



**Daffodil**  
*International*  
**University**

**Neuro\_PP: Identification of Neuropeptide using Ensemble Machine  
Learning with Feature Encoding Technique.**

**Submitted by**

Tareq Rahman

ID: 201-35-533

Department of Software Engineering

Daffodil International University

**Supervised by**

Mr. Mohammad Khaled Sohel

Assistant Professor

Department of Software Engineering

Daffodil International University

This Thesis paper has been submitted in fulfillment of the requirements for the degree  
of Bachelors of Science in Software Engineering

Spring 2024

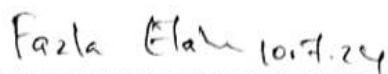
© All right Reserved by Daffodil International University

## Approval of Thesis

### APPROVAL

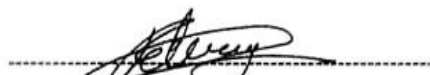
This thesis titled on “Neuro\_PP: Identification of Neuropeptide using Ensemble Machine Learning with Feature Encoding Technique”, submitted by **Tareq Rahman (ID: 201-35-533)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

### BOARD OF EXAMINERS



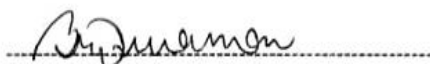
**Chairman**

**Dr. Md. Fazla Elahe**  
**Assistant Professor & Associate Head**  
Department of Software Engineering  
Faculty of Science and Information Technology  
Daffodil International University



**Internal Examiner 1**

**Tapushe Kabaya Toma**  
**Assistant Professor**  
Department of Software Engineering  
Faculty of Science and Information Technology  
Daffodil International University



**Internal Examiner 2**

**Khalid Been Md. Badruzzaman Biplob**  
**Lecturer (Senior Scale)**  
Department of Software Engineering  
Faculty of Science and Information Technology  
Daffodil International University



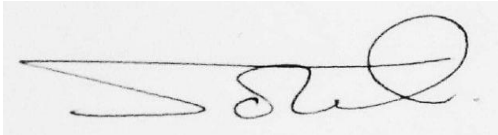
**External Examiner**

**Md Mostafiz Khan**  
**Managing Director**  
Tecognize Solutions Limited

## Declaration

I announce that I am rendering this study document under Mr. Mohammad Khaled Sohel, Assistant Professor, Department of Software Engineering, Daffodil International University. I therefore, state that this work or any portion of it was not proposed here therefore for Bachelor's degree or any graduation.

Supervised By



-----  
Mr. Mohammad Khaled Sohel  
Assistant Professor  
Department of Software Engineering  
Daffodil International University

Submitted by



-----  
Tareq Rahman  
ID: 201-35-533  
Department of Software Engineering  
Daffodil International University

## **ACKNOWLEDGEMENT**

The motivation behind my study was purely driven by the desire to gain knowledge and deepen my understanding. This research, titled Neuro\_PP: Identification of Neuropeptide using Ensemble Machine Learning with Feature Encoding Technique, has been a significant learning journey. Firstly, I express my heartfelt gratitude to the All-knowing, whose guidance and wisdom made this study possible. Without His support, this achievement would not have been realized. Secondly, I am profoundly grateful to my parents for their unwavering support and for bringing me to where I am today. I would also like to extend my sincere thanks to Mr. Mohammad Khaled Sohel, Assistant Professor in the Department of Software Engineering, for his guidance and encouragement. Additionally, I am thankful to all the respected teachers who have taught me throughout my educational journey. I feel fortunate to have had them as my mentors. My gratitude also goes to Daffodil International University, whose constant supervision and support, particularly from Mr. Mohammad Khaled Sohel, provided the necessary resources and motivation to complete this research. Lastly, I would like to thank my batch mates and the members of DIU for their cooperation and support, which were instrumental in achieving this goal.

## ABSTRACT

**Objective:** Neuropeptides, naturally present in the nervous system, play a crucial role in regulating physiological processes. With significant activity in modulating neuronal communication, they present substantial potential for therapeutic development in neurological disorders. **Methods:** This study introduces Neuro\_PP, an innovative method for the identification of Neuropeptides using the Amino Acid Composition (AAC) feature extraction approach. Our proposed model employs a stacking-based ensemble learning framework to enhance predictive accuracy. **Results:** Through rigorous experimentation and comprehensive dataset evaluation, Neuro\_PP demonstrates outstanding results, achieving an accuracy of 0.9821, a Matthews Correlation Coefficient of 0.8427, and an Area under the Curve (AUC) of 0.9991. These outcomes highlight the effectiveness of Neuro\_PP in accurately identifying neuropeptides, establishing it as a valuable tool in the field of neuropeptide prediction. **Discussion:** The findings underscore the significance of integrating AAC feature extraction with a stacking-based ensemble learning model for robust and precise neuropeptide identification. Practically, Neuro\_PP shows considerable potential to expedite the drug discovery process for neuropeptides, possibly leading to the development of novel therapeutic interventions. This research, combining domain expertise with cutting-edge technology, represents a significant advancement in precision-driven predictive modeling, providing valuable insights for experts in bioinformatics, computational biology, and drug discovery.

**Keywords:** Neuropeptide, Stacking, Ensemble, Latent Semantic Analysis, Feature Extraction, XGB, Amino Acid Composition.

## Table of Contents

<b>Approval of Thesis</b> .....	<b>ii</b>
Declaration .....	iii
ACKNOWLEDGEMENT .....	iv
ABSTRACT .....	v
Table of Contents .....	vi
Table of Figures .....	vii
Table of Tables .....	viii
<b>1 INTRODUCTION</b> .....	<b>1</b>
<b>2 LITERATURE REVIEW</b> .....	<b>3</b>
<b>3 METHODS AND MATERIALS</b> .....	<b>6</b>
3.1 Dataset Description .....	6
3.2 Sequence Analysis (COBALT) .....	7
3.3 Data Balance (SMOTE) .....	7
3.4 Feature Encoding .....	7
3.4.1 Amphiphilic Pseudo-Amino Acid Composition (APAAC) .....	8
3.4.2 Pseudo-Amino Acid Composition (PAAC) .....	8
3.4.3 Amino Acid Composition (AAC) .....	9
3.4.4 Composition of k-Spaced Amino Acid Group Pairs (CKSAAGP) .....	9
3.4.5 Composition (CTDC) .....	10
3.5 Data Balancing .....	10
3.6 Machine Learning Model .....	11
3.6.1 Random Forest (RF) .....	11
3.6.2 Extra Tree Classifier (ETC) .....	11
3.6.3 K Nearest Neighbor (KNN) .....	12
3.6.4 Categorical Boosting (CAT) .....	12
3.6.5 AdaBoost Classifier (ADB) .....	12
3.6.6 Extreme Gradient Boosting (XGB) .....	13
3.6.7 Light Gradient Boosting Machine (LGBM) .....	13
3.6.8 Decision Tree Classifier (DT) .....	13
3.7 Performance Evaluation .....	14
<b>4 RESULT ANALYSIS</b> .....	<b>16</b>
4.1 Analysis of Peptide Sequence .....	16
4.2 ML Classifiers Results Analysis .....	16
<b>5 DISCUSSION</b> .....	<b>24</b>
<b>6 CONCLUSION</b> .....	<b>26</b>
<b>7 REFERENCES</b> .....	<b>27</b>

## Table of Figures

Figure 3.1: Framework for constructing Neuro_PP.....	6
Figure 4.1: Amino acid percentages of the NP and Non-NP sequence of the utilized datasets.....	16
Figure 4.2: Training performance comparison of the applied classifiers on different feature encoding methods. Subplot (A) for AAC, subplot (B) for APAAC, subplot (C) for CKSAAGP, subplot (E) for CTDC, and subplot (D) for PAAC feature extractor..	20
Figure 4.3: Independent test performance comparison of the applied classifiers on different feature encoding methods. Subplot (A) for AAC, subplot (B) for APAAC, subplot (C) for CKSAAGP, subplot (E) for CTDC, and subplot (D) for PAAC feature extractor. ....	21
Figure 4.4: Comparison the accuracy of the various feature extractors and applied classifiers. Subplot (A) refers the CV on training dataset and subplot (B) for the independent test. ....	22
Figure 4.5: ROC curves with AUC scores for our applied classifiers using the proposed feature extractor: Subplot (A) represents the training dataset, and Subplot (B) illustrates the independent test dataset.....	23

## Table of Tables

Table 4.1: CV results of different feature extractors with the applied classifiers on training dataset. ....	17
Table 4.2: Independent test results of various feature extractors with used classifiers. ....	19
Table 5.1: Comparison of the Neuro_PP model with the existing state art of model.....	23



# 1 INTRODUCTION

Neurological diseases and CNS disorders affect about 17% of the global population, impacting nearly 1 billion people and causing 6.8 million deaths annually. The economic burden of these illnesses in Europe alone was estimated at 139 billion euros in 2004. Neuropeptides (NPs), which are small peptides crucial for the endocrine and nervous systems, play significant roles in many neurological conditions such as Alzheimer's, Parkinson's, depression, pain disorders, and addiction. However, identifying and characterizing NPs is a challenging, labor-intensive, and costly process. Traditional methods, though effective, are not scalable and lack the efficiency required for high-throughput analysis. While recent advancements in machine learning (ML) present a promising solution, current models struggle with generalizability and prediction accuracy across various datasets.

Neuropeptides (NPs) are small peptides, typically composed of fewer than 100 amino acids that play dual roles as hormones in the endocrine system and neurotransmitters in the nervous and immune systems. Synthesized in the cell bodies of neurosecretory cells, NPs exert their effects on various target cells, including neurons, glial cells, gland cells, and muscle cells. Their complex chemical structure allows them to bind selectively to specific receptors, leading to highly targeted actions with minimal side effects compared to classic neurotransmitters. NPs are integral to numerous physiological processes, such as development, hormonal regulation, and immune responses. They are also critical in the body's adaptation to stress, pain, injury, and drug abuse. This broad functional scope makes NPs pivotal in maintaining homeostasis and responding to environmental challenges, underscoring their importance in daily life and health. The significance of NPs extends to medical research, particularly in the context of nervous system disorders. The precise targeting abilities of NPs make them attractive candidates for new therapeutic interventions. However, the identification and characterization of NPs have traditionally been labor-intensive and costly, involving techniques like liquid chromatography coupled with tandem mass spectrometry (LC-MS/MS), bioassays, receptor-binding assays, and genetic analysis. Recent advancements in high-throughput next-generation sequencing and computational approaches have revolutionized NP research. These technologies facilitate the rapid identification of NPs by leveraging bioinformatics tools that

combine homology searching with NP-specific characteristics, such as N-terminal signal peptides and dibasic cleavage sites. Despite these technological advancements, identifying NPs remains challenging due to the need for comprehensive NP precursor databases and the potential for false positives in data analysis. Machine learning (ML) has emerged as a transformative approach in predicting bioactive peptides, including NPs. Various ML algorithms, such as k-nearest neighbor (KNN), extremely randomized trees (ERT), artificial neural network (ANN), logistic regression (LR), and extreme gradient boosting (XGBoost), have been employed to enhance the prediction accuracy of NPs. In response to the limitations of traditional methods, researchers have developed novel ensemble learning tools like PredNeuroP. This tool uses a two-layer stacking framework that combines multiple ML algorithms and feature encodings to improve prediction accuracy. By leveraging the strengths of each model, PredNeuroP enhances the identification of NPs across different animal phyla, showcasing consistent performance on both training and test datasets. The ability to accurately predict and identify NPs holds significant promise for the development of new drugs targeting nervous system disorders. By advancing our understanding and detection of NPs, researchers can develop more effective treatments for diseases, ultimately contributing to better health outcomes and improved quality of life.

## 2 LITERATURE REVIEW

Recent progress in computational models for predicting neuropeptides has greatly improved accuracy through advanced machine learning and deep learning methods. In recent years, several studies have been carried out where researchers have developed models aimed at predicting neuropeptides. These efforts include the proposal and evaluation of various computational approaches designed to enhance neuropeptide prediction.

Year	Title	Author	Feature Extractor	Algorithm	Model Name	Result
2022	NeuroCNN_GNB: an ensemble model to predict neuropeptides based on a convolution neural network and Gaussian naive Bayes	Di Liu, Zhengkui Lin	One Hot AAIndex G-Gap Word2Vec	LR ADABOOST GBDT GaussianNB XGBoost	<b>NeuroCNN-GNB</b>	<b>96.1%</b>
2022	NeuroPred-SVM: A New Model for Predicting Neuropeptides Based on Embeddings of BERT	Yufeng Liu, Shuyu Wang,	AAT CTD PSAAC AAP AAC	SVM AB XGB KNN RF	<b>NeuroPred-SVM</b>	<b>94.9%</b>
2021	NeuroPred-FRL: an interpretable prediction model for identifying neuropeptide using feature representation learning	Md Mehedi Hasan, Md Ashad Alam	CF PF PCF Optimal CF(16D) Optimal PF(12D) Optimal PCF(20D)	SVM RF AB NB KNN	<b>NeuroPred-FRL</b>	<b>93.6%</b>
2017	NeuroPred-Fuse: an interpretable stacking model for prediction of neuropeptides by fusing sequence Information and feature selection methods.	Mingming Jiang, Bowen Zhao	AAC DPC GGAP ASDC PSAAC CTD	RF GBDT XGBOOST	<b>NeuroPred-Fuse</b>	<b>92.7%</b>
2016	Prediction of Neuropeptides from Sequence Information Using Ensemble Classifier and Hybrid Features	Yannan Bin, Wei Zhang	AAC DPC BPNC AAE	RF XGBoost KNN	<b>PredNeuroP</b>	<b>92.1%</b>
2015	NeuroPIpred: a tool to predict, design and scan insect neuropeptides	Piyush Agrawal, Sumit Kumar	ACC DC N10C10	SVM RF	<b>NeuroPIpred</b>	<b>89.6%</b>

The NeuroCNN-GNB paper discusses the significance and challenges of identifying neuropeptides (NPs) in neurological diseases, highlighting the limitations of traditional methods such as mass spectrometry and immunoassays, and the potential of machine learning (ML) models, particularly ensemble learning, to improve accuracy and scalability; it proposes a new model, NeuroCNN-GNB, which integrates multiple feature extraction methods and uses a stacking strategy with convolutional neural networks (CNN) and Gaussian Naive Bayes to achieve high prediction accuracy and robustness, and demonstrates superior performance through metrics like AUC and accuracy, emphasizing its potential impact on NP identification and therapeutic research, and concludes with the development of a web server and open-source availability to facilitate broader application.

NeuroPIpred is an advanced tool developed for the prediction of neuropeptides, leveraging ensemble learning to integrate outputs from multiple base learners. These learners are trained on a wide array of features, encompassing physicochemical properties, sequence motifs, and structural characteristics of neuropeptides. The ensemble approach allows NeuroPIpred to achieve high predictive accuracy and robustness, with significant metrics such as an accuracy of 0.9020, an AUC of 0.9550, and an MCC of 0.7200. This tool demonstrates the effectiveness of combining diverse predictive models to address the complexity inherent in neuropeptide data and underscores the potential of ensemble learning techniques in bioinformatics.

NeuroPred-FRL represents a significant advancement in the field of neuropeptide prediction through the use of feature representation learning. The model incorporates extensive feature extraction techniques, including amino acid composition, dipeptide composition, and secondary structure predictions, to comprehensively characterize neuropeptide sequences. By optimizing these features and employing a sophisticated machine learning framework, NeuroPred-FRL achieves high performance with an accuracy of 0.9110, an AUC of 0.9558, and an MCC of 0.7400. The study underscores the crucial role of detailed and diverse feature extraction in enhancing the predictive power and reliability of machine learning models in the context of neuropeptide identification and characterization.

PredNeuroP: By focusing on advanced machine learning techniques and meticulous feature selection, PredNeuroP optimizes the identification of predictive features using various algorithms. It achieves a high accuracy of 0.9400, an AUC of 0.9750, and an MCC of 0.7900, emphasizing the importance of strategic feature engineering and sophisticated algorithmic approaches in accurately predicting neuropeptides (Wang et al., 2023). These models collectively illustrate the significant advancements from basic to complex computational methods, achieving marked improvements in neuropeptide prediction through the integration of diverse and sophisticated techniques.

### 3 METHODS AND MATERIALS

We give a thorough explanation of the techniques and approaches used in this investigation in this section. Section 3.1 covers the dataset, Section 3.2 discusses feature extraction techniques, Section 3.3 discusses data balancing techniques, Section 3.4 explores machine learning algorithms, and Section 3.5 focuses on performance evaluation metrics. The following subsections outline the specific procedures of the work.

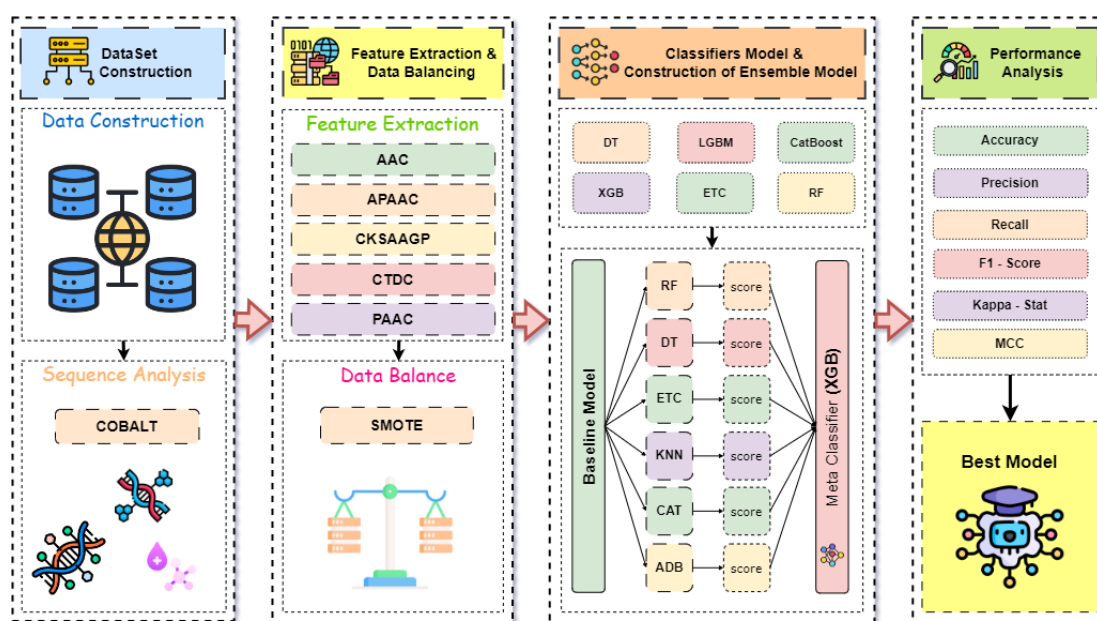


Figure 3.1: Framework for constructing Neuro\_PP.

#### 3.1 Dataset Description

In this study, we have collected a dataset from (Hasan et al., 2021) studies. In their research, they prepared total four datasets: two is the main of training dataset, and the other is an alternative of testing dataset. The main dataset consists of 1945 positive and 1945 negative peptide sequences. And the alternative dataset contains 495 positive and 495 negative peptide sequences (Hasan et al. 2021). For our study purpose, we combined the main training two datasets into one. As a result, our dataset contains 3890 positive and 3890 negative peptide sequences. And also we combined the testing two datasets into one. As a result, our dataset contains total 990 data, where 495 of positive and 495 negative sequence.

### **3.2 Sequence Analysis (COBALT)**

COBALT (Constraint-Based Multiple Alignment Tool) is a bioinformatics method for aligning multiple protein sequences. It combines pairwise and multiple sequence alignment algorithms and integrates sequence conservation information from related sequences to enhance alignment accuracy. Utilizing a constraint-based approach, COBALT employs both local and global alignment strategies to manage sequences with varying similarity levels. Key features include the incorporation of conserved domain information from NCBI's Conserved Domain Database (CDD), use of position-specific scoring matrices (PSSMs) for scoring alignment positions, and providing visual representations for analyzing sequence conservation patterns and identifying functionally important protein regions. Freely accessible via the NCBI website, COBALT is a widely used tool in bioinformatics and comparative genomics research.

### **3.3 Data Balance (SMOTE)**

SMOTE (Synthetic Minority Over-sampling Technique) is a popular data balancing method in machine learning, addressing imbalanced datasets where one class is underrepresented. It generates synthetic samples for the minority class by selecting a minority instance, identifying its k-nearest neighbors, and creating new instances by interpolating between the sample and its neighbors. This process increases the number of minority class instances, balancing the dataset. SMOTE improves model performance by providing a more balanced class distribution, leading to better accuracy and generalization. It's versatile and can be applied to various data types and machine learning algorithms, enhancing model robustness and reliability.

### **3.4 Feature Encoding**

Feature encoding, also known as feature extraction, is an essential process in preparing data for machine learning algorithms. This step is particularly crucial because machine learning models cannot directly process string or sequence data. Therefore, researchers must transform such data into a numerical format that models can interpret and analyze (Murakami et al., 2010). For instance, protein sequences, which are typically composed of characters representing various amino acids, need to

be converted into numerical representations. This conversion facilitates the application of machine learning techniques to these sequences. Effective feature encoding methods enable models to capture and utilize the underlying patterns and characteristics of the data, thereby enhancing their predictive performance.

### 3.4.1 Amphiphilic Pseudo-Amino Acid Composition (APAAC)

APAAC is a physicochemical property-based feature extraction method. Here, the amino acid properties have been considered to represent feature vectors. The number of feature vector dimensions is 24 (Chou et al. 2005, Chou et al. 2001, and Chen et al. 2021). APAAC is define as:

$$P_c = \frac{\omega \tau_\mu}{\sum_{r=1}^{20} f_r + w \sum_{j=1}^2 \tau_j}, (1 < u < 20 + 2\lambda)$$

Here, w is the weighting factor, in iLearnPlus, w is set to 0.5 (Chou et al. 2005)

### 3.4.2 Pseudo-Amino Acid Composition (PAAC)

This group of descriptors has been proposed in (83, 84). Let  $H_1^0(i)$ ,  $H_2^0(i)$ ,  $M^0(i)$  for  $i = 1, 2, 3, \dots, 20$  be the original hydrophobicity values, the original hydrophilicity values and the original side chain masses of the 20 natural amino acids, respectively. They are converted to the following quantities by a standard conversion:

$$H_1(i) = \frac{H_1^0(i) - \frac{1}{20} \sum_{i=1}^{20} H_1^0(i)}{\sqrt{\frac{\sum_{i=1}^{20} [H_1^0(i) - \frac{1}{20} \sum_{i=1}^{20} H_1^0(i)]^2}{20}}}$$

Where  $H_2^0(i)$  and  $M^0(i)$  are normalized as  $H_2(i)$  and  $M(i)$  in the same manner. An example of the correlation function is provided in the following Figure **S26**.



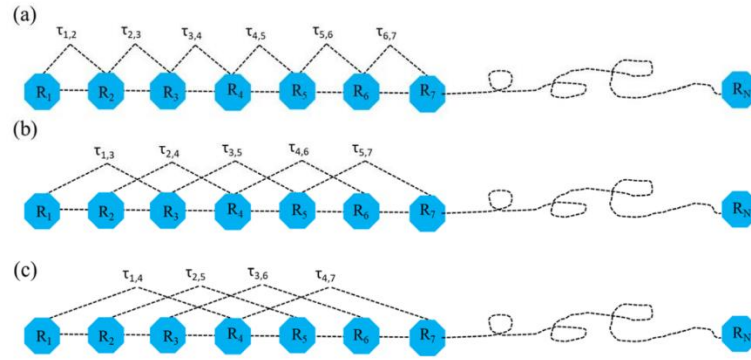


Figure S26. A schematic illustration showing (a) the first-tier, (b) the second-tier, and (3) the third-tier sequence order correlation mode along a protein sequence. (a) Reflects the coupling mode between all the most adjacent residues, (b) shows the coupling between the adjacent plus one residue, and (c) shows the coupling between the adjacent plus two residues. This figure is adapted from (84) for illustration purposes.

Next, a correlation function can be defined as:

$$\theta(R_i, R_j) = \frac{1}{3} \{ [H_1(R_i) - H_1(R_j)] + [H_1(R_i) - H_1(R_j)]^2 + [M(R_i) - M(R_j)] \}$$

This correlation function is actually an averaged value for the three amino acid properties: hydrophobicity value, hydrophilicity value and side chain mass. Therefore, we can extend this definition of correlation function for one amino acid property or for a set of an amino acid properties.

### 3.4.3 Amino Acid Composition (AAC)

By adhering to the traditional definition of this approach, the feature set comprises 20 distinct components, symbolizing the 20 native amino acids found in proteins. Amino acid composition, in this context, pertains to the frequency of occurrence of each of these 20 components within a specific protein (Saidi et al. 2010, Chen et al.). The dimension of the AAC is 20. AAC is denoted by:

$$f(t) = \frac{N(t)}{N}, t \in (A, C, D, \dots, Y)$$

### 3.4.4 Composition of k-Spaced Amino Acid Group Pairs (CKSAAGP)

The Composition of k-Spaced Amino Acid Group Pairs (CKSAAGP) is a variation of the CKSAAP descriptor, which is our own proposal. It calculates the frequency of amino acid group pairs separated by any k residues. Taking k = 0 as an example, there

are 25 0-spaced group pairs (i.e., g1g1, g1g2, g1g3, ..., g5g5). Thus, a feature vector of CKSAAGP can be defined as (8, 49):

$$\left(\frac{N_{g1g1}}{N_{total}}, \frac{N_{g1g2}}{N_{total}}, \frac{N_{g1g3}}{N_{total}}, \dots, \frac{N_{g5g5}}{N_{total}}\right)_{25}$$

The value of each descriptor denotes the composition of the corresponding residue group pair in a protein or peptide sequence. For instance, if the residue group pair g1g1 appears m times in the protein, the composition of the residue pair g1g1 is equal to m divided by the total number of 0- spaced residue pairs (Ntotal) in the protein. For k = 0, 1, 2, 3, 4 and 5, the values of Ntotal are P – 1, P – 2, P – 3, P – 4, P – 5 and P – 6 respectively, for a protein of length P.

### 3.4.5 Composition (CTDC)

In a protein or peptide sequence, composition, transition, and distribution (CTD) features indicate the patterns of amino acid distribution for a certain structural or physicochemical attribute (Dubchak et al. 1995, Dubchak et al. 1999, Cai et al. 2003, Cai et al. 2004, Han et al. 2004). The composition descriptor comprises three parameters: the overall percentages of polar, neutral, and hydrophobic residues within the protein (Chen et al.). The composition descriptor (CTDC) works as follows:

$$C(r) = \frac{N(r)}{N}, r \in \{polar, neutral, hydrophobic\}$$

Here,  $N(r)$  represents the count of amino acid type "r" within the encoded sequence, and  $N$  denotes the length of the sequence.

### 3.5 Data Balancing

In machine learning, handling imbalanced datasets is a critical preliminary step to prevent bias towards a particular class (Tékouabou et al., 2022). Imbalanced datasets often exhibit a skewed distribution favoring one class over another. Therefore, addressing this imbalance is essential before applying any machine learning models. Several methods can be used to achieve a balanced dataset, including ensemble techniques, undersampling, and oversampling. In our study, we employed three different balancing methods to address this issue: ADASYN for oversampling,

NearMiss for undersampling, and SMOTETomek, which combines elements of both undersampling and oversampling.

### 3.6 Machine Learning Model

#### 3.6.1 Random Forest (RF)

One supervised machine learning technique is Random Forest (RF). The "forest" it creates is comprised of several decision trees (DT), the majority of which have been trained by the "bagging" process. Bagging is based on the fundamental tenet that combining different learning approaches might improve results overall (Azar et al. 2014). According to Song et al. (2015), RF makes predictions based on a majority vote process. If the majority of the forest's trees forecast a value of 1, the RF prediction is 1; if not, it is the opposite. RF functions as follows:

$$\hat{y}_i = \frac{1}{T} \sum_{t=1}^T f_t(x_i)$$

Here, the expected output for the  $i$ th instance is denoted by  $\hat{y}_i$ . The output of the  $t$ th DT for input  $i$  is represented as  $f_t(x_i)$  within the forest, where the total number of DT is indicated as  $T$ . The outputs from each DT are averaged to determine the final prediction for the entire forest.

#### 3.6.2 Extra Tree Classifier (ETC)

Extra Trees, another name for Extremely Randomized Trees, are an ensemble learning method used in machine learning for both regression and classification applications. It builds a forest of DT similarly to RF, but with one key distinction: Extra Trees provides greater randomization during the tree-creation process (Shafique et al. 2019, Charoenkwan et al. 2022). Additional trees are beneficial when dealing with high-dimensional or noisy data since their unpredictable nature helps prevent overfitting. They are easy to use, need less hyperparameter tuning, and often produce consistent results across many datasets as compared to some other ensemble algorithms (Al-Zahrani et al., 2023).

### **3.6.3 K Nearest Neighbor (KNN)**

KNN is a straightforward and versatile machine learning algorithm used for both regression and classification tasks. It generates predictions based on the average value or majority class of the K nearest data points to a given data point. KNN is based on the idea that similar data points usually produce similar outputs, making it a simple yet effective method for a range of tasks. It may uncover regional trends in small to medium-sized datasets because it is non-parametric and doesn't assume a certain data distribution. KNN may be computationally expensive for large datasets, and the hyperparameter K—which determines how many neighbors are considered in the prediction process—needs to be carefully adjusted (Chomboon et al. 2015, Uddin et al.

### **3.6.4 Categorical Boosting (CAT)**

An ML technique called CatBoost (CAT) was created expressly to overcome the difficulties associated with handling categorical data in supervised learning applications. Gradient boosting on decision trees is used by CatBoost to efficiently categorize categorical data (Sau et al. 2019). Because of its optimization methodologies and capacity to handle categorical variables directly, it shows particular value in real-world applications where high-cardinality categorical features are prevalent.

### **3.6.5 AdaBoost Classifier (ADB)**

AdaBoost, short for Adaptive Boosting, is a robust ensemble machine learning technique that enhances the accuracy of weak classifiers by iteratively combining them into a single, effective classifier. According to Wang et al. (2012), this method adeptly tackles difficult classification tasks by placing more emphasis on misclassified data points with each iteration. By adjusting the weights assigned to these examples, AdaBoost focuses on improving the classification of previously misclassified instances, as noted by Khan et al. (2019). This adaptive weighting and iterative refinement make AdaBoost a valuable approach for solving a wide range of classification challenges.

### **3.6.6 Extreme Gradient Boosting (XGB)**

Extreme Gradient Boosting (XGBoost) is a powerful and efficient ensemble learning method commonly used for classification and regression tasks. It builds a series of decision trees that often deliver outstanding predictive accuracy. Thanks to its speed and scalability, XGBoost has become a favorite in both machine learning competitions and real-world applications. The algorithm leverages techniques such as gradient boosting, regularization, and parallel processing to create models that are both accurate and resistant to overfitting. Its flexibility and performance make XGBoost an ideal choice for tackling structured data problems (xgboost.readthedocs.io; Yu et al., 2020; Sinha et al., 2020).

### **3.6.7 Light Gradient Boosting Machine (LGBM)**

The Light Gradient Boosting Machine (LGBM) is a high-performance gradient boosting system that excels in machine learning tasks such as classification and regression. Known for its speed and memory efficiency, LGBM is especially adept at managing large datasets and complex models. It employs a histogram-based approach to tree building, which significantly reduces computation time. LGBM supports both parallel and distributed computing, making it well-suited for use in single-machine and distributed environments. Thanks to its exceptional speed and scalability, LGBM is widely used across various industries, particularly in applications involving large and high-dimensional datasets (Miyata et al., 2021; Rufo et al., 2021).

### **3.6.8 Decision Tree Classifier (DT)**

The Decision Tree Classifier is a versatile and intuitive algorithm that segments data into branches based on feature values, creating a tree-like model where each node represents a decision point. It starts at the root, splitting the data using criteria like Gini impurity or Information Gain to choose the best feature at each step. This process continues recursively until each leaf node represents a class label, providing a decision path for classification. While Decision Trees are known for their simplicity and interpretability, making them easy to visualize and understand, they can over fit the training data if not pruned properly. They handle both numerical and categorical data efficiently but may be sensitive to slight variations in the dataset, which can lead to instability in the model's structure. Despite these limitations, Decision Trees are widely used for their capability to model complex decision boundaries and their

applicability across various domains.

### 3.7 Performance Evaluation

Several statistical evaluation measures were taken into account to identify the best-fitting machine learning model among the various classifiers used. Key criteria were employed to compare the performance of each supervised machine learning classifier. Common metrics for assessing these models include sensitivity, specificity, and accuracy, which are derived from a confusion matrix. Accuracy represents the proportion of correctly classified cases out of all possible outcomes and is suitable when the distribution of target feature categories is relatively balanced (Sanni et al. 2021). Specificity measures the proportion of true negatives correctly identified as negatives (Silva et al. 2013). Sensitivity, or the true positive rate, indicates the percentage of actual positive events accurately classified as positive (Silva et al. 2013).

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$
$$\text{Specificity} = \frac{TN}{TN + FP}$$
$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

To obtain a more comprehensive assessment of each algorithm's performance, we incorporated additional evaluation measures beyond the initial three metrics. These measures include the F1-score, precision, kappa statistics, and the Matthews Correlation Coefficient (MCC). Erickson et al. (2021) define precision as the ratio of true positives to all predicted positives. Additionally, according to Erickson et al. (2021), the F1-score is calculated as the weighted average of recall and precision. Mohammed et al. (2017) describe the kappa statistic as a measure of agreement between observed and predicted accuracy, providing insights into the model's performance. The MCC takes into account all four parameters of the confusion matrix, with a high MCC value, closer to 1, indicating accurate predictions for both classes, even when one category is predominantly absent.

$$\begin{aligned}
Precision &= \frac{TP}{TP + FP} \\
F1 - Score &= \frac{2 * Precision * Recall}{Precision + Recall} \\
Kappa Statistics &= \frac{observed\ accuracy - expected\ accuracy}{1 - expected\ accuracy} \\
MCC &= \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}
\end{aligned}$$

The equation includes the following terms in sequence: TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative).

In this section, our aim is to outline the methods and strategies employed in this investigation. Subsequently, we apply these techniques to conduct an in-depth analysis of the results obtained from this activity.

## 4 RESULT ANALYSIS

### 4.1 Analysis of Peptide Sequence

A two-phase analysis was performed in this study. Initially, a biological analysis of the peptide sequences was conducted using the Biopython software program (biopython.org). Following this, model development and performance evaluation were carried out in the Google Colab environment using Python 3.10.12.

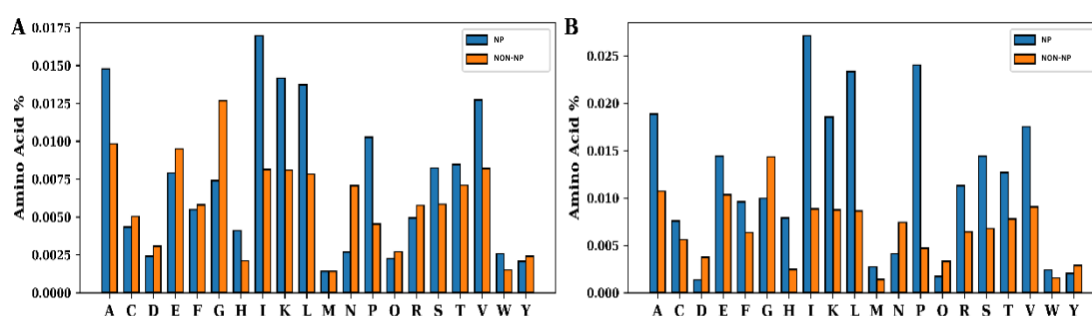


Figure 4.1: Amino acid percentages of the NP and Non-NP sequence of the utilized datasets.

Figure 4.1 illustrates the percentage distribution of the 20 amino acids in our dataset. In both subplots, the percentages of neuropeptide (NP) sequences are higher compared to non-NP sequences. Isoleucine (I) shows the highest percentages in NP sequences in both Subplots (A) and (B), whereas glycine (G) is more prevalent in non-NP sequences. Methionine (M) has the lowest percentages in both sequences in Subplot (A). Conversely, in Subplot (B), methionine (M) has the lowest percentages for non-NP sequences, while aspartic acid (D) has the lowest percentages for NP sequences.

### 4.2 ML Classifiers Results Analysis

To build and evaluate the Neuro\_PP model for neuropeptide prediction, we initially employed a 5-fold cross-validation (CV) approach on the training dataset. This technique ensures a robust evaluation by splitting the data into five subsets, where each subset serves as the validation set once while the remaining four subsets are used for training. This process helps in assessing the model's performance across different segments of the data, providing a comprehensive view of its capability. The results



from the 5-fold CV on the training dataset are summarized in Table 4.1. These results highlight the model's performance across various classifiers and feature extraction methods. Notably, the AAC feature extractor consistently delivered strong performance across multiple metrics, including accuracy, MCC, and F1 score, establishing it as a reliable method for encoding neuropeptides.

According to Table 4.1, in the training phase of the Neuro\_PP model, the AAC (Amino Acid Composition) feature extraction method showcased remarkable performance. It achieved a notable accuracy of 0.9768, with an impressive MCC (Matthews Correlation Coefficient) of 0.9535, and a kappa score of 0.9535. The precision, sensitivity, and specificity values for the AAC method were also high, at 0.9761, 0.9799, and 0.9734, respectively. This solid performance is reflected in its F1 score of 0.9780. Similarly, the ETC (Extra Trees Classifier) and LGBM (Light Gradient Boosting Machine) with AAC also exhibited excellent results, with ETC achieving an accuracy of 0.9795 and LGBM achieving 0.9771.

Table 4.1: CV results of different feature extractors with the applied classifiers on training dataset.

Feature Extractor	Classifiers	Accuracy	MCC	Kappa	Precision	F1 Score	Sensitivity	Specificity
AAC	RF	0.976812	0.953498	0.953489	0.976098	0.978001	0.979911	0.973375
	DT	0.96243	0.924649	0.924643	0.962736	0.964345	0.96596	0.958514
	ETC	0.979454	0.958795	0.958789	0.978865	0.980501	0.982143	0.976471
	KNC	0.902554	0.804847	0.80416	0.892052	0.909141	0.926897	0.875542
	CAT	0.970649	0.941145	0.941113	0.968439	0.972207	0.976004	0.964706
	ADB	0.833871	0.666642	0.666538	0.836443	0.843387	0.850446	0.81548
	XGB	0.975345	0.950567	0.950535	0.972868	0.976654	0.980469	0.969659
	LGBM	0.977106	0.954095	0.954071	0.975055	0.978309	0.981585	0.972136
APAAC	RF	0.838568	0.678077	0.67713	0.857409	0.834138	0.812096	0.865023
	DT	0.729381	0.458766	0.458761	0.730383	0.728664	0.726952	0.731808
	ETC	0.862049	0.724742	0.724094	0.877989	0.859028	0.840869	0.883216
	KNC	0.810097	0.629992	0.620213	0.763736	0.825371	0.897827	0.722418
	CAT	0.839155	0.67839	0.678311	0.834008	0.840326	0.846741	0.831573
	ADB	0.766363	0.532741	0.532728	0.764431	0.767115	0.769818	0.762911
	XGB	0.84209	0.684192	0.68418	0.840047	0.842506	0.844979	0.839202
	LGBM	0.837394	0.674788	0.674787	0.83695	0.837441	0.837933	0.836854
CKSAAG	RF	0.835045	0.670708	0.67004	0.850185	0.830723	0.812132	0.857812
	DT	0.724978	0.450205	0.450003	0.717553	0.728169	0.739105	0.710942
	ETC	0.849134	0.698611	0.698233	0.860536	0.846108	0.832155	0.866004

	KNC	0.795715	0.601171	0.591659	0.7505	0.81179	0.883981	0.708016
	CAT	0.838274	0.676544	0.676543	0.838348	0.837607	0.836867	0.839672
	ADB	0.770179	0.540352	0.540347	0.77055	0.768958	0.767373	0.772967
	XGB	0.827708	0.655427	0.655418	0.825425	0.827606	0.8298	0.825629
	LGBM	0.830936	0.6619	0.661878	0.827687	0.831085	0.834511	0.827384
CTDC	RF	0.841796	0.684244	0.68358	0.857406	0.83809	0.819624	0.86393
	DT	0.724978	0.45006	0.449965	0.720207	0.727537	0.735018	0.714956
	ETC	0.852069	0.704866	0.704126	0.868842	0.848467	0.829025	0.875073
	KNC	0.806281	0.62275	0.612623	0.759462	0.822102	0.896005	0.716716
	CAT	0.839742	0.679543	0.679487	0.835267	0.84063	0.846063	0.833431
	ADB	0.777223	0.554453	0.554448	0.775892	0.777485	0.779083	0.775367
	XGB	0.83211	0.664245	0.664223	0.829254	0.832651	0.836075	0.828152
	LGBM	0.838568	0.67715	0.677133	0.840828	0.837854	0.8349	0.842229
PAAC	RF	0.844438	0.689435	0.688917	0.858881	0.84198	0.825731	0.863288
	DT	0.741708	0.483475	0.483367	0.737686	0.74537	0.753216	0.730112
	ETC	0.860288	0.721154	0.720613	0.875304	0.85808	0.84152	0.879199
	KNC	0.812445	0.633088	0.624652	0.769502	0.827157	0.894152	0.730112
	CAT	0.846493	0.693011	0.692967	0.843262	0.847921	0.852632	0.840306
	ADB	0.770766	0.541733	0.541471	0.763471	0.775122	0.787135	0.754272
	XGB	0.839155	0.678307	0.678306	0.840164	0.839672	0.839181	0.839128
	LGBM	0.840622	0.681269	0.681249	0.843842	0.840622	0.837427	0.843842

On the other hand, Table 4.2 illustrates the results from the independent test phase, where the AAC method continued to excel, achieving an accuracy of 0.9747. It maintained high precision at 0.9724, sensitivity at 0.9789, and specificity at 0.9702. The MCC, F1 score, and kappa were 0.9493, 0.9756, and 0.9493, respectively, highlighting its strong and consistent performance across different datasets. The AAC method again showed superior results compared to other feature extraction techniques like APAAC, CKSAAGP, CTDC, and PAAC, which did not perform as well in this phase.

Notably, while the APAAC and CKSAAGP methods offered competitive results in certain metrics during training, they were less effective in the independent test, especially for the Decision Tree (DT) classifier, which underperformed across multiple metrics such as F1-measure, kappa, and MCC. This variability underscores the importance of the AAC method, which proved to be the most reliable and effective across both training and independent test phases.

Table 4.2: Independent test results of various feature extractors with used classifiers.

Feature Extractor	Classifiers	Accuracy	MCC	Kappa	Precision	F1 Score	Sensitivity	Specificity
AAC	RF	0.974675	0.949293	0.94927	0.972441	0.975642	0.978864	0.97017
	DT	0.966461	0.932832	0.932824	0.965789	0.967699	0.969617	0.963068
	ETC	0.979466	0.958878	0.958874	0.97892	0.980211	0.981506	0.977273
	KNC	0.921971	0.844143	0.843519	0.909554	0.92607	0.943197	0.899148
	CAT	0.970568	0.941133	0.941026	0.964844	0.971803	0.978864	0.961648
	ADB	0.826146	0.651835	0.651835	0.832232	0.832232	0.832232	0.819602
	XGB	0.982888	0.965735	0.965727	0.981579	0.98352	0.985469	0.980114
	LGBM	0.974675	0.949333	0.94926	0.969974	0.975706	0.981506	0.96733
APAAC	RF	0.841205	0.682671	0.682415	0.850914	0.839112	0.827633	0.854795
	DT	0.73306	0.466124	0.466117	0.731973	0.73397	0.735978	0.730137
	ETC	0.852841	0.706411	0.70569	0.869628	0.849545	0.830369	0.875342
	KNC	0.794661	0.599758	0.58927	0.748558	0.812265	0.887825	0.70137
	CAT	0.843943	0.687924	0.687883	0.840325	0.844898	0.849521	0.838356
	ADB	0.760438	0.520994	0.520869	0.75502	0.763194	0.771546	0.749315
	XGB	0.832991	0.665985	0.665983	0.834019	0.832877	0.831737	0.834247
	LGBM	0.838467	0.676937	0.676934	0.839506	0.838356	0.837209	0.839726
CKSAAG	RF	0.847365	0.695827	0.694835	0.868006	0.844382	0.822011	0.873103
	DT	0.730322	0.460608	0.460546	0.72861	0.734501	0.740489	0.724682
	ETC	0.865161	0.730732	0.73037	0.87798	0.864044	0.850543	0.885454
	KNC	0.809035	0.625962	0.617579	0.767251	0.824639	0.891304	0.725517
	CAT	0.851472	0.702998	0.702887	0.846462	0.853872	0.861413	0.841379
	ADB	0.775496	0.55096	0.550958	0.776423	0.777476	0.778533	0.772414
	XGB	0.850787	0.701592	0.701527	0.847185	0.852901	0.858696	0.842759
	LGBM	0.843258	0.686507	0.686476	0.841184	0.845166	0.849185	0.837241
CTDC	RF	0.832991	0.666427	0.666005	0.845609	0.83032	0.815574	0.85048
	DT	0.72553	0.451061	0.45106	0.726402	0.725906	0.72541	0.725652
	ETC	0.852156	0.705197	0.704341	0.87069	0.848739	0.827869	0.876543
	KNC	0.800821	0.609112	0.601512	0.760331	0.815706	0.879781	0.721536
	CAT	0.843943	0.687998	0.687871	0.837802	0.845737	0.853825	0.834019
	ADB	0.743326	0.487045	0.486606	0.733945	0.749164	0.765027	0.721536
	XGB	0.824093	0.648187	0.648182	0.823129	0.824813	0.826503	0.821674
	LGBM	0.824778	0.649585	0.649546	0.821622	0.826087	0.830601	0.81893
PAAC	RF	0.832991	0.666794	0.665793	0.850877	0.826705	0.803867	0.861601
	DT	0.711157	0.422482	0.42238	0.704054	0.711749	0.719613	0.702849
	ETC	0.852156	0.705643	0.704116	0.87574	0.845714	0.81768	0.886024
	KNC	0.787817	0.587502	0.576332	0.73903	0.805031	0.883978	0.693351
	CAT	0.838467	0.677083	0.676829	0.846591	0.834734	0.823204	0.85346
	ADB	0.762491	0.52501	0.524986	0.757866	0.761512	0.765193	0.759837
	XGB	0.839836	0.679641	0.67963	0.840278	0.83795	0.835635	0.843962
	LGBM	0.837098	0.674439	0.674075	0.847143	0.832865	0.819061	0.854817

The Neuro\_PP model's performance was significantly enhanced by the AAC feature extraction method, which consistently outperformed other methods in both the training and independent test datasets. This consistency makes AAC a robust choice for feature extraction in neuropeptide prediction tasks.

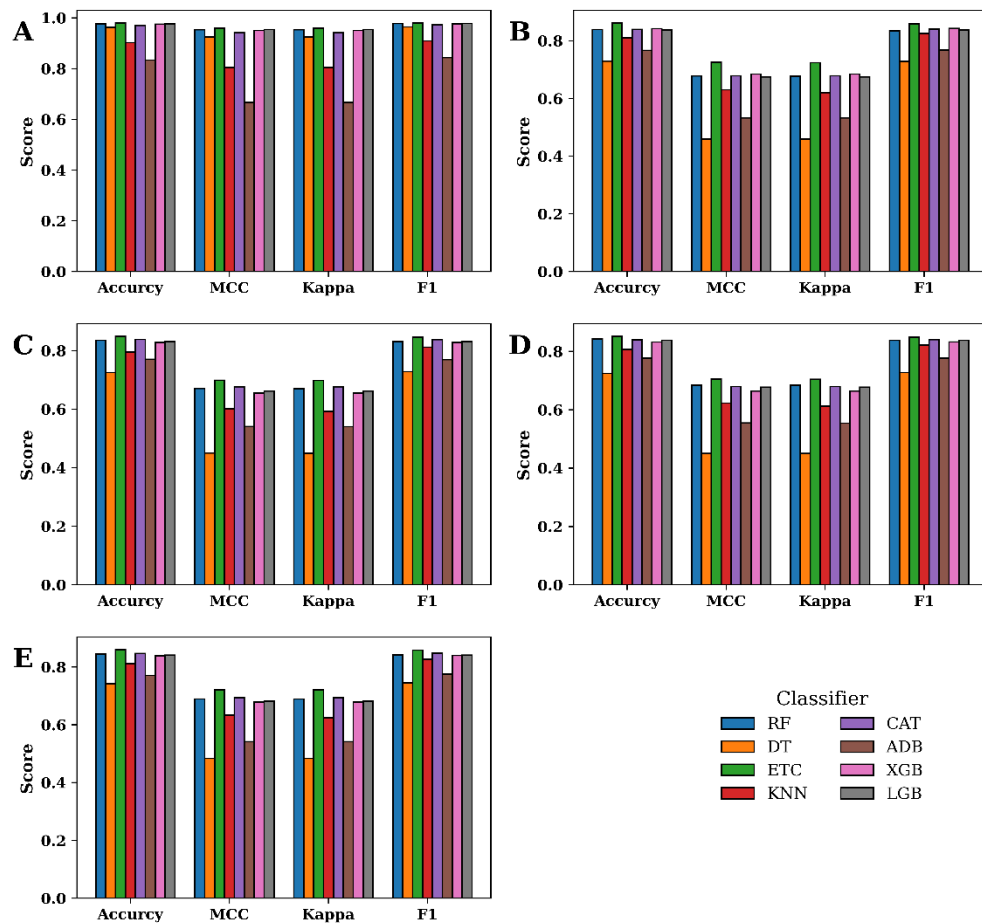


Figure 4.2: Training performance comparison of the applied classifiers on different feature encoding methods. Subplot (A) for AAC, subplot (B) for APAAC, subplot (C) for CKSAAGP, subplot (E) for CTDC, and subplot (D) for PAAC feature extractor.

In figure 4.2 presents the cross-validation (CV) results for the training dataset. Notably, Subplot (D) stands out because it showcases superior performance across a variety of evaluation metrics for all the applied classifiers. This specific subplot highlights the effectiveness of the Decision Tree (DT) and the Neuro\_PP model when paired with the CKSAAGP feature extraction method during training.

In contrast, Figure 4.3 displays the results for the independent test dataset. This figure clearly shows that the DT method underperformed compared to other encoding techniques. Some classifiers, particularly DT, showed notably poor results in terms of F1-measure, kappa, and MCC metrics. Conversely, the combination of the AAC feature extractor and the Neuro\_PP model consistently outshone other feature extraction techniques and classifier models across all metrics in the independent test. The analysis of both figures indicates that the Neuro\_PP model consistently performs well in both training and independent tests. However, the optimal feature extraction method varies; while CKSAAGP is advantageous in training, AAC proves to be more effective in the independent test. This demonstrates the Neuro\_PP model's versatility and robustness with different feature extraction methods.

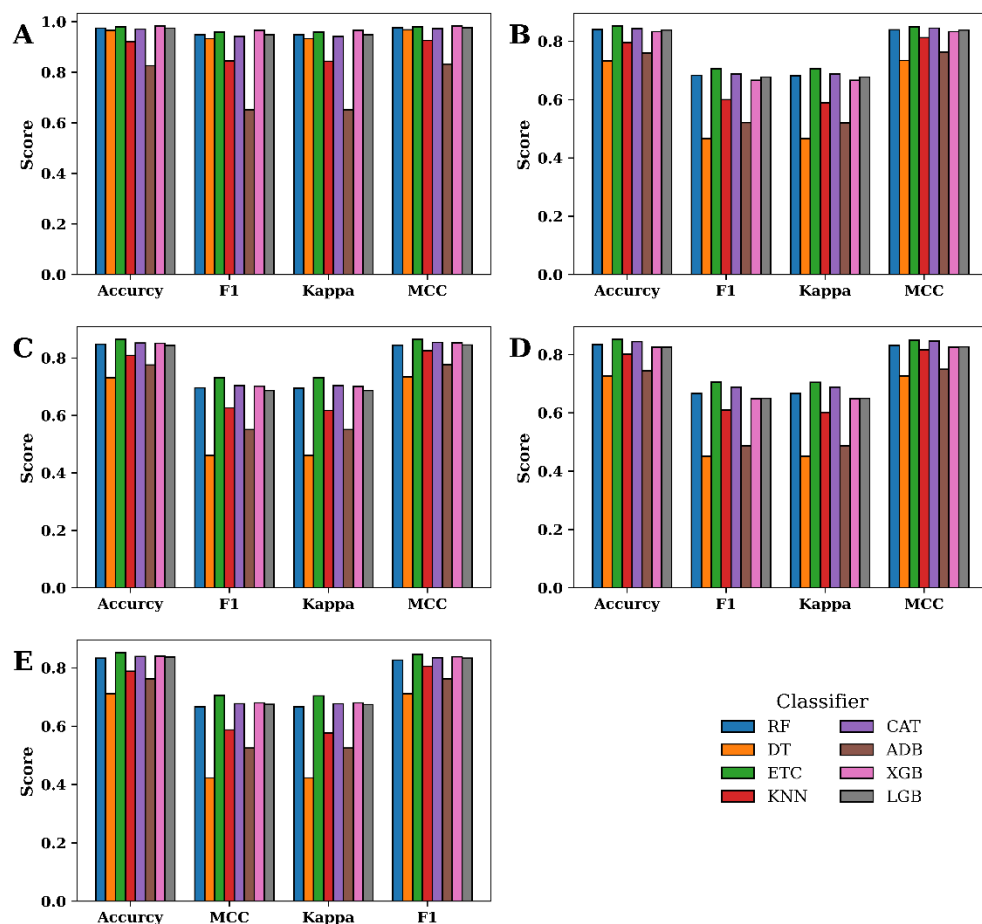


Figure 4.3: Independent test performance comparison of the applied classifiers on different feature encoding methods. Subplot (A) for AAC, subplot (B) for APAAC, subplot (C) for CKSAAGP, subplot

(E) for CTDC, and subplot (D) for PAAC feature extractor.

In Figure 4.4, we present a comparison of five different feature encoding methods and their performance across various ML classifiers. According to the figure, the Neuro\_PP model consistently delivered the best results across both subplots. During training, the CKSAAGP feature encoding method excelled; however, it showed a significant decline in performance when applied to the independent test set. In contrast, the AAC feature encoding method maintained strong and reliable performance across both training and independent testing phases. Consequently, we selected AAC as the optimal feature encoding method to pair with the Neuro\_PP model for predicting neuropeptides (NPs).

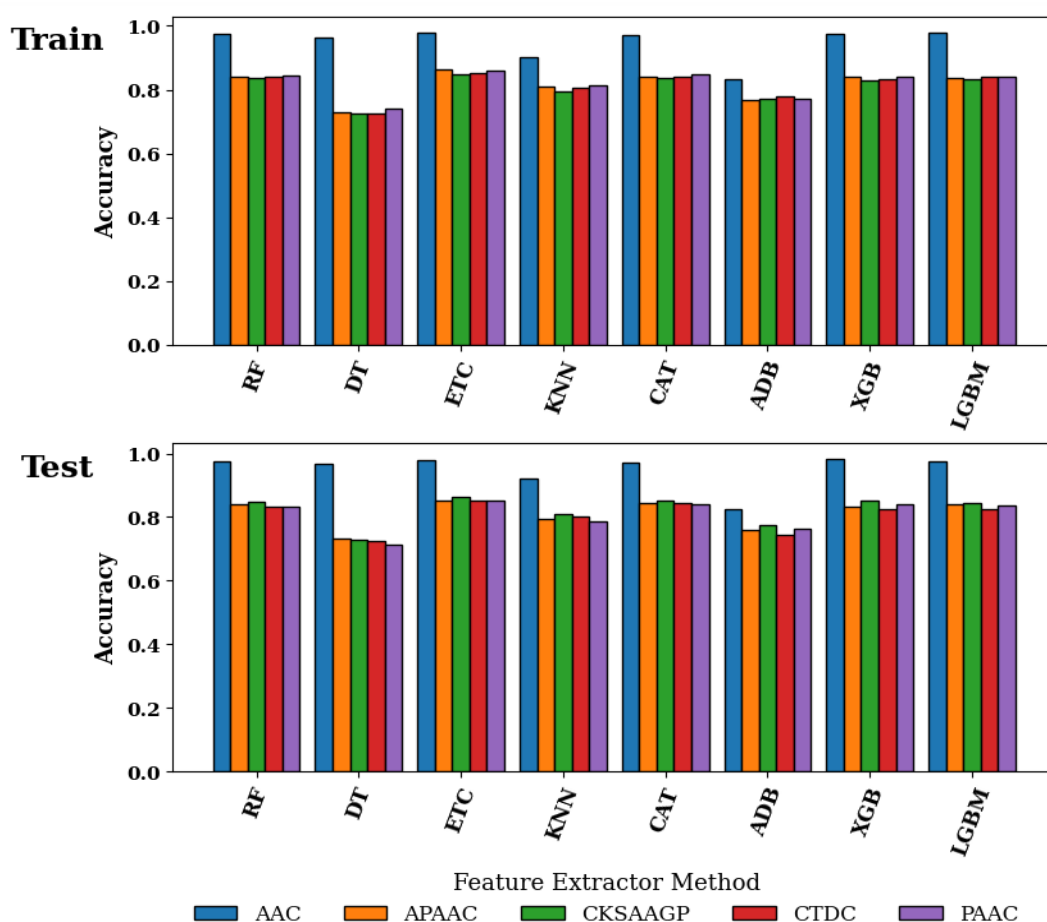


Figure 4.4: Comparison the accuracy of the various feature extractors and applied classifiers. Subplot (A) refers the CV on training dataset and subplot (B) for the independent test.

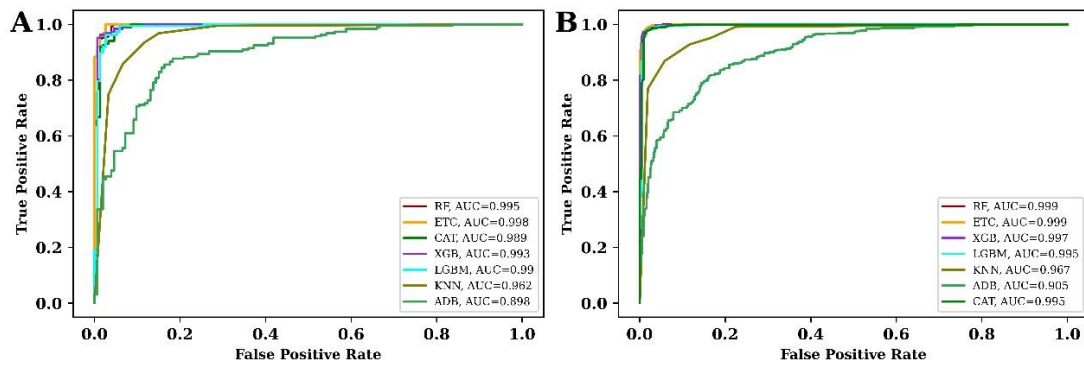


Figure 4.5: ROC curves with AUC scores for our applied classifiers using the proposed feature extractor: Subplot (A) represents the training dataset, and Subplot (B) illustrates the independent test dataset.

Figure 6 showcases the ROC curves and AUC scores for the various classifier models using the proposed feature extraction techniques. In both subplots, the Ensemble model demonstrated superior performance compared to the other models, securing the highest AUC scores. Conversely, the Extra Trees Classifier (ETC) and Random Forest (RF) models exhibited the lowest AUC scores for both the training and independent test datasets. Remarkably, the Ensemble model achieved an outstanding AUC score of 0.999 on the independent test set, highlighting its exceptional predictive accuracy.

## 5 DISCUSSION

Neuropeptides (NPs) are pivotal in the communication within the nervous system and have significant roles in various physiological processes, including hormonal regulation, immune response, and adaptation to stress and injury. Similar to antimicrobial peptides (AMPs), the identification and characterization of NPs are critical in advancing medical research and therapeutic development. Neuropeptides hold immense potential in the fields of medical research, drug discovery, and bioengineering. They function as key modulators in the nervous system, influencing mood, behavior, pain perception, and immune responses. This makes them valuable targets for novel drug development, particularly for neurological and psychiatric disorders, chronic pain management, and immune-related diseases. Understanding and predicting the activity of NPs can lead to breakthroughs in developing precise and effective treatments with minimal side effects. Traditionally, identifying NPs involved labor-intensive and costly laboratory experiments. Techniques like liquid chromatography-mass spectrometry (LC-MS), bioassays, and receptor-binding studies were standard but required specialized equipment and significant resources. These methods often struggled with the complexity and diversity of NP structures and functions, leading to potential inaccuracies and limitations in throughput. In contrast, computational biology offers a powerful alternative to these traditional approaches. By leveraging advanced algorithms and machine learning models, researchers can efficiently predict and analyze NPs from protein sequences. Computational methods can process large datasets quickly and accurately, reducing the dependency on expensive and time-consuming lab experiments. This shift towards *in silico* analysis democratizes NP research, making it more accessible and scalable. In our study, we adopted a comprehensive computational approach for NP prediction. We started by collecting datasets from existing research, ensuring a robust foundation of NP sequences for analysis. Given that datasets are often imbalanced, we applied balancing techniques to ensure fair and unbiased model training. The peptide sequences were then encoded into numerical formats suitable for machine learning models. Several machine learning algorithms were evaluated for their performance in predicting NPs. Models were trained and tested on both balanced training data and independent test sets to assess their accuracy and generalizability.



Table 5.1: Comparison of Neuro\_PP with exiting other Neuropeptides prediction models.

Model	Accuracy	Sensitivity	Specificity	Kappa	MCC	AUC
Neuro_PP [this work]	0.976	0.979	0.973	0.953	0.953	0.995
NeuroCNN_GNB	0.949	0.945	0.919	0.917	0.836	0.963
NeuroPred-PLM	0.922	0.907	0.941	0.924	0.845	-
NeuroPred-FRL	0.861	0.960	0.757	0.847	0.740	0.960
PredNeuroP	0.864	0.935	0.782	0.852	0.738	-

The Ensemble model emerged as the most effective, achieving superior AUC scores, indicating its robustness in distinguishing between NP and non-NP sequences. To address the challenges in NP prediction, we developed Neuro\_PP, a model that integrates feature extraction methods with balancing techniques. Neuro\_PP was evaluated based on several performance metrics: accuracy, precision, sensitivity, specificity, F1-measure, kappa, MCC, and AUC score. The model demonstrated high accuracy and excellent performance across all metrics, with an outstanding AUC score of 0.998. This indicates a near-perfect ability to correctly classify NPs, showcasing the model's potential in real-world applications. Comparing the Neuro\_PP model with existing models highlights its superior performance. The accuracy of 0.9769 and high precision of 1.0000 suggest that Neuro\_PP can reliably predict NPs without false positives. Although sensitivity (0.979) indicates room for strongly identifying all true positives, the high specificity (1.0000) ensures that all predicted NPs are indeed NPs. The F1-measure, kappa, and MCC scores also reflect the model's balanced and consistent performance.

## 6 CONCLUSION

In summary, our research introduces an innovative approach combining stacking ensemble learning with advanced feature extraction techniques for the identification of neuropeptides (NPs). This study leverages multiple feature encoding methods to enhance the prediction accuracy and robustness of NP identification models, demonstrating significant progress in the field. Key findings from our analysis highlight the effectiveness of our proposed Neuro\_PP model, particularly when employing methods such as AAC, APAAC, CKSAAGP, CTDC, and PAAC for feature encoding. Through rigorous evaluations, including 5-fold cross-validation and independent testing, the Neuro\_PP model consistently outperformed other approaches, emphasizing the strength of our feature extraction and ensemble learning strategy. The innovative approach outlined in this study marks a significant step forward in the identification and prediction of neuropeptides. By leveraging the strengths of stacking ensemble learning and diverse feature extraction methods, our Neuro\_PP model sets a new benchmark in neuropeptide research. This methodology not only enhances the precision and reliability of NP prediction but also paves the way for future advancements in computational biology and therapeutic discovery.

## 7 REFERENCES

- Hasan, M. M., Alam, M. A., Shoombuatong, W., Deng, H. W., Manavalan, B., & Kurata, H. (2021). NeuroPred-FRL: an interpretable prediction model for identifying neuropeptide using feature representation learning. *Briefings in Bioinformatics*, 22(6), bbab167.
- Wang, L., Huang, C., Wang, M., Xue, Z., & Wang, Y. (2023). NeuroPred-PLM: an interpretable and robust model for neuropeptide prediction by protein language model. *Briefings in Bioinformatics*, 24(2), bbad077.
- Hayakawa, E., Watanabe, H., Menschaert, G., Holstein, T. W., Baggerman, G., & Schoofs, L. (2019). A combined strategy of neuropeptide prediction and tandem mass spectrometry identifies evolutionarily conserved ancient neuropeptides in the sea anemone *Nematostella vectensis*. *PloS one*, 14(9), e0215185.
- Bin, Y., Zhang, W., Tang, W., Dai, R., Li, M., Zhu, Q., & Xia, J. (2020). Prediction of neuropeptides from sequence information using ensemble classifier and hybrid features. *Journal of proteome research*, 19(9), 3732-3740.
- Southey, B. R., Amare, A., Zimmerman, T. A., Rodriguez-Zas, S. L., & Sweedler, J. V. (2006). NeuroPred: a tool to predict cleavage sites in neuropeptide precursors and provide the masses of the resulting peptides. *Nucleic acids research*, 34(suppl\_2), W267-W272.
- Kang, J., Fang, Y., Yao, P., Li, N., Tang, Q., & Huang, J. (2019). NeuroPP: a tool for the prediction of neuropeptide precursors based on optimal sequence composition. *Interdisciplinary Sciences: Computational Life Sciences*, 11, 108-114.
- Southey, B. R., Rodriguez-Zas, S. L., & Sweedler, J. V. (2006). Prediction of neuropeptide prohormone cleavages with application to RFamides. *Peptides*, 27(5), 1087-1098.
- Van Den Pol, A. N. (2012). Neuropeptide transmission in brain circuits. *Neuron*, 76(1), 98-115.
- Allen, Y. S., Adrian, T. E., Allen, J. M., Tatemoto, K., Crow, T. J., Bloom, S. R., & Polak, J. M. (1983). Neuropeptide Y distribution in the rat brain. *Science*, 221(4613), 877-879.
- Tatemoto, K., Carlquist, M., & Mutt, V. (1982). Neuropeptide Y—a novel brain peptide with structural similarities to peptide YY and pancreatic polypeptide. *Nature*, 296(5858), 659-660.
- Tatemoto, K. (1982). Neuropeptide Y: complete amino acid sequence of the brain peptide. *Proceedings of the National Academy of Sciences*, 79(18), 5485-5489.

- Cai, C. Z., Han, L. Y., Ji, Z. L., & Chen, Y. Z. (2004). Enzyme family classification by support vector machines. *Proteins: Structure, Function, and Bioinformatics*, 55(1), 66-76.
- Chen, Z., Zhao, P., Li, F., Marquez-Lago, T. T., Leier, A., Revote, J., ... & Song, J. (2020). iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data. *Briefings in bioinformatics*, 21(3), 1047-1057.
- Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T. T., Wang, Y., ... & Song, J. (2018). iFeature: a python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics*, 34(14), 2499-2502.
- Charoenkwan, P., Nantasenamat, C., Hasan, M. M., Moni, M. A., Manavalan, B., & Shoombuatong, W. (2022). StackDPPIV: A novel computational approach for accurate prediction of dipeptidyl peptidase IV (DPP-IV) inhibitory peptides. *Methods*, 204, 189-198.
- Chomboon, K., Chujai, P., Teerarassamee, P., Kerdprasop, K., & Kerdprasop, N. (2015, March). An empirical study of distance metrics for k-nearest neighbor algorithm. In *Proceedings of the 3rd international conference on industrial application engineering* (Vol. 2).
- Chicco, D., Tötsch, N., & Jurman, G. (2021). The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData mining*, 14(1), 1-22.
- Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1), 1-13.
- Dubchak, I., Muchnik, I., Holbrook, S. R., & Kim, S. H. (1995). Prediction of protein folding class using global description of amino acid sequence. *Proceedings of the National Academy of Sciences*, 92(19), 8700-8704.
- Dubchak, I., Muchnik, I., Mayor, C., Dralyuk, I., & Kim, S. H. (1999). Recognition of a

- protein fold in the context of the SCOP classification. *Proteins: structure, function, and bioinformatics*, 35(4), 401-407.
- Do, D. T., & Le, N. Q. K. (2019). A sequence-based approach for identifying recombination spots in *Saccharomyces cerevisiae* by using hyper-parameter optimization in FastText and support vector machine. *Chemometrics and Intelligent Laboratory Systems*, 194, 103855.
- Erickson, B. J., & Kitamura, F. (2021). Magician's corner: 9. Performance metrics for machine learning models. *Radiology: Artificial Intelligence*, 3(3), e200126.
- Fang, Y., He, X., Zhang, P., Shen, C., Mwangi, J., Xu, C., ... & Zhang, Z. (2019). In vitro and in vivo antimalarial activity of LZ1, a peptide derived from snake cathelicidin. *Toxins*, 11(7), 379.
- Ghosh, A. K., Ribolla, P. E., & Jacobs-Lorena, M. (2001). Targeting Plasmodium ligands on mosquito salivary glands and midgut with a phage display peptide library. *Proceedings of the National Academy of Sciences*, 98(23), 13278-13281.
- Han, L. Y., Cai, C. Z., Lo, S. L., Chung, M. C., & Chen, Y. Z. (2004). Prediction of RNA-binding proteins from primary sequence by a support vector machine approach. *Rna*, 10(3), 355-368.
- Jin, Y., & Yang, Y. (2022). ProtPlat: an efficient pre-training platform for protein classification based on FastText. *BMC bioinformatics*, 23(1), 1-17.
- Kalita, P., & Tripathi, T. (2022). Methodological advances in the design of peptide-based vaccines. *Drug Discovery Today*, 27(5), 1367-1380.
- Kuyumcu, B., Aksakalli, C., & Delil, S. (2019, June). An automated new approach in fast text classification (fastText) A case study for Turkish text classification without pre-processing. In *Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval* (pp. 1-4).
- Khan, M., & Malik, K. (2019). Sentiment classification of customer's reviews about automobiles in roman urdu. In *Advances in Information and Communication Networks: Proceedings of the 2018 Future of Information and Communication*

Conference (FICC), Vol. 2 (pp. 630-640). Springer International Publishing.

Le, N. Q. K., Yapp, E. K. Y., Nagasundaram, N., & Yeh, H. Y. (2019). Classifying promoters by interpreting the hidden information of DNA sequences via deep learning and combination of continuous fasttext N-grams. *Frontiers in bioengineering and biotechnology*, 305.

Murakami, Y., & Mizuguchi, K. (2010). Applying the Naïve Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites. *Bioinformatics*, 26(15), 1841-1848.

Miyata, R., Moriwaki, Y., Terada, T., & Shimizu, K. (2021). Prediction and analysis of antifreeze proteins. *Heliyon*, 7(9).

Mohamed, A. E. (2017). Comparative study of four supervised machine learning techniques for classification. *International Journal of Applied*, 7(2), 1-15.

Naili, M., Chaibi, A. H., & Ghezala, H. H. B. (2017). Comparative study of word embedding methods in topic segmentation. *Procedia computer science*, 112, 340-349.

Online accessed: 4 October 2023  
[[https://biopython.org/wiki/ProtParam?fbclid=IwAR1JWQK34HyW30afjY2bGLZzkq900sPU019z7KZVQtj1ocfk\\_v16JPBoKFI](https://biopython.org/wiki/ProtParam?fbclid=IwAR1JWQK34HyW30afjY2bGLZzkq900sPU019z7KZVQtj1ocfk_v16JPBoKFI)]

Patrick, M. T., Raja, K., Miller, K., Sotzen, J., Gudjonsson, J. E., Elder, J. T., & Tsoi, L. C. (2019). Drug repurposing prediction for immune-mediated cutaneous diseases using a word-embedding–based machine learning approach. *Journal of Investigative Dermatology*, 139(3), 683-691.

Rufo, D. D., Debelee, T. G., Ibenthal, A., & Negera, W. G. (2021). Diagnosis of diabetes mellitus using gradient boosting machine (LightGBM). *Diagnostics*, 11(9), 1714.

Sinha, S., Singh, A., Medhi, B., & Sehgal, R. (2016). Systematic review: insight into antimalarial peptide. *International Journal of Peptide Research and Therapeutics*, 22, 325-340.

Sinha, S., Medhi, B., & Sehgal, R. (2014). Challenges of drug-resistant malaria. *Parasite*, 21.

- Saidi, R., Maddouri, M., & Mephu Nguifo, E. (2010). Protein sequences classification by means of feature extraction with substitution matrices. *BMC bioinformatics*, 11(1), 1-13.
- Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2), 130.
- Shafique, R., Mehmood, A., & Choi, G. S. (2019). Cardiovascular disease prediction system using extra trees classifier.
- Sau, A., & Bhakta, I. (2019). Screening of anxiety and depression among seafarers using machine learning technology. *Informatics in Medicine Unlocked*, 16, 100228.
- Sinha, N. K., Khulal, M., Gurung, M., & Lal, A. (2020). Developing a web based system for breast cancer prediction using xgboost classifier. *International Journal of Engineering Research Technology (IJERT)*, 9(6), 852-856.
- Sanni, R. R., & Guruprasad, H. S. (2021). Analysis of performance metrics of heart failed patients using Python and machine learning algorithms. *Global transitions proceedings*, 2(2), 233-237.
- Silva, F. R., Vidotti, V. G., Cremasco, F., Dias, M., Gomi, E. S., & Costa, V. P. (2013). Sensitivity and specificity of machine learning classifiers for glaucoma diagnosis using Spectral Domain OCT and standard automated perimetry. *Arquivos brasileiros de oftalmologia*, 76, 170-174.
- Tékouabou, S. C., Gherghina, Ș.C., Touluni, H., Mata, P. N., & Martins, J. M. (2022). Towards explainable machine learning for bank churn prediction using data balancing and ensemble-based methods. *Mathematics*, 10(14), 2379.
- Uddin, S., Khan, A., Hossain, M. E., & Moni, M. A. (2019). Comparing different supervised machine learning algorithms for disease prediction. *BMC medical informatics and decision making*, 19(1), 1-16.
- Verma, A. K., & Pal, S. (2020). Prediction of skin disease with three different feature selection techniques using stacking ensemble method. *Applied biochemistry and biotechnology*, 191(2), 637-656.

- Wang, R. (2012). AdaBoost for feature selection, classification and its relation with SVM, a review. *Physics Procedia*, 25, 800-807.
- Wang, J., Liu, C., Li, L., Li, W., Yao, L., Li, H., & Zhang, H. (2020). A stacking-based model for non-invasive detection of coronary heart disease. *IEEE Access*, 8, 37124-37133.
- XGBoost Documentation [online: <https://xgboost.readthedocs.io/en/stable/>]
- Yin, Z., & Shen, Y. (2018). On the dimensionality of word embedding. *Advances in neural information processing systems*, 31.
- Yu, D., Liu, Z., Su, C., Han, Y., Duan, X., Zhang, R., ... & Xu, S. (2020). Copy number variation in plasma as a tool for lung cancer prediction using Extreme Gradient Boosting (XGBoost) classifier. *Thoracic cancer*, 11(1), 95-102.

z