metrics. DeepPhosPred enhances Sn, Sp, Ac and MCC up to (10%, 6.8%, 8%, and 0.11), (60.9%,21%,24% and 0.82) and (68.8%,31.1%,33.9%,and 0.94) over RotPhosPred, MP-Site and NetPhosBack, respectively, for predicting pS. From Table 4.7, we can observe that DeepPhosPred improves Sn, Sp Ac and MCC up to (23.8%, 1.8%, 12.7%, and 0.20), (40.9%,23.3%,24.8% and 0.72) and (89.8%,6.2%,13.6% and 0.91) over RotPhos-Pred, MPSite and NetPhosBack, respectively, for predicting pT. Table 4.8 shows that DeepPhosPred archives up to 21.1%, 4%, 12.3%, and 0.21 improvements in Sn, Sp,



(a)



(b)



(c)

**Figure 4.7:** MCC scores for different window sizes for predicting phosphorylation on (a) Serine, (b) Threonine and (c) Tyrosine residues.

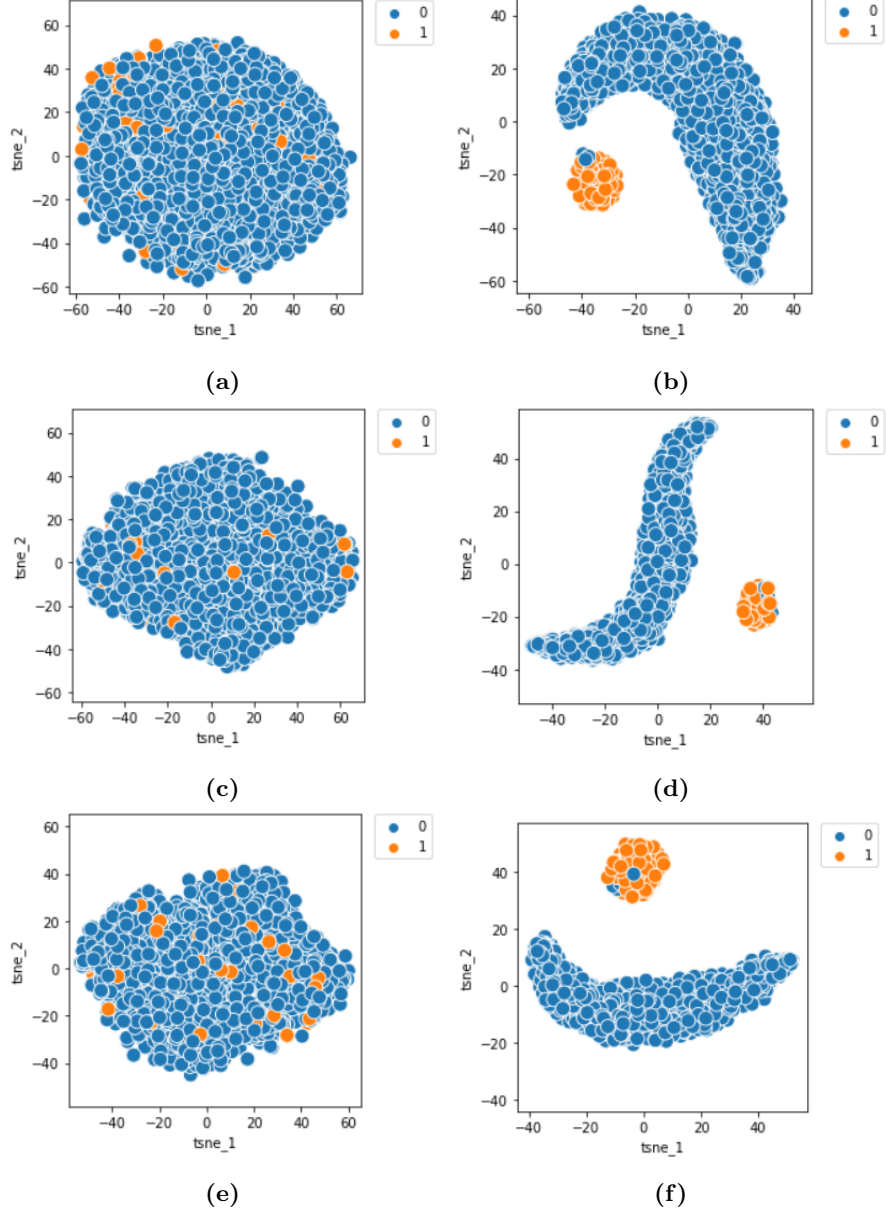


**Figure 4.8:** Two-dimensional projection of row input and features learned by DeepPho-Pred (left to right) using t-SNE for pS, pT and pT tasks (top to bottom); blue and orange points are negative and positive samples respectively.

| Predictors      | Sn(%)       | Sp(%)       | Ac(%)       | MCC         |
|-----------------|-------------|-------------|-------------|-------------|
| NetPhosBac [37] | 31.2        | 67.8        | 65.1        | -0.01       |
| MPSite [40]     | 39.1        | 77.9        | 75          | 0.11        |
| RotPhoPred      | <b>90.0</b> | <b>92.1</b> | <b>91.0</b> | <b>0.82</b> |
| DeepPhoPred     | <b>100</b>  | <b>98.9</b> | <b>99.0</b> | <b>0.93</b> |

**Table 4.6:** Comparison of our methods and current predictors for pS site identification on the independent test set.

| Predictors      | Sn(%)       | Sp(%)       | Ac(%)       | MCC         |
|-----------------|-------------|-------------|-------------|-------------|
| NetPhosBac [37] | 9.4         | 92.8        | 85.4        | 0.03        |
| MPSite [40]     | 58.3        | 75.7        | 74.2        | 0.22        |
| RotPhoPred      | <b>75.4</b> | <b>97.2</b> | <b>86.3</b> | <b>0.74</b> |
| DeepPhoPred     | <b>99.2</b> | <b>99.0</b> | <b>99.0</b> | <b>0.94</b> |

**Table 4.7:** Comparison of our methods and current predictors for pT site identification on the independent test set

Ac and MCC over RotPhosPred, respectively, in pY site prediction. RotPhosPred is the second best predictor and performs much better than MPsite and NetPhosBac in terms of all metrics in both pS and pY sites prediction. The results demonstrate that our proposed predictors can identify phosphorylated and non-phosphorylated sites with minimum error. In the case of DeepPhosPred, only 0%, 0.8%, 0.7% phosphoserine(pS), phospho-threonine and phospho-tyrosine residues remain undetected while RotPhosPred fails to detect only 10%, 24.6% and 21.8% pS, pT and pY sites respectively. However, MPsite and NetPhosBac show much higher specificity than sensitivity, revealing the models are biased toward the negative class. As a result, they can not detect a remarkably higher number of phosphorylated sites. The results suggest that our proposed models can complement the experimental method by helping to initially screen phosphorylation sites.

## 4.3 Summary

In this chapter, we have proposed a two-headed CNN model with squeeze and excitation block that can simultaneously process evolutionary and structural information and accurately predict microbial pS, pT and pY sites. We have performed an extensive ex-

---

| Predictors  | Sn(%)       | Sp(%)       | Ac(%)       | MCC         |
|-------------|-------------|-------------|-------------|-------------|
| RotPhoPred  | 78.2        | 94.7        | 86.4        | 0.74        |
| DeepPhoPred | <b>99.3</b> | <b>98.7</b> | <b>98.7</b> | <b>0.95</b> |

---

**Table 4.8:** Comparison of our predictors for pY site identification on the independent test set.

periment to justify the effectiveness of DeepPhoPred. We compared DeepPhoPred with RotPhoPred proposed in the previous chapter and with two state-of-the-art predictors, MPsite and NetPhosBac.

The reported results show that DeepPhoPred outperforms all these tools by achieving significantly higher values in all evaluation metrics for pS, pT and pY site prediction. DeepPhoPred shows such an outstanding performance because it can learn features from both structural and evolutionary information using a highly efficient deep learning architecture.

However, we have evaluated the performance of DeepPhoPred on a single dataset. It is vital to examine the performance on more datasets to justify the model’s scalability. Moreover, how accurately the model can identify phosphorylation on higher-order organisms is yet to explore. In future, we aim to focus on these issues.

## Chapter 5

# Summary, Conclusions, and Future Work

### 5.1 Summary

Phosphorylation is a substantial post-translational modification that refers to adding a phosphate group to the amino acid side chain after translation operation in the ribosome. It is vital for coordinating cellular and organic functions such as regulating metabolism, proliferation, apoptosis, subcellular trafficking, inflammation, and other crucial physiological processes. Phosphorylation prediction in a microbial organism can assist in understanding pathogenesis and host-pathogen interaction, drug, and antibody design, and antimicrobial agent development. Experimental methods for phosphorylation detection are expensive, time-consuming, and tedious. Hence there is an urgent necessity for computational approaches.

This thesis proposes two computational tools for predicting microbial pS, pT, and pY sites, exploiting protein’s evolutionary and structural information. The first tool, RotPhoPred, is based on machine learning which extracts structural features and evolutionary bigram profiles from peptide fragments and amalgamates them to train a Rotation Forest for phosphorylation site prediction. The data imbalance issue is tackled in this predictor using the Nearmiss undersampling technique.

In the second part of the thesis, we design DeepPhoPred, a two-headed CNN architecture incorporating squeeze and excitation blocks that can simultaneously learn intrinsic and substantial features from two information sources to detect phosphoryla-

tion sites. SMOTE oversampling method is utilized in this predictor for balancing the training data.

Comprehensive experimental results demonstrate that both tools can suppress the state-of-the-art predictors with significantly high values in all evaluation metrics. RotPhoPred achieves sensitivity, specificity, accuracy, and MCC of (90.0%, 92.1%, 91.0% and 0.82), (75.4%, 97.2%, 86.3% and 0.74) and (78.2%, 94.7%, 86.4% and 0.74) for predicting phosphorylation on S, T and Y sites respectively. DeepPhoPred performs better than RotPhoPred, attaining sensitivity, specificity, accuracy, and MCC of (100.0%, 98.9%, 99% and 0.93), (99.2%, 99% and 99% and 0.94) and (99.3%, 98.7% and 98.7% and 0.95) respectively for predicting pS, pT and pY sites. These results support our starting hypothesis: fusion of evolutionary and structural features and application of deep learning techniques can improve microbial phosphorylation site prediction accuracy.

## 5.2 Conclusions

From this thesis, we can draw the following conclusion:

- The optimal window size for feature extraction or a sequence encoding for each site is 21 residues, comprising 10 upstream and 10 downstream, with S/T/Y at the central position for both predictors. When the window size is minimized, the model's performance degrades due to underfitting caused by fewer features. On the other hand, model performance remains unchanged with a window size of more than 21, but its computation complexity increased due to increased features/parameters.
- For data balancing, downsampling and upsampling techniques show the best results for RotPhoPred and DeepPhoPred, respectively. The finding is justified since deep learning improves generalization with more data.
- Prediction performance improves significantly when evolutionary and structural features are combined than when either structural or evolutionary features are used. This finding is consistent for both RotPhoPred and DeepPhoPred predicting pS, pT, and pY sites. Structural features show comparatively better results

than evolutionary features in the case of machine learning, while the reverse is true for deep learning.

- With the same set of features, Rotation Forest constantly outperforms the state-of-the-art machine learning algorithms for microbial phosphorylation prediction.
- The squeeze and excitation blocks incorporated in CNN architecture help to minimize the false positive rate and balance the sensitivity and specificity.

### 5.3 Future Work

This thesis opens new bases for future research. Some of them are enlisted below:

- We aim to develop a user-friendly and robust web server to provide data interpretation ability using graphical support.
- We have evaluated the tools' performance on a single dataset. It is vital to examine the performance on more datasets to justify the model's scalability. Moreover, how accurately the models can identify phosphorylation on higher-order organisms is yet to explore. In the future, we aim to focus on these issues too.
- We plan to adopt popular word embedding strategies such as Word2Vec, Glove, and FastText for NLP tasks to encode peptides for preserving local sequence order information and input them into the CNN model for phosphorylation site prediction.
- Furthermore, many pre-trained protein sequence models have been made accessible to the public recently, motivated by the success of self-supervised pre-training on NLP tasks. The LSTM sequence model UDSMProt [72] learned from unlabeled Swiss-Prot protein sequences exploiting self-supervised autoregressive methods demonstrated remarkable results in classifying protein levels after some tweaking. Similarly, a convolutional transformer and attention-based model ProteinBERT [73] pre-trained on sequence correction and GO annotation prediction task showed outstanding performance on protein level regression task. In future work, we aim to study the prospect of fine-tuning these pre-trained models to integrate with DeepPhoPred for predicting phosphorylation sites.

- The deepPhoPred model lacks interpretability due to the black-box nature of CNN. Introducing good interpretability to explain how the model decides on the phosphorylation sites will be investigated in future research.
- In current machine learning and deep learning, data is stored in a central server and processed in an unprotected manner, posing challenges to preserving user privacy, specifically in healthcare and genomics, where data is sensitive. It is straightforward to identify someone using his/her genomic information. Therefore, the organization is reluctant to share user data because of privacy concerns and liability risks, hindering genomic research. We aim to introduce federated learning to develop a privacy-preserving framework that allows multi-institutional collaboration without sharing user data to advance genomic research.

# Bibliography

- [1] M. M. Rashid, S. Shatabda, M. Hasan, H. Kurata *et al.*, “Recent development of machine learning methods in microbial phosphorylation sites,” *Current Genomics*, vol. 21, no. 3, pp. 194–203, 2020. 1
- [2] S. Jamal, W. Ali, P. Nagpal, A. Grover, and S. Grover, “Predicting phosphorylation sites using machine learning by integrating the sequence, structure, and functional information of proteins,” *Journal of translational medicine*, vol. 19, no. 1, pp. 1–11, 2021. 1, 6
- [3] B. Trost and A. Kusalik, “Computational prediction of eukaryotic phosphorylation sites,” *Bioinformatics*, vol. 27, no. 21, pp. 2927–2935, 2011. 1, 6
- [4] C.-W. Chen, L.-Y. Huang, C.-F. Liao, K.-P. Chang, and Y.-W. Chu, “Gasphos: protein phosphorylation site prediction using a new feature selection approach with a ga-aided ant colony system,” *International journal of molecular sciences*, vol. 21, no. 21, p. 7891, 2020. 1, 6
- [5] Y. A. Chen and S. A. Eschrich, “Computational methods and opportunities for phosphorylation network medicine,” *Translational cancer research*, vol. 3, no. 3, p. 266, 2014. 1, 6
- [6] J. Loughery and D. Meek, “Switching on p53: an essential role for protein phosphorylation?” *BioDiscovery*, vol. 8, p. e8946, 2013. 1, 6
- [7] Y. Shi, Y. Zhang, S. Lin, C. Wang, J. Zhou, D. Peng, and Y. Xue, “dbpsp 2.0, an updated database of protein phosphorylation sites in prokaryotes,” *Scientific Data*, vol. 7, no. 1, pp. 1–9, 2020. 1, 6, 16

- [8] B. Yu, Z. Yu, C. Chen, A. Ma, B. Liu, B. Tian, and Q. Ma, “Dnnace: prediction of prokaryote lysine acetylation sites through deep neural networks with multi-information fusion,” *Chemometrics and intelligent laboratory systems*, vol. 200, p. 103999, 2020. 2
- [9] S. Li, H. Li, M. Li, Y. Shyr, L. Xie, and Y. Li, “Improved prediction of lysine acetylation by support vector machines,” *Protein and peptide letters*, vol. 16, no. 8, pp. 977–983, 2009. 2
- [10] S. Islam, S. B. S. Mugdha, S. R. Dipta, M. Arafat, S. Shatabda, H. Alinejad-Rokny, and I. Dehzangi, “Methevo: an accurate evolutionary information-based methylation site predictor,” *Neural Computing and Applications*, pp. 1–12, 2022. 2
- [11] S. D. Ali, H. Tayara, and K. T. Chong, “Interpretable machine learning identification of arginine methylation sites,” *Computers in Biology and Medicine*, vol. 147, p. 105767, 2022. 2
- [12] S. M. Azim, A. Sharma, I. Noshadi, S. Shatabda, and I. Dehzangi, “A convolutional neural network based tool for predicting protein ampylation sites from binary profile representation,” *Scientific reports*, vol. 12, 2022. 2
- [13] I. Dehzangi, A. Sharma, and S. Shatabda, “iprotgly-ss: A tool to accurately predict protein glycation site using structural-based features,” in *Computational Methods for Predicting Post-Translational Modification Sites*. Springer, 2022, pp. 125–134. 2
- [14] J. Yu, S. Shi, F. Zhang, G. Chen, and M. Cao, “Predgly: predicting lysine glycation sites for homo sapiens based on xgboost feature optimization,” *Bioinformatics*, vol. 35, no. 16, pp. 2749–2756, 2019. 2
- [15] S. R. Dipta, G. Taherzadeh, M. W. Ahmad, M. E. Arafat, S. Shatabda, and A. Dehzangi, “Semal: Accurate protein malonylation site predictor using structural and evolutionary information,” *Computers in biology and medicine*, vol. 125, p. 104022, 2020. 2, 6, 14, 20, 22, 23

- [16] M. Wang, X. Cui, S. Li, X. Yang, A. Ma, Y. Zhang, and B. Yu, “Deepmal: Accurate prediction of protein malonylation sites by deep neural networks,” *Chemometrics and Intelligent Laboratory Systems*, vol. 207, p. 104175, 2020. 2
- [17] Y. López, A. Dehzangi, H. M. Reddy, and A. Sharma, “C-isumo: a sumoylation site predictor that incorporates intrinsic characteristics of amino acid sequences,” *Computational Biology and Chemistry*, vol. 87, p. 107235, 2020. 2
- [18] R. Harmel and D. Fiedler, “Features and regulation of non-enzymatic post-translational modifications,” *Nature chemical biology*, vol. 14, no. 3, pp. 244–252, 2018. 5
- [19] F. Ali, J. S. Dar, A. R. Magray, B. A. Ganai, M. Chishti *et al.*, “Posttranslational modifications of proteins and their role in biological processes and associated diseases,” in *Protein Modificomics*. Elsevier, 2019, pp. 1–35. 5, 6
- [20] S. Ramazi and J. Zahiri, “Post-translational modifications in proteins: resources, tools and prediction methods,” *Database*, vol. 2021, 2021. 5, 6
- [21] M. M. Hasan and M. S. Khatun, “Prediction of protein post-translational modification sites: an overview,” *Ann Proteom Bioinform*, vol. 2, pp. 049–57, 2018. 5
- [22] S. Zhao, W. Xu, W. Jiang, W. Yu, Y. Lin, T. Zhang, J. Yao, L. Zhou, Y. Zeng, H. Li *et al.*, “Regulation of cellular metabolism by protein lysine acetylation,” *Science*, vol. 327, no. 5968, pp. 1000–1004, 2010. 6
- [23] H. Xu, Y. Wang, S. Lin, W. Deng, D. Peng, Q. Cui, and Y. Xue, “Ptmd: a database of human disease-associated post-translational modifications,” *Genomics, proteomics & bioinformatics*, vol. 16, no. 4, pp. 244–251, 2018. 6
- [24] O. N. Jensen, “Modification-specific proteomics: characterization of post-translational modifications by mass spectrometry,” *Current opinion in chemical biology*, vol. 8, no. 1, pp. 33–41, 2004. 6
- [25] B. Bhinder, C. Gilvary, N. S. Madhukar, and O. Elemento, “Artificial intelligence in cancer research and precision medicine,” *Cancer Discovery*, vol. 11, no. 4, pp. 900–915, 2021. 12

- [26] M. Oberti and I. I. Vaisman, “cnnalpha: protein disordered regions prediction by reduced amino acid alphabets and convolutional neural networks,” *Proteins: Structure, Function, and Bioinformatics*, vol. 88, no. 11, pp. 1472–1481, 2020. 12
- [27] N. Q. K. Le, Q.-T. Ho, T.-T.-D. Nguyen, and Y.-Y. Ou, “A transformer architecture based on bert and 2d convolutional neural network to identify dna enhancers from sequence information,” *Briefings in bioinformatics*, vol. 22, no. 5, p. bbab005, 2021. 12
- [28] L. Wei, P. Xing, J. Tang, and Q. Zou, “Phospred-rf: a novel sequence-based predictor for phosphorylation sites using sequential information only,” *IEEE transactions on nanobioscience*, vol. 16, no. 4, pp. 240–247, 2017. 12, 13
- [29] Y. Dou, B. Yao, and C. Zhang, “Phosphosvm: prediction of phosphorylation sites by integrating various protein sequence attributes with a support vector machine,” *Amino acids*, vol. 46, no. 6, pp. 1459–1469, 2014. 12, 13
- [30] N. Blom, S. Gammeltoft, and S. Brunak, “Sequence and structure-based prediction of eukaryotic protein phosphorylation sites,” *Journal of molecular biology*, vol. 294, no. 5, pp. 1351–1362, 1999. 12, 13
- [31] A. K. Biswas, N. Noman, and A. R. Sikder, “Machine learning approach to predict protein phosphorylation sites by incorporating evolutionary information,” *BMC bioinformatics*, vol. 11, no. 1, pp. 1–17, 2010. 12, 13
- [32] J. Gao, J. J. Thelen, A. K. Dunker, and D. Xu, “Musite, a tool for global prediction of general and kinase-specific phosphorylation sites,” *Molecular & Cellular Proteomics*, vol. 9, no. 12, pp. 2586–2600, 2010. 12
- [33] D. Wang, D. Liu, J. Yuchi, F. He, Y. Jiang, S. Cai, J. Li, and D. Xu, “Musitedeep: a deep-learning based webserver for protein post-translational modification site prediction and visualization,” *Nucleic Acids Research*, vol. 48, no. W1, pp. W140–W146, 2020. 13
- [34] S. Ahmed, M. Kabir, M. Arif, Z. U. Khan, and D.-J. Yu, “Deepppsite: a deep learning-based model for analysis and prediction of phosphorylation sites using efficient sequence information,” *Analytical biochemistry*, vol. 612, p. 113955, 2021. 13

- [35] F. Luo, M. Wang, Y. Liu, X.-M. Zhao, and A. Li, “Deepphos: prediction of protein phosphorylation sites with deep learning,” *Bioinformatics*, vol. 35, no. 16, pp. 2766–2773, 2019. 13
- [36] N. Thapa, M. Chaudhari, A. A. Iannetta, C. White, K. Roy, R. Newman, L. M. Hicks, and K. Dukka, “Chlamy-enphossite: A deep learning-based approach for chlamydomonas reinhardtii-specific phosphorylation site prediction,” 2021. 13
- [37] M. L. Miller, B. Soufi, C. Jers, N. Blom, B. Macek, and I. Mijakovic, “Netphosbac—a predictor for ser/thr phosphorylation sites in bacterial proteins,” *Proteomics*, vol. 9, no. 1, pp. 116–125, 2009. 13, 34, 51
- [38] Z. Li, P. Wu, Y. Zhao, Z. Liu, and W. Zhao, “Prediction of serine/threonine phosphorylation sites in bacteria proteins,” in *Advance in Structural Bioinformatics*. Springer, 2015, pp. 275–285. 13
- [39] Q.-b. Zhang, K. Yu, Z. Liu, D. Wang, Y. Zhao, S. Yin, and Z. Liu, “Prediction of prkc-mediated protein serine/threonine phosphorylation sites for bacteria,” *PloS one*, vol. 13, no. 10, p. e0203840, 2018. 13
- [40] M. Hasan, M. Rashid, M. Khatun, H. Kurata *et al.*, “Computational identification of microbial phosphorylation sites by the enhanced characteristics of sequence information,” *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019. 14, 34, 51
- [41] A. Dehzangi, S. Sohrabi, R. Heffernan, A. Sharma, J. Lyons, K. Paliwal, and A. Sattar, “Gram-positive and gram-negative subcellular localization using rotation forest and physicochemical-based features,” *BMC bioinformatics*, vol. 16, no. 4, pp. 1–8, 2015. 14, 22, 23
- [42] S. Shatabda, S. Saha, A. Sharma, and A. Dehzangi, “iphloc-es: Identification of bacteriophage protein locations using evolutionary and structural features,” *Journal of theoretical biology*, vol. 435, pp. 229–237, 2017. 14
- [43] A. Dehzangi, Y. López, S. P. Lal, G. Taherzadeh, A. Sattar, T. Tsunoda, and A. Sharma, “Improving succinylation prediction accuracy by incorporating the secondary structure via helix, strand and coil, and evolutionary information from profile bigrams,” *PloS one*, vol. 13, no. 2, p. e0191900, 2018. 14, 20

- [44] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li, “Cd-hit: accelerated for clustering the next-generation sequencing data,” *Bioinformatics*, vol. 28, no. 23, pp. 3150–3152, 2012. 17
- [45] W. Li and A. Godzik, “Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences,” *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006. 17
- [46] Y. Huang, B. Niu, Y. Gao, L. Fu, and W. Li, “Cd-hit suite: a web server for clustering and comparing biological sequences,” *Bioinformatics*, vol. 26, no. 5, pp. 680–682, 2010. 17
- [47] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, “Gapped blast and psi-blast: a new generation of protein database search programs,” *Nucleic acids research*, vol. 25, no. 17, pp. 3389–3402, 1997. 18
- [48] Y. Yang, R. Heffernan, K. Paliwal, J. Lyons, A. Dehzangi, A. Sharma, J. Wang, A. Sattar, and Y. Zhou, “Spider2: A package to predict secondary structure, accessible surface area, and main-chain torsional angles by deep neural networks,” in *Prediction of protein secondary structure*. Springer, 2017, pp. 55–63. 18, 21
- [49] R. Heffernan, K. Paliwal, J. Lyons, A. Dehzangi, A. Sharma, J. Wang, A. Sattar, Y. Yang, and Y. Zhou, “Improving prediction of secondary structure, local backbone angles and solvent accessible surface area of proteins by iterative deep learning,” *Scientific reports*, vol. 5, no. 1, pp. 1–11, 2015. 18, 21
- [50] I. Mani and I. Zhang, “knn approach to unbalanced data distributions: a case study involving information extraction,” in *Proceedings of workshop on learning from imbalanced datasets*, vol. 126. ICML, 2003, pp. 1–7. 18, 22
- [51] A. Sharma, J. Lyons, A. Dehzangi, and K. K. Paliwal, “A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition,” *Journal of theoretical biology*, vol. 320, pp. 41–46, 2013. 20
- [52] A. Chandra, A. Sharma, A. Dehzangi, D. Shigemizu, and T. Tsunoda, “Bigram-pgk: phosphoglycerylation prediction using the technique of bigram probabilities of position specific scoring matrix,” *BMC molecular and cell biology*, vol. 20, no. 2, pp. 1–9, 2019.

- [53] M. W. Ahmad, M. E. Arafat, G. Taherzadeh, A. Sharma, S. R. Dipta, A. Dehzangi, and S. Shatabda, “Mal-light: Enhancing lysine malonylation sites prediction problem using evolutionary-based features,” *IEEE access*, vol. 8, pp. 77 888–77 902, 2020. 20
- [54] M. M. Islam, S. Saha, M. M. Rahman, S. Shatabda, D. M. Farid, and A. Dehzangi, “iprotgly-ss: Identifying protein glycation sites using sequence and structure based features,” *Proteins: Structure, Function, and Bioinformatics*, vol. 86, no. 7, pp. 777–789, 2018. 20
- [55] H. M. Reddy, A. Sharma, A. Dehzangi, D. Shigemizu, A. A. Chandra, and T. Tsunoda, “Glystruct: glycation prediction using structural properties of amino acid residues,” *BMC bioinformatics*, vol. 19, no. 13, pp. 55–64, 2019. 20
- [56] A. Dehzangi, K. Paliwal, J. Lyons, A. Sharma, and A. Sattar, “Enhancing protein fold prediction accuracy using evolutionary and structural features,” in *IAPR International Conference on Pattern Recognition in Bioinformatics*. Springer, 2013, pp. 196–207. 20
- [57] S. Y. Chowdhury, S. Shatabda, and A. Dehzangi, “idnaprot-es: Identification of dna-binding proteins using evolutionary and structural features,” *Scientific reports*, vol. 7, no. 1, pp. 1–14, 2017. 20
- [58] H. He, Y. Bai, E. A. Garcia, and S. Li, “Adasyn: Adaptive synthetic sampling approach for imbalanced learning,” in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 1322–1328. 22
- [59] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002. 22
- [60] I. Tomek, “Two modifications of cnn.” 1976. 22
- [61] A. Dehzangi, S. Phon-Amnuaisuk, M. Manafi, and S. Safa, “Using rotation forest for protein fold prediction problem: An empirical study,” in *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. Springer, 2010, pp. 217–227. 22

- [62] A. Bustamam, M. I. Musti, S. Hartomo, S. Aprilia, P. P. Tampubolon, and D. Lestari, “Performance of rotation forest ensemble classifier and feature extractor in predicting protein interactions using amino acid sequences,” *BMC genomics*, vol. 20, no. 9, pp. 1–13, 2019. 22
- [63] L. Wang, Z.-H. You, X. Yan, S.-X. Xia, F. Liu, L.-P. Li, W. Zhang, and Y. Zhou, “Using two-dimensional principal component analysis and rotation forest for prediction of protein-protein interactions,” *Scientific reports*, vol. 8, no. 1, pp. 1–10, 2018. 22
- [64] L. Wang, Z.-H. You, X. Chen, X. Yan, G. Liu, and W. Zhang, “Rfdt: A rotation forest-based predictor for predicting drug-target interactions using drug structure and protein sequence information,” *Current Protein and Peptide Science*, vol. 19, no. 5, pp. 445–454, 2018. 22
- [65] Z.-H. You, X. Li, and K. C. Chan, “An improved sequence-based prediction protocol for protein-protein interactions using amino acids substitution matrix and rotation forest ensemble classifiers,” *Neurocomputing*, vol. 228, pp. 277–282, 2017. 22
- [66] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, “Rotation forest: A new classifier ensemble method,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 10, pp. 1619–1630, 2006. 22, 30
- [67] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010. 23, 30
- [68] Y. Deng, Y. Fu, H. Zhang, X. Liu, and Z. Liu, “Protein post-translational modification site prediction using deep learning,” *Procedia Computer Science*, vol. 198, pp. 480–485, 2022. 36
- [69] Z. U. Khan and D. Pi, “Deepsspred: A deep learning based sulfenylation site predictor via a novel nsegmented optimize federated feature encoder,” *Protein and Peptide Letters*, vol. 28, no. 6, pp. 708–721, 2021. 36

- [70] Q. Ning, X. Zhao, and Z. Ma, “A novel method for identification of glutarylation sites combining borderline-smote with totem links technique in imbalanced data,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021. 36
- [71] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141. 38
- [72] N. Strodthoff, P. Wagner, M. Wenzel, and W. Samek, “Udsmprot: universal deep sequence models for protein classification,” *Bioinformatics*, vol. 36, no. 8, pp. 2401–2409, 2020. 55
- [73] N. Brandes, D. Ofer, Y. Peleg, N. Rappoport, and M. Linial, “Proteinbert: A universal deep-learning model of protein sequence and function,” *Bioinformatics*, vol. 38, no. 8, pp. 2102–2110, 2022. 55

## Appendix A

# Supplementary Data for RotPhoPred

### A.1 Comparing different window sizes

Table A.1 presents the performance of RotPhoPred predicting pS, pT, and pY sites when different window sizes are used for feature extraction. The results are based on the independent test set.

| Task | Window Size | Ac (%)      | MCC         | Pr          | auROC       | F1          | Sn(%)       | Sp(%)       |
|------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| pS   | 11          | 88.1        | 0.78        | <b>0.97</b> | 0.88        | 0.87        | 78.6        | <b>97.6</b> |
|      | 21          | <b>91.0</b> | <b>0.82</b> | 0.92        | <b>0.96</b> | <b>0.91</b> | <b>90.0</b> | 92.1        |
|      | 31          | 88.2        | 0.77        | 0.89        | 0.88        | 0.88        | 86.8        | 89.7        |
|      | 41          | 87.6        | 0.75        | 0.87        | 0.88        | 0.87        | 86.5        | 88.8        |
| pT   | 11          | 80.7        | 0.62        | 0.77        | 0.81        | 0.82        | 87.4        | 74.0        |
|      | 21          | <b>86.3</b> | <b>0.74</b> | <b>0.96</b> | <b>0.90</b> | <b>0.85</b> | <b>75.4</b> | <b>97.2</b> |
|      | 31          | 84.1        | 0.69        | 0.92        | 0.84        | 0.82        | 75.0        | 93.3        |
|      | 41          | 81.4        | 0.64        | 0.88        | 0.81        | 0.76        | 72.0        | 90.8        |
| pY   | 11          | 84.7        | 0.69        | 0.88        | 0.85        | 0.84        | <b>79.9</b> | 89.4        |
|      | 21          | <b>86.4</b> | <b>0.74</b> | <b>0.94</b> | <b>0.89</b> | <b>0.84</b> | 78.2        | 94.7        |
|      | 31          | 93.3        | 0.71        | 0.88        | 0.81        | 0.74        | 64.1        | <b>98.4</b> |
|      | 41          | 83.5        | 0.68        | 0.92        | 0.83        | 0.81        | 72.9        | 94.0        |

**Table A.1:** Independent test results of RotPhosPred using different window sizes for feature extraction.

## A.2 Comparing data balancing techniques

Table A.2 demonstrates the 5-fold cross-validation results of RotPhoPred on upsampling and downsampling techniques: SMOTE and NearMiss, respectively, for pS, pT, and pY site prediction.

| Task | Balancing Methods | Sn (%) | Sp (%) | Pr   | Ac   | F1   | MCC  | auROC |
|------|-------------------|--------|--------|------|------|------|------|-------|
| pS   | NearMiss          | 87.7   | 97.6   | 0.93 | 92.7 | 0.92 | 0.86 | 0.93  |
|      | SMOTE             | 85.4   | 81.7   | 0.83 | 83.5 | 0.84 | 0.67 | 0.84  |
| pT   | NearMiss          | 79.4   | 99.3   | 0.99 | 89.3 | 0.88 | 0.80 | 0.89  |
|      | SMOTE             | 85.5   | 82.2   | 0.83 | 83.9 | 0.84 | 0.68 | 0.84  |
| pY   | NearMiss          | 75.2   | 99.3   | 0.99 | 87.3 | 0.85 | 0.77 | 0.87  |
|      | SMOTE             | 84.2   | 81.3   | 0.82 | 82.8 | 0.83 | 0.66 | 0.83  |

**Table A.2:** 5-fold cross-validation results of RotPhoPred for predicting pS, pT, and pY when NearMiss and SMOTE methods are applied to balance the training dataset.

## Appendix B

# Supplementary Data for DeepPhoPred

### B.1 Cross-validation results of DeepPhoPred with and without SE blocks

Table B.1 shows the results of DeepPhoPred with and without SE blocks on 5-folds cross-validation.

| Task | Model            | Ac (%) | MCC  | Pr   | auROC | F1   | Sn (%) | Sp (%) | auPR |
|------|------------------|--------|------|------|-------|------|--------|--------|------|
| pS   | With SE block    | 99.5   | 0.99 | 0.99 | 0.99  | 1.00 | 100    | 99.0   | 1.00 |
|      | Without SE block | 96.8   | 0.94 | 0.95 | 0.97  | 0.97 | 99.0   | 94.5   | 1.00 |
| pT   | With SE block    | 99.4   | 0.99 | 0.99 | 0.99  | 0.99 | 99.7   | 99.1   | 1.00 |
|      | Without SE block | 95.6   | 0.91 | 0.92 | 0.96  | 0.96 | 99.6   | 91.5   | 1.00 |
| pY   | With SE block    | 98.8   | 0.98 | 0.98 | 0.99  | 0.99 | 99.5   | 98.1   | 1.00 |
|      | Without SE block | 96.0   | 0.92 | 0.93 | 0.96  | 0.96 | 99.7   | 92.2   | 1.00 |

**Table B.1:** Cross-validation results of DeepPhoPred with and without SE blocks

### B.2 Comparing different encoding methods on independent test

Table B.2 presents the performance of DeepPhoPred using different encoding methods on the independent test.

### B.3 Loss and Accuracy Curves

Figure B.1 shows the loss and accuracy curves (left to right) for training DeepPhoPred to predict pS, pT, and pY sites (top to bottom) using both evolutionary and structural features.

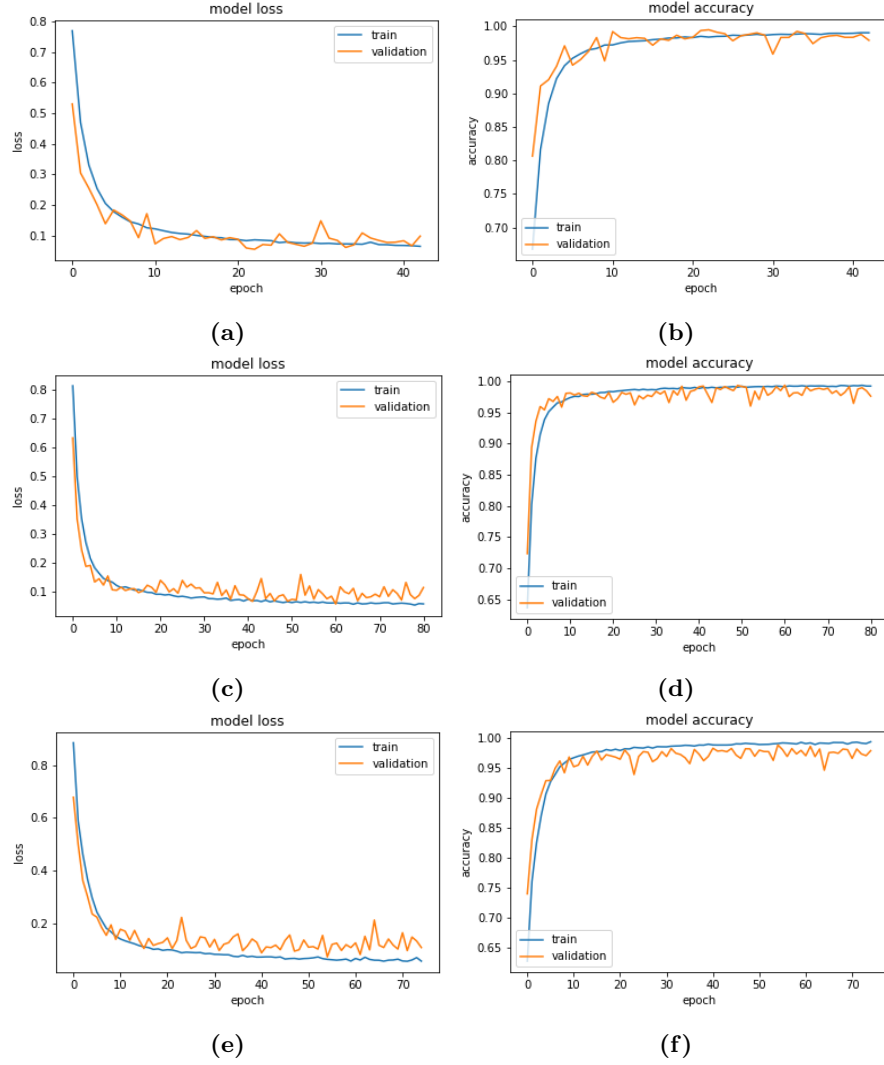


**Figure B.1:** Loss and accuracy curves (left to right) for training DeepPhoPred to predict pS, pT, and pY sites (top to bottom)

| Task | Encoding Method | Ac(%) | MCC  | Pr   | auROC | F1   | Sn (%) | Sp (%) | auPR |
|------|-----------------|-------|------|------|-------|------|--------|--------|------|
| pS   | Combined        | 99.0  | 0.93 | 0.88 | 0.99  | 0.94 | 100    | 98.9   | 1.00 |
|      | Structural      | 85.6  | 0.54 | 0.34 | 0.92  | 0.51 | 99.7   | 84.4   | 0.82 |
|      | Evolutionary    | 87.4  | 0.57 | 0.37 | 0.93  | 0.54 | 100    | 86.4   | 0.95 |
| pT   | Combined        | 99.0  | 0.94 | 0.90 | 0.99  | 0.95 | 99.2   | 99.0   | 0.99 |
|      | Structural      | 85.0  | 0.55 | 0.37 | 0.91  | 0.54 | 98.8   | 83.6   | 0.79 |
|      | Evolutionary    | 87.4  | 0.57 | 0.37 | 0.94  | 0.54 | 100    | 86.4   | 0.97 |
| pY   | Combined        | 98.7  | 0.95 | 0.93 | 0.99  | 0.96 | 99.3   | 98.7   | 1.00 |
|      | Structural      | 84.0  | 0.61 | 0.48 | 0.90  | 0.64 | 97.9   | 81.6   | 0.88 |
|      | Evolutionary    | 83.3  | 0.60 | 0.47 | 0.89  | 0.63 | 96.8   | 80.9   | 0.93 |

**Table B.2:** Independent test results of different encoding methods.