Thesis title: StackIL13: a stacking ensemble model for the prediction of IL-13 inducing peptides

Submitted by

Md Rajib Mia

ID: 221-44-244

Department of Software Engineering

Daffodil International University

Supervised by

Dr Imran Mahmud

Associate Professor & Head

Department of Software Engineering

Daffodil International University

This Thesis paper has been submitted in fulfillment of the requirements for the Degree of Master's in Software Engineering (Major in Data Science).

Date of Submission: 18 August 2023

APPROVAL

This thesis titled on "A stacking ensemble model for the prediction of IL-13 inducing peptides", submitted by Md Rajib Mia, ID: 221-44-244 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 Masters of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Daffodil International University

Fazla Elahe 18.08.23

Internal Examiner 1

Dr. Md. Fazla Elahe
Assistant Professor and Associate Head
Department of Software Engineering
Daffodil International University

Internal Examiner 2

Afsana Begum
Assistant Professor
Department of Software Engineering
Daffodil International University

External Examiner

Dr. Md. Sazzadur Rahman,
Associate Professor
Institute of Information Technology
Jahangirnagar University

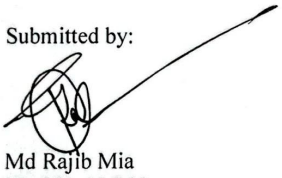
DECLARATION

I announce that I am rendering this study document under Dr. Imran Mahmud, Associate Professor & Head, Department of Software Engineering, Daffodil International University. I therefore, state that this work or any portion of it was not proposed here therefore for Master's degree or any graduation.

Supervised by:

Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Daffodil International University

Submitted by:



Md Rajib Mia
ID: 221-44-244
Department of Software Engineering
Daffodil International University

ACKNOWLEDGEMENT

The study I have done is only by the inspiration of gaining knowledge and learning more. The study is based on the state-of-the-art Machine Learning approach to identify Interleukin-13, Anti Inflammatory Peptides. First of all, I would like to thank the Almighty who has clearly guided me and given me the knowledge to learn and do things that are right. Without His help, this study could not have become a reality. Secondly, my parents, whom I am extremely indebted to for bringing me to where I am now. Then I would like to sincerely thank **Prof. Dr. Imram Mahmud**, Head of the Department of Software Engineering. Then, to all the respected teachers who taught me throughout my journey of learning. I am grateful that I got them as my teachers and guidance.Again, I am deeply grateful to my research supervisor, **Prof. Dr. Imram Mahmud**, for his guidance and support throughout this research. His expertise and knowledge were invaluable to me, and I am truly grateful for his willingness to share them with me.

Finally, I would like to thank my batch mates, members of DIU for their kind co- operation and consolation that helped me to achieve this goal.

Abstract

Inflammatory mediators play an important function in a variety of disorders. IL-13 tightly regulates immune responses, notably those associated with allergies and inflammatory reactions. Predicting interleukin-13 (IL-13) activity is critical since it can be used to identify those who are more likely to develop IL-13-driven illnesses such as asthma and atopic dermatitis. Because the major goal of this study is to construct an accurate Interleukin-13 prediction model utilizing ensemble machine learning methods, we present an enhanced prediction of IL-13-inducing peptides here. The positive and negative datasets were collected from a recent study (IL13Pred), and feature extraction was performed using the ILearnplus package. We used the Best Peptide Sequence Extractor and reported the results of various techniques individually. The data collection was unbalanced, therefore we used the Adasyn Algorithm to balance it. For feature selection, modern feature engineering approaches (Recursive Shapley Value) were used.The results show that when utilizing our StackingClassifier, specific feature sets as CKSAAP, DPC, CTDC, and CTraid may be used to accurately classify data.Our StackingClassifier has improved in terms of accuracy, AUC, and MCC value. Machine learning techniques for interleukin-13 prediction contribute to a better knowledge of IL-13 and its possible consequences in healthcare. It improves the accuracy and reliability of Interleukin-13 prediction, allowing for more informed medical decisions to be made for better patient care and treatment outcomes. This study's successful conclusion increases our understanding of IL-13 prediction while also highlighting the potential of machine learning technologies in addressing complicated biological challenges.

Table of Contents

CHAPTER ONE

1.1 Background

Interleukin-13 (IL-13) is a cytokine of big significance in bioinformatics research. Bioinformatics offers a range of computational equipment and tactics to reading IL-13, which include gene and protein sequence analysis, shape and characteristic prediction, correlation detection, and discovery of correlations. its interplay with different molecules. This finds out about affords treasured records on the relationship between IL-13 and a variety of diseases, evolutionary components and organic processes [7].

In the subject of bioinformatics, gene sequencing performs an indispensable function in IL-13 research. The researchers used pc equipment and algorithms to look at the DNA or RNA sequences that code for IL-13. Sequence alignment strategies permit examining IL-13 sequences between extraordinary species, perceive conserved regions, and predict doable practical elements or regulatory motifs [8]. These analyzes furnish treasured insights into the evolutionary conservation and genetic law of IL-13.

In addition to sequence analysis, bioinformatics researchers are focusing on predicting the three-D shape of the IL-13 protein. Techniques such as homology modeling make it less difficult to infer the shape of a protein with the aid of evaluating it with recognized protein structures. Molecular dynamics modeling lets in simulation in silico kinetics and folding of IL-13. These expected constructions furnish data about practical domains, ligand binding sites, and attainable interplay interfaces with different proteins.

Bioinformatics additionally performs a necessary position in grasping the interactions of IL-13 with different molecules. The researchers' aim used to be to pick out practicable receptors or signaling pathways worried in IL-13-mediated responses and to predict IL-13 binding websites and its affinity. with accomplice molecules. Molecular binding methods enable modeling of IL-13 binding to different receptors or molecules of interest. Molecular dynamics simulations in addition elucidate the dynamics and balance of these interactions, offering perception into the underlying molecular mechanisms. Bioinformatics lookup on IL-13 extends to its involvement in a number of ailments and organic processes. Through genomic evaluation or large-scale transcriptome information sets, researchers can discover genes or pathways worried in IL-13 in pathological conditions. The integration of various datasets, which include gene expression profiles, protein-protein interplay networks, and genetic variants, helps to perceive the underlying mechanisms concerned in IL-13-associated diseases. such as asthma, allergic rhinitis or inflammatory bowel disease. This research makes a contribution to a higher appreciation of the position of IL-13 in pathogenesis and open doable possibilities for therapeutic interventions.

In summary, bioinformatics research of IL-13 consists of gene and protein sequencing, shape and characteristic prediction, interplay research with different molecules, and involvement of IL-13 in illnesses and disorders. organic process. The utility of computational techniques and equipment in these research presents a complete appreciation of the underlying molecular mechanisms of IL-13 and its practicable as a therapeutic goal for these disorders. immune issues and more than a few infections.

1.2 Motivation of the research

There are numerous factors that use bioinformatics methods that are expecting interleukin-thirteen (IL-13) behavior and residences inside the discipline of Machine Learning. IL-13 is a cytokine that plays a crucial role in numerous biological processes which includes immune reaction, irritation and tissue restore. knowledge of the complicated interactions and capabilities of IL-13 can provide treasured insights into the development of healing procedures for various illnesses which include asthma, allergic illnesses and positive styles of illnesses. most cancers.

System gaining knowledge of algorithms have verified to be very powerful in studying huge organic records sets and extracting significant styles and relationships. by means of harnessing the electricity of gadget learning in the context of IL-13 bioinformatics, researchers can find hidden styles, discover ability biomarkers, and higher apprehend the molecular mechanisms that motive illnesses related to IL-13. you may have a deeper know-how.
The predictive energy of system studying models helps identify new drug targets and broaden customized clinical techniques. accurately predicting IL-13 conduct lets in researchers to apprehend the effectiveness of various healing interventions and optimize remedy strategies for individual sufferers. this could revolutionize the sector of precision remedy, permitting healthcare companies to customize remedies based at the particular wishes and traits of every patient.

Similarly, the mixing of gadget learning and bioinformatics could accelerate the drug improvement process. the usage of computational fashions and predictive algorithms, researchers can search huge libraries of compounds and prioritize capacity drug candidates for in addition experimental validation. This elevated drug discovery procedure has the capability to seriously reduce the prices, time, and assets related to traditional trial and blunders methods

In precise, the dynamics of the usage of gadget learning to predict interleukin-thirteen bioinformatics will enhance our understanding of IL-13 related illnesses, identify healing targets, optimize techniques of treatment and accelerate drug improvement. springing up from ability. Through combining the strength of gadget-gaining knowledge of algorithms with the significant number of organic statistics to be had, researchers can benefit from precious insights that could exchange and enhance health care. affected person final results.

CHAPTER TWO

2.1 Introduction

IL-13 (Interleukin-13) is a cytokine, which is a tiny protein that plays a function in cell signalling and communication within the immune system. It is produced by many immune cells, including T helper 2 (Th2) cells, mast cells, and eosinophils.

Immune responses, particularly those connected to allergy and inflammatory reactions (McKenzie et al,. 1993), are tightly regulated by IL-13. It increases the development and activation of many immune cells, including B cells, T cells, and macrophages, and can also influence the activity of epithelial and endothelial cells.

Particularly, IL-13 plays a role in the pathogenesis of a number of allergic illnesses like asthma and atopic dermatitis as well as inflammatory illnesses like inflammatory bowel disease. For these conditions, IL-13 signalling targeting has emerged as a crucial therapeutic strategy.

Interleukin-13 (IL-13), a cytokine that promotes inflammation, is discussed in the study (Ningjing et al., 2021) in relation to cardiovascular diseases (CVDs). The heart's reaction to both acute and long-term injury is characterised by inflammation, although the molecular underpinnings and underlying mechanisms are poorly known. IL-13 has been linked to a number of CVDs, including myocarditis, myocardial infarction, and heart failure, according to recent investigations.

Predicting interleukin-13 (IL-13) activity is crucial because it can be used to spot those who are at a higher risk of contracting IL-13-driven disorders including asthma and atopic dermatitis (Corren et al., 2013). Healthcare professionals can track the development of disease and modify treatment regimens by identifying raised IL-13 levels or enhanced IL-13 activity. Predicting IL-13 activity can also help with the creation of specialised treatments for certain illnesses. Patients with high levels of IL-13 activity, for instance, may be more likely to respond to biologic medications that block IL-13 signalling, such as dupilumab, which is licensed for the treatment of atopic dermatitis and asthma(Rael et al., 2011). This can lessen the possibility of negative effects and increase treatment outcomes. Predicting IL-13 activity can also help with the creation of new treatments for illnesses that are IL-13-mediated. Understanding the mechanisms underlying IL-13 activity will help researchers find novel therapeutic targets and increase the variety of treatments available to patients.

There are many ways to predict interleukin 13, including experimental techniques like ELISA assays and Flow cytometry, as well as some computational techniques like Molecular Docking and Molecular Dynamics Simulations.

2.2 Literature Review

In biomedical research, machine learning is a crucial technique that enables the study of huge and complicated information. Machine learning algorithms can find patterns and relationships in data that aren't always obvious and can be used to forecast outcomes or categorise data based on certain characteristics. Finding previously undiscovered correlations between biological parameters is one of machine learning's key advantages in biomedical research. For instance, new biomarkers for disease diagnosis and prognosis can be found using machine learning algorithms. Drug targets can also be predicted based on the molecular characteristics of a disease. The capability of machine learning to handle huge and complicated datasets is another benefit. The application of machine learning in biomedical research has drawbacks as well. The calibre of the data being evaluated is one restriction. Large, high-quality datasets are necessary for machine learning algorithms to work well, and noisy or missing data might result in incorrect predictions or classifications.

ELISA, flow cytometry, and western blotting are examples of experimental tests used to forecast IL-13 activity. In computational approaches, a big collection of gene expression or protein structure data is analysed using machine learning algorithms. These techniques can be used to pinpoint prospective treatment targets and forecast the impact of IL-13 on various cell types and tissues.One essential TH2 cytokine, IL-13, is responsible for many significant aspects of airway remodelling and inflammation in allergic asthma patients (Oshima et al., 2001). Anti-IL-13 mAbs and IL-4 receptor antagonists are two promising focussed treatments for asthma that target the IL-13/IL-4/Signal Transducer and Activator of Transcription 6 pathway (Ingram et al., 2012) . IL-13 may play a significant role in the emergence of allergy disorders [3]. In a different study, researchers reported differential expression levels of 14 cytokines, including IL-13, in healthy controls, moderate COVID-19 patients, and they found that the severity of COVID-19 was directly correlated with the greater level of IL-13 expression (Yang et al,. 2020)]. In contrast to non-PAH controls, [Yuan et al.,] address the higher levels of IL-13 in blood and lung tissue in both animal models of PAH and patients with PAH.

In this paper, we provide a stacking classifier that uses the LR, RF, SVM, XGB, DT, and LGBM models to distinguish between peptides/epitopes that induce IL-13 and those that do not. We used experimentally verified human IL-13 triggering and non-triggering peptides from IEDB. We applied several cutting-edge machine learning classifiers to this dataset and assessed the model's performance.

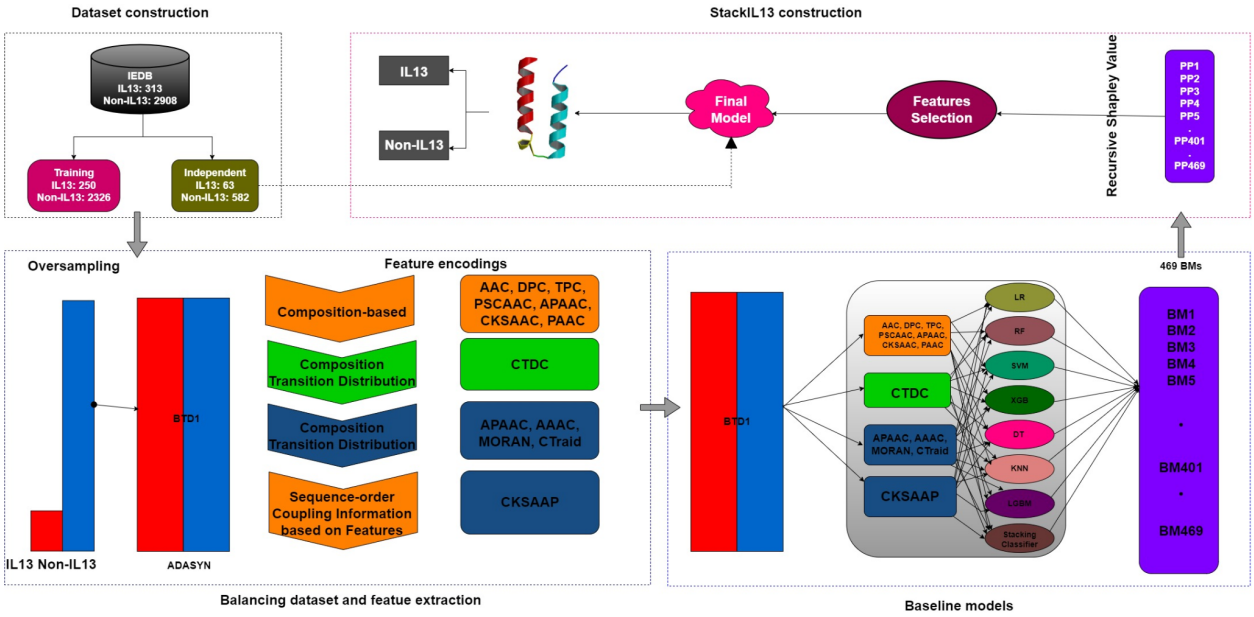
CHAPTER THREE
RESEARCH METHODOLOGY

3.1 Overview

This study used machine learning methods to estimate the levels of interleukin-13 (IL-13). Data collection, feature extraction using AAC, APAAC, and DPC techniques, addressing the problem of unbalanced data through the use of ADASYN, and feature selection using SHAP(Shapley Additive explanations) were all part of the research approach. For the purpose of predicting IL-13, multiple machine learning algorithms were trained using the chosen features. This chapter will look at,

- ❖ Data collection
- ❖ Feature Extraction
- ❖ Addressing Data Imbalance
- ❖ Feature Selection
- ❖ Splitting Data for Train - Test
- ❖ Applying Machine learning Knowledge

3.2 Methodology



3.3 Data Collection Process

In order to build the interleukin-13 prediction model (StackIL13), we obtained a benchmark dataset from an article on IL-13Pred that had already been published. The IEDB, the biggest collection of immunological epitopes, served as the source of this dataset. It includes 2908 experimentally proven non-IL-13 producing peptides/epitopes in addition to 313 experimentally validated peptides/epitopes that induce IL-13. I used Adysan to do Data balancing because the sequence numbers of the positive (IL-13-inducing peptides) and negative cases were so drastically out of balance.

3.4 Feature Extraction

The process of turning raw or highly dimensional data into a condensed set of significant and representative features is referred to as feature extraction. It entails choosing or developing a subset of pertinent features that capture the crucial data needed for a specific job or study. By concentrating on the most informative parts of the data, feature extraction tries to streamline data representation, increase interpretability, decrease dimensionality, and boost the efficiency of machine learning algorithms.

For feature extraction, the iLearnPlus Python library was utilized in this study. The development of automated machine-learning pipelines for computational analysis and predictions utilizing nucleic acid and protein sequences is made possible by iLearnPlus, the first machine-learning platform with both a graphical and web-based user interface. To the best of our knowledge, iLearnPlus integrates 21 machine-learning algorithms, more than any other web server or standalone tool currently available for biological sequence analysis. These algorithms include 12 traditional classification algorithms, two ensemble-learning frameworks, and seven deep-learning approaches.

3.4.1 AAC(Amino Acids Composition)

The relative frequencies are referred to as an amino acid sequence's makeup. The sequence contains varying quantities of the various amino acids. if the abundance of each amino acid is determined and the findings are vectorial.

3.4.2 APAAC

APAAC, or Amphiphilic Pseudo-Amino Acid Composition, is a term used in bioinformatics. Protein sequences are quantitatively represented using the feature encoding technique known as APAAC for machine learning and computational analysis.

The physicochemical characteristics and information on the amino acid sequence order of proteins are both captured by APAAC. It takes into account the arrangement of the amino acids in the sequence as well as their hydrophobic and hydrophilic characteristics.

In this method, a fixed-length feature vector contains both local and global sequence information.

3.4.3 CKSAP

The feature encoding technique CKSAP (Composition of k-spaced Amino Acid Pairs) is widely utilized in bioinformatics. In protein sequences, it captures the pairwise connections between amino acids. When calculating the frequency of k-spaced amino acid pairs, CKSAP takes into account the number of places in the sequence that separate the two amino acids. With the use of this method, tasks like protein categorization, structure prediction, and function prediction may be performed using a representation that takes into account both the order and spacing of amino acids.

3.4.4 TPC

TPC (Tri-Peptide Composition) is a feature encoding approach for representing protein sequences in the context of bioinformatics. Tri-peptides, which are repeated sequences of three amino acids, are calculated using TPC to determine their frequency of occurrence. This method helps to capture crucial structural and functional information in proteins while capturing local sequence patterns. In machine learning methods for applications like protein classification, protein-protein interaction prediction, and protein structure prediction, TPC is frequently used as a feature representation.

3.4.5 CTDC

Protein sequences are represented by the feature encoding technique CTDC (Composition Transition Distribution Complement) in bioinformatics. A protein sequence's distribution of amino acid compositions and their transitions are captured by CTDC. It takes into account both the structure of amino acids and their chronological arrangement. A succinct representation that includes both local and global sequence data is offered by CTDC. Machine learning algorithms frequently use this feature encoding technique for applications including predicting protein subcellular localization, identifying protein folds, and predicting protein function.

3.4.6 CTraid

Using the bioinformatics program CTraid, one may locate conserved areas in protein sequences. It is based on the idea of hidden Markov models (HMMs), statistical models that can be used to depict the likelihood that an amino acid sequence would occur. CTraid makes use of a collection of HMMs that stand in for conserved areas in well-known protein sequences. When a new protein sequence is uploaded to CTraid, the program analyzes it against a library of HMMs to find any potentially conserved areas.

3.4.7 DPC

DPC is an abbreviation for "dipeptide composition." Dipeptide composition is another feature representation frequently employed in protein and peptide analysis, similar to amino acid composition (AAC). The dipeptide makeup of the sequence is represented by the generated DPC feature vector. The frequency or percentage of each dipeptide is represented by a vector element.

3.4.8 Moran

The association between a variable's values at each site and those at neighboring places is determined by the Moran's I formula. The spatial autocorrelation pattern of a variable can be captured and features that reflect the spatial characteristics of the data can be extracted using Moran's I.

3.4.9 PAAC

The acronym "PAAC" stands for "Pseudo Amino Acid Composition." This feature representation is frequently used in protein and peptide analysis in the context of bioinformatics and machine learning. The amino acid composition (AAC) is an extension of the amino acid composition (PAAC), which captures additional information about the peptide sequence by taking into account the physicochemical properties of amino acids.

3.5 Data Balance

Machine learning approaches used to balance out datasets that have an uneven distribution of classes are known as data balancing. By either raising the minority class samples (oversampling) or decreasing the majority class samples (undersampling), it seeks to alter the class representation to improve model performance. As a result, bias is reduced and effective learning from all classes is made possible.

The use of data balancing procedures is essential for verifying the validity and importance of our findings given the study work's extremely unbalanced dataset. We can reduce biases and improve the overall accuracy and dependability of our research findings by resolving the class imbalance issue through appropriate data balancing procedures. I started using Adasyn to balance the unbalanced data of this work.

3.5.1 Adasyn

An extensively used method for data balancing to address class imbalance in machine learning is ADASYN (Adaptive Synthetic Sampling). By taking into account the density distribution of the minority class, it is a version of the SMOTE algorithm that aims to

improve the creation of synthetic samples. Based on the density distribution of the minority class, ADASYN modifies the production of synthetic samples. It prioritizes lower-density locations and generates more synthetic samples for samples that are more difficult to accurately represent. This flexibility helps to improve the overall balance of the dataset and decrease the impacts of class imbalance.

3.6 Feature Selection

The process of choosing a smaller subset of pertinent features from a larger collection of available features in a dataset is known as feature selection in machine learning. Finding the most discriminative and informative characteristics that significantly improve a machine learning model's capacity for prediction is the aim of feature selection. By concentrating on the most important features of the data, this technique reduces dimensionality, increases model efficiency, reduces the danger of overfitting, and enhances the model's interpretability.

Subsequently, SHAP (SHapley Additive exPlanations) was employed for feature selection in this research. SHAP is a technique for selecting and interpreting features that can be implemented in machine learning models. SHAP is predominantly utilized for feature importance analysis and interpretation; however, it can indirectly aid in feature selection by identifying the most influential features in a model's predictions.

There are many ways to predict interleukin 13, including experimental techniques like ELISA assays and Flow cytometry, as well as some computational techniques like Molecular Docking and Molecular Dynamics Simulations.

3.7 Experimental Methods

A popular experimental method for determining the amount of IL-13 in biological materials such as blood, plasma, or cell culture supernatant is the enzyme-linked immunosorbent assay (ELISA). Commercially accessible ELISA kits offer a rather easy and affordable technique to measure IL-13 levels. The low sensitivity, constrained dynamic range, and interference from other proteins in the sample are only a few of the drawbacks of ELISA assays. Another experimental method for assessing IL-13 activity in cells is Flow cytometry. In addition to measuring IL-13-induced alterations in downstream signalling pathways, flow cytometry can be utilised to identify IL-13 expression on the cell surface or intracellularly. However, because flow cytometry needs specific tools and knowledge, it might not be practical for all laboratories.

3.8 Computational methods

Additionally, we can forecast Interleukin 13 using computational techniques like Molecular Docking and Molecular Dynamics Simulations.A computational technique called molecular docking makes predictions about the affinity of a ligand (like IL-13) for a receptor (like the IL-13 receptor) based on the three-dimensional (3D) structures of both

molecules. Potential therapeutic candidates that can inhibit or boost IL-13 activity can be found through molecular docking. The precision of the protein-ligand interaction models and the availability of high-resolution protein structures, however, are constraints for molecular docking (Dhall.,et al 2021) . Protein conformational changes and the dynamics of protein-ligand interactions over time can be predicted using molecular dynamics simulations. In order to understand how IL-13 binds to its receptor and the subsequent signalling pathways involved, molecular dynamics simulations can be performed. However, setting up and analysing molecular dynamics simulations requires enormous computer resources.

3.9 Machine learning Methods

Interleukin-13 (IL-13) activity can be predicted using a variety of machine learning models, including

3.9.1 RF

For classification and regression tasks, a well-liked machine learning technique called random forest is used. Multiple decision trees are combined in this ensemble learning technique to produce predictions. The fundamental principle of random forest is to build many decision trees, each trained on a random subset of the data and characteristics. This improves the precision and robustness of the predictions while reducing overfitting. The main steps in creating a random forest model are listed below.

Data preparation -> Random sampling -> Random feature selection -> Building decisions trees -> Aggregating predictions

Using a collection of input features and the ensemble learning algorithm RF, IL-13 activity may be classified. In various research, including the prediction of IL-13 response in atopic dermatitis patients, RF has been utilised to predict the activity of IL-13. In comparison to other machine learning methods, random forest provides a number of benefits, including high accuracy, robustness, non-parametric, and feature importance

3.9.2 LR

A statistical technique called logistic regression is used to examine the relationship between a categorical dependent variable and one or more independent variables. It is a sort of regression analysis frequently used for classification tasks, like determining whether or not to forecast that a consumer would purchase a product or whether or not to determine whether an email is spam. The fundamental goal of logistic regression is to simulate the link between the independent factors and the dependent variable's probability, which has only two possible values (for example, 0 or 1). The linear

combination of the independent variables is transformed into a probability value between 0 and 1 using the logistic function, also referred to as the sigmoid function.

The Mathematical notations for Logistic regression given below

$p(y = 1|x) = \sigma(\beta0 + \beta1*x1 + \beta2*x2 + ...)$

$\sigma(x) = 1 / (1 + \exp(-x))$ [Logistic Function]

Where,

$p(y = 1|x)$ is the probability of the outcome being 1, given the values of the predictors x

$\sigma()$ = is the logistic function [$\sigma(x) = 1 / (1 + \exp(-x))$]

$\beta0$ = the intercept term

$\beta1$, $\beta2$ = are the coefficients for the predictors x1, x2

The building steps of a logistic regression model given below

Data preparation -> Model fitting -> Model evaluation -> Prediction

LR is a supervised algorithm that can be used to forecast interleukin-13 (IL-13) activity in logistic regression. When attempting to predict whether a sample belongs to one of two classes, such as IL-13 responsive or unresponsive, logistic regression is frequently used. The logistic regression model is a linear one, therefore it might not be able to capture intricate non-linear correlations between the input data and the output class. Additionally, logistic regression makes two assumptions that may not always hold true in practice: that the input features are independent, and that the relationship between the characteristics and the output class is monotonic.

3.9.3 SVM

A strong and popular machine learning algorithm for both classification and regression problems is called Support Vector Machine (SVM). It operates by identifying the hyperplane in a high-dimensional space that best classifies the data into distinct groups or accurately predicts the dependent variable in a regression issue. SVM is more robust, effective in high-dimensional domains, memory-efficient, and versatile than other machine learning methods.

The mathematical equations of SVM can be described by given below terms,

Hyperplane equation: In Support Vector Machines (SVMs), the hyperplane equation is essential because it establishes the decision boundary that divides the data points into various classifications.

w^T * x + b = 0 [Hyperplane equation]

Here,

w = weight vector orthogonal to the hyperplane.

x = input feature vector.

b = bias term (intercept).

Margin: The margin is the separation between each class's closest data points and the hyperplane. SVM aims to increase this margin as much as possible.

margin = y_i * (w^T * x_i + b)

Here,

x_i = A data point.

y_i = corresponding class label (-1 or 1).

Decision function: The decision function is applied to forecast a new data point's class label. We can define decision function by given below format,

f(x) = sign(w^T * x + b)

Here,

f(x) = Decision function

Hyperplane equation = w^T * x + b

sign() = SVM transforms the continuous output (wT * x + b) into discrete class labels using the sign function.

SVM is a supervised learning method that can categorize IL-13 activity based on input properties like protein structures or gene expression levels. An SVM model can be trained to predict the activity of interleukins, such as interleukin-13 activity, given a set of features (such as physicochemical qualities or sequencing information). By transforming the data into a higher-dimensional space where they are more easily separable, kernel functions allow SVM to handle nonlinear correlations between the features and interleukin activity. In a number of studies, including the prediction of IL-13 activity, SVM has been applied.

Data preparation -> Feature selection -> Model training-> Model evaluation -> Prediction

3.9.4 XGBoost

Extreme Gradient Boosting, or XGBoost, is a well-liked and very efficient machine learning technique that can be used for both classification and regression applications. It is a variation of gradient boosting that makes predictions by combining a number of different decision trees. The primary processes in creating an XGBoost model are listed below.

The Objective function, gradient descent are all included in the XGBoost formulas.

1. Objective function:

   $O(\theta) = L(\theta) + \Omega(\theta)$

   Here,

   $L(\theta)$ = loss function that measures the model's fit to the training data

   $\Omega(\theta)$ = regularization term

2. Gradient descent:

   $\theta\_new = \theta\_old - \eta \nabla L(\theta\_old)$

   Here,

   η = learning rate

   $\nabla L(\theta\_old)$ = gradient of the loss function at θ_old

Data preparation -> Model fitting -> Regularization -> Model evaluation -> Prediction

XGBoost is faster and more scalable than other machine learning algorithms, which are only a few of its advantages. extremely accurate and comprehensible.In machine learning, XGBoost is a potent method, especially for structured data and tabular datasets. An effective machine learning algorithm for predicting interleukin-13 (IL-13) activity is called XGBoost. The name "eXtreme Gradient Boosting" (XGBoost) refers to an improved application of the gradient boosting algorithm.The effectiveness of the XGBoost model will be influenced by the caliber and volume of the input data, the choice of pertinent features, and the selection of hyperparameters. Because of this, it's crucial to carefully preprocess and choose your data, as well as to use cross-validation techniques to adjust the XGBoost model's hyperparameters.

3.9.5 LGBM

A machine learning algorithm called LGBM (Light Gradient Boosting Machine) is intended to outperform conventional gradient boosting techniques, especially for large and complicated datasets. Similar to Random Forest in that it makes use of a decision tree framework, LGBM is a form of gradient boosting technique that takes a different approach to tree construction. The main steps of LGBM given below

Data preparation -> Decision tree framework -> Gradient boosting -> Light optimization -> model evaluation -> Prediction

By developing a machine learning model that incorporates information on the molecular characteristics and features of interleukin 13 and outputs a prediction of its activity, LGBM can be used to predict interleukin 13 activity. A dataset of known interleukin 13 activity values and associated features can be used to train the model. In situations where conventional experimental approaches may be time-consuming or unworkable, LGBM can be a helpful tool for forecasting interleukin 13 activity.

3.9.6 Stacking classifier

A form of ensemble learning method used in machine learning called stacking classifier combines several base models to increase the prediction's overall accuracy. Additionally called stacked generalization.

In a stacking classifier, multiple separate models are trained on the same data and their predictions are combined to generate the final prediction, as opposed to merely utilising one base model to make predictions. This is accomplished by developing a meta-model that uses the predictions from the underlying models as input to generate the final prediction. Numerous elements, including genetic, environmental, and lifestyle factors, have an impact on interleukin 13 expression. It can be challenging to adequately predict each of these variables with a single algorithm. A stacking classifier allows us to combine the benefits of various algorithms to provide predictions that are more precise.On our data, we may train a variety of basic models, including Random Forest, Support Vector Machines, Gradient Boosting, LR, DT, and KNN. These models might each perform admirably on certain aspects of the data while failing miserably on others. We can lower the overall error and raise the precision of the final prediction by merging the predictions of various models.

In order to find patterns and associations that can be used to predict IL-13 levels or activity, machine learning models can be trained on vast databases of biological data, including gene expression profiles, protein interactions, and clinical outcomes. To increase the precision and robustness of the predictions, these models can use a variety of features or inputs, including patient demographics, genetic variations, environmental factors, and illness state. Machine learning can be a powerful tool to predict IL-13 levels

or activity and improve our understanding of its role in a variety of biological processes and diseases, in addition to providing early detection, personalised medicine, and drug discovery.

On the other side, machine learning strategies can be utilised to get around some of these restrictions and enhance the prediction of IL-13 interactions. Incorporating more information, learning from sizable datasets, addressing flexibility and dynamics, and enhancing speed and scalability are a few ways machine learning might improve the accuracy of IL-13 predictions. To augment molecular docking and get around some of its limitations in predicting IL-13 interactions, machine learning can be used.

CHAPTER 4
RESULTS AND DISCUSSION

We used a variety of evaluation measures in this work to evaluate the effectiveness of the classifiers. Here is a quick summary of each metric.

Accuracy: An indicator of accuracy is the percentage of accurately predicted instances among all instances. It shows how well the classifier performed overall.

The Matthews Correlation Coefficient (MCC), which accounts for true positives, true negatives, false positives, and false negatives, assesses the accuracy of binary classifications. Its value falls between -1 and +1, with +1 denoting flawless predictions, 0 denoting random predictions, and -1 denoting complete discrepancy between predictions and actual results.

AUC: The capacity of a classifier to discriminate between positive and negative classes is measured by the area under the receiver operating characteristic (AUC) curve. Higher numbers correspond to better performance, and the scale runs from 0 to 1.

Sensitivity: The proportion of accurately predicted positive instances among all actual positive instances is known as the True Positive Rate (TPR), often referred to as Recall.

Specificity: Specificity, also referred to as the True Negative Rate (TNR), is the percentage of all negative events that were accurately predicted.

TABLE 1 | Performance comparison of single features (AAC) on imbalanced training dataset by using various Classifiers

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.902950311	0	0.5	1	0
RandomForestClassifier	0.913043478	0.306579757	0.557355116	0.998710232	0.116
SVC	0.904503106	0.120289956	0.508	1	0.016
XGBClassifier	0.907608696	0.270526262	0.572196045	0.988392089	0.156
DecisionTreeClassifier	0.864130435	0.256945067	0.633802236	0.919604471	0.348
KNeighborsClassifier	0.895962733	0.131955204	0.533616509	0.983233018	0.084
LGBMClassifier	0.913043478	0.359812866	0.616261393	0.984522786	0.248
StackingClassifier	0.920419255	0.420317533	0.63105589	0.99011178	0.272

The evaluation metrics for eight different classifiers (0 to 7) are shown in the table along with the relevant values for accuracy, MCC, AUC, sensitivity, and specificity. The classifiers include Decision Tree, K-Nearest Neighbors, LightGBM, Random Forest, XGBoost, Logistic Regression, and Stacking Classifier. The **Stacking Classifier**

outperforms the other classifiers in terms of **accuracy (0.920)** and **MCC (0.420),** showing superior overall performance and predictive power. The Decision Tree Classifier displays a comparatively high AUC (0.634), demonstrating good capacity to differentiate between positive and negative classes. The classifiers with the lowest specificity values, SVC and K-Nearest Neighbors, may have trouble accurately predicting the negative class.

TABLE 2 | Performance comparison of single features (APAAC) on imbalanced training dataset by using various Classifiers

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.906055901	0.180171915	0.526710232	0.997420464	0.056
RandomForestClassifier	0.906832298	0.190417353	0.52	1	0.04
SVC	0.909549689	0.248613191	0.534	1	0.068
XGBClassifier	0.906055901	0.17715641	0.524925193	0.997850387	0.052
DecisionTreeClassifier	0.847437888	0.164045657	0.585288048	0.910576096	0.26
KNeighborsClassifier	0.909549689	0.256038776	0.550065348	0.996130696	0.104
LGBMClassifier	0.906055901	0.189925838	0.532065348	0.996130696	0.068
StackingClassifier	0.916537267	0.357547089	0.582495271	0.996990542	0.168

According to the accuracy metric in the table, the classifiers' overall performance seems to be rather strong, with accuracies ranging from roughly 84.7% to 91.7%. The percentage of accurate predictions made by the classifier relative to all samples in the dataset is known as accuracy. With the exception of the DecisionTreeClassifier, which had an accuracy of roughly 84.7%, the classifiers all reached accuracy levels above 90%. This shows that a large chunk of the dataset can be correctly predicted by the majority of the classifiers. The StackingClassifier outperformed all other classifiers in terms of producing accurate predictions, achieving the maximum accuracy of 91.7%.

TABLE 3 | Performance comparison of single features (CKSAAP) on imbalanced training dataset by using various Classifiers

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.902950311	0	0.5	1	0
RandomForestClassifier	0.906444099	0.18382865	0.525140155	0.99828031	0.052
SVC	0.904891304	0.136161081	0.513570077	0.999140155	0.028
XGBClassifier	0.904114907	0.135240244	0.518495271	0.996990542	0.04
DecisionTreeClassifier	0.87189441	0.231100676	0.609540843	0.935081685	0.284
KNeighborsClassifier	0.29076087	0.095978494	0.566209802	0.224419604	0.908
LGBMClassifier	0.920419255	0.41324962	0.620345658	0.992691316	0.248
StackingClassifier	0.918090062	0.391792127	0.615485813	0.990971625	0.24

A feature extraction technique known as CKSAAP (Composition of k-Spaced Amino Acid Pairs) is used in bioinformatics and machine learning to describe protein sequences for classification tasks. The classifiers' accuracy levels ranged from about 29.1% to 92.0%. The KNeighborsClassifier had the lowest accuracy, while the StackingClassifier and LGBMClassifier excelled. The MCC values were between -0.095 to 0.413. The highest MCC values were attained by the StackingClassifier and LGBMClassifier, suggesting strong overall performance. The range of specificity was 2.8% to 90.8%. With the highest specificity and the ability to correctly identify negative cases, the KNeighborsClassifier stood out.

TABLE 4 | Performance comparison of single features (TPC) on imbalanced training dataset by using various Classifiers

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.902950311	0	0.5	1	0
RandomForestClassifier	0.911102484	0.275757862	0.549140155	0.99828031	0.1
SVC	0.904503106	0.123111498	0.511570077	0.999140155	0.024
XGBClassifier	0.904114907	0.108840175	0.509570077	0.999140155	0.02
DecisionTreeClassifier	0.88431677	0.274627098	0.623559759	0.947119518	0.3
KNeighborsClassifier	0.224767081	0.098117757	0.55644196	0.144883921	0.968
LGBMClassifier	0.903726708	0.085024792	0.504	1	0.008
StackingClassifier	0.92197205	0.423865844	0.61228031	0.996560619	0.228

The classifiers' accuracy levels ranged from about 22.5% to 92.2%. The KNeighborsClassifier had the lowest accuracy, while the StackingClassifier did the best.The AUC values were between 0.504 and 0.623. The highest AUC, indicating higher discriminative power, was attained by the DecisionTreeClassifier. Overall, various measures for the StackingClassifier and RandomForestClassifier exhibited promising results, indicating their potential applicability for the specified classification task. The StackingClassifier's excellent accuracy shows that it can successfully aggregate the predictions of various base classifiers. This is a noteworthy finding since it shows that the StackingClassifier can build on each base classifier's advantages to provide predictions that are more accurate than those made by any one classifier alone.

TABLE 5 | Performance comparison of single features (CTDC) on imbalanced training dataset by using various Classifiers

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.9029503106	0	0.5	1	0
RandomForestClassifier	0.9079968944	0.2198134944	0.5349251935	0.99785038	0.072
SVC	0.9029503106	0	0.5	1	0
XGBClassifier	0.9079968944	0.2653836557	0.5670558899	0.99011177	0.144
DecisionTreeClassifier	0.8486024845	0.1775953388	0.5930730868	0.91014617	0.276
KNeighborsClassifier	0.900621118	0.1391343671	0.5290558899	0.99011177	0.068
LGBMClassifier	0.9126552795	0.3312473985	0.5928409286	0.98968185	0.196
StackingClassifier	0.9114906832	0.3247145761	0.5939810834	0.98796216	0.2

The accuracy, specificity, and sensitivity of various classifiers for a particular classification task are displayed in the table. The CTDC (Continuous Time Dynamical Composition) feature extraction approach is used to assess the classifiers. The LogisticRegression() classifier has the lowest accuracy, specificity, and sensitivity, as seen in the table. The classifiers' accuracy levels ranged from about 84.9% to 91.3%. The DecisionTreeClassifier had the lowest accuracy while the LGBMClassifier fared the best.Overall, the RandomForestClassifier() classifier comes in second place to the StackingClassifier() classifier in terms of performance. According to CTDC, each classifier performed their duties admirably.

TABLE 6 | Performance comparison of single features (CTraid) on imbalanced training dataset by using various Classifiers

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.8994565217	0.120442606	0.524840928	0.989681857	0.06
RandomForestClassifier	0.9048913043	0.151436379	0.520710232	0.997420464	0.044
SVC	0.9029503106	0	0.5	1	0
XGBClassifier	0.9033385093	0.060109931	0.502	1	0.004
DecisionTreeClassifier	0.849378882	0.115405060	0.556017196	0.920034393	0.192
KNeighborsClassifier	0.9017857143	0.101530659	0.515420464	0.994840928	0.036
LGBMClassifier	0.9048913043	0.206526640	0.545700773	0.991401547	0.1
StackingClassifier	0.9076086957	0.221097560	0.540065348	0.996130696	0.084

The CTraid (Composition of Traid) feature extraction outcomes and several classifier performance metrics are shown in the table. The classifiers' accuracy levels ranged from about 84.9% to 90.8%. The DecisionTreeClassifier had the lowest accuracy, while the StackingClassifier did the best. The range of sensitivity values was roughly 92.0% to 100%. High sensitivity was shown by the SVC, XGBClassifier, and DecisionTreeClassifier, demonstrating their capacity to recognize positive cases. Finally, we can state that the LGBMClassifier and StackingClassifier both displayed encouraging results across a number of measures, indicating a potential fit for the specified classification objective.

TABLE 7 | Performance comparison of single features (DPC) on imbalanced training dataset by using various Classifiers

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.9029503106	0	0.5	1	0
RandomForestClassifier	0.9072204969	0.2136098087	0.53806534	0.99613069	0.08
SVC	0.9033385093	0.07492556795	0.50557007	0.99914015	0.012
XGBClassifier	0.9048913043	0.1432100892	0.51714015	0.99828030	0.036
DecisionTreeClassifier	0.8792701863	0.2661535215	0.62433533	0.94067067	0.308
KNeighborsClassifier	0.6541149068	0.2019874519	0.66566638	0.65133276	0.68
LGBMClassifier	0.9142080745	0.3767585392	0.62583147	0.98366294	0.268
StackingClassifier	0.9184782609	0.4062782291	0.62998108	0.98796216	0.272

According to the accuracy metric in the table, the classifiers' overall performance varies amongst the models. The classifiers' accuracy levels ranged from about 65.4% to

91.8%. The KNeighborsClassifier had the lowest accuracy, only 65.4%, while the StackingClassifier did the best, at 91.8%. When selecting the most effective classifier for the task at hand, it is crucial to take additional performance indicators and special requirements into account. The maximum accuracy (91.4% and 91.8%, respectively) and comparatively high MCC and AUC values are displayed by these two classifiers. Their sensitivity and specificity scores, however, are considerably dissimilar from one another, indicating that they have varying degrees of accuracy in differentiating between positive and negative samples. Understanding the causes of this variation can offer important insights.

TABLE 8 | Performance comparison of single features (Moran) on imbalanced training dataset by using various Classifiers

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
Classifier	0.90295031	0	0.5	1	0
LogisticRegression	0.89596273	-0.0019252298	0.4997007739	0.9914015477	0.008
RandomForestClassifier	0.90295031	0	0.5	1	0
SVC	0.90256211	-0.0064606547	0.4997850387	0.9995700774	0
XGBClassifier	0.82996894	0.040130875	0.5202785899	0.9045571797	0.136
DecisionTreeClassifier	0.89479813	-0.0057016722	0.4990558899	0.9901117799	0.008
KNeighborsClassifier	0.89402173	-0.019078272	0.4968409286	0.9896818573	0.004
LGBMClassifier	0.90295031	0	0.5	1	0

The classifiers' overall performance indicators are often lower for MORAN. While the accuracy of classifiers based on DPC ranges from around 65.4% to 91.8%, that of classifiers based on MORAN ranges from approximately 82.9% to 90.3%. In comparison to DPC-based classifiers, MORAN-based classifiers typically have lower MCC, AUC, sensitivity, and specificity values. The lower MORAN performance ratings imply that the feature extraction method may not be properly capturing the discriminative patterns in the data as DPC.

TABLE 9 | Performance comparison of single features (PAAC) on imbalanced training dataset by using various Classifiers

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.90566770	0.1851623858	0.5318503869	0.9957007739	0.068
RandomForestClassifier	0.90799689	0.2172362133	0.526	1	0.052
SVC	0.90916149	0.2401279721	0.5337850387	0.9995700774	0.068
XGBClassifier	0.90838509	0.2275525354	0.5369251935	0.9978503869	0.076
DecisionTreeClassifier	0.8365683	0.1374225494	0.5739140155	0.899828031	0.248
KNeighborsClassifier	0.90760869	0.2290806022	0.5454204643	0.9948409286	0.096
LGBMClassifier	0.90644409	0.1980390746	0.5340653482	0.9961306965	0.072
StackingClassifier	0.9149844	0.3383471289	0.5780653482	0.9961306965	0.16

The classifiers' accuracy levels ranged from about 83.7% to 91.5%. Indicating moderate to significant agreement between predicted and actual classes, MCC values range from 0.137 to 0.338. AUC values between 0.526 and 0.578 imply a strong ability to distinguish between positive and negative samples. The StackingClassifier, which has an accuracy of 91.5% in the PAAC table, is the best-performing classifier in terms of accuracy. The StackingClassifier was the best-performing classifier according to these criteria, achieving the greatest Matthews Correlation Coefficient (MCC) of 0.338 and Area Under the Curve (AUC) of 0.578. The DecisionTreeClassifier, on the other hand, performed the worst out of all the classifiers in the table.

TABLE 10 | Performance comparison of single features (PseKRAAC) on imbalanced training dataset by using various Classifiers

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.9087732919	0.2334402239	0.53	1	0.06
RandomForestClassifier	0.9056677019	0.3114552514	0.606822012	0.9776440241	0.236
SVC	0.9095496894	0.2486131914	0.534	1	0.068
XGBClassifier	0.9091614907	0.2424034698	0.5409251935	0.9978503869	0.084
DecisionTreeClassifier	0.9091614907	0.3301876296	0.6087566638	0.9815133276	0.236
KNeighborsClassifier	0.9064440994	0.2418801928	0.5590558899	0.9901117799	0.128
LGBMClassifier	0.9087732919	0.2842320457	0.576411006	0.988822012	0.164
StackingClassifier	0.9145962733	0.3352469332	0.5796354256	0.9952708512	0.164

The StackingClassifier seems to be the most appropriate classifier for the given dataset and research problem, according to the results. In comparison to other classifiers, it obtained the greatest accuracy, MCC, and AUC values, demonstrating its overall strong performance in accurately identifying the data. The classifiers attained accuracy levels between 90.5% and 91.5%. The RandomForestClassifier had the lowest accuracy, while the StackingClassifier did the best. Each classifier's sensitivity is also essentially the same.

4.1 Best Performance of Different Classifiers on imbalanced training dataset

The data set includes 2908 experimentally proven non-IL-13 producing peptides/epitopes in addition to 313 experimentally validated peptides/epitopes that induce IL-13. We used ten feature extraction approaches due to the dataset's extreme imbalance. Among these methods, the performance of CKSAAP, CTDC, CTraid, and DPC was particularly impressive. Our findings in Tables 3, 4, 5, 6, and 7 show that the StackingClassifier performed better than other classifiers. The accuracy of the StackingClassifier was 91.80%, and its remarkable sensitivity value was 0.99. Similar results were obtained in Table 5 with a sensitivity value of 0.987 and an accuracy of 91.11%. These outcomes demonstrate the StackingClassifier's outstanding performance when paired with the CKSAAP, CTDC, CTraid, and DPC feature extraction methods. Notable classifiers that also displayed excellent performance included LogisticRegression, RandomForestClassifier, and LGBMClassifier, with accuracy levels of about 92%.

Table 11 | Performance of various classifiers utilizing (AAC) features on Balanced training dataset.

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.6384876805	0.2776835137	0.637764388	0.57695614	0.6985726
RandomForestClassifier	0.9615548003	0.9230988603	0.9615367261	0.96001719	0.9630562
SVC	0.9022939677	0.815492252	0.9012946771	0.81728288	0.9853064
XGBClassifier	0.958368734	0.9167749844	0.958418433	0.9625967	0.954240
DecisionTreeClassifier	0.90144435	0.8034343492	0.9011878282	0.87962166	0.9227539
KNeighborsClassifier	0.7994902294	0.6523173389	0.7970765262	0.59415305	1
LGBMClassifier	0.9590059473	0.9180196811	0.9590329954	0.96130696	0.9567590
StackingClassifier	0.9668649108	0.9337609616	0.9667995781	0.96130696	0.9722921

The table shows the performance metrics for various classifiers that extract features based on amino acid composition (AAC). The accuracy, Matthews Correlation Coefficient (MCC), Area Under the Curve (AUC), sensitivity (true positive rate), and

specificity (true negative rate) of the classifiers were examined using a dataset to assess their performance. The classifier with the best accuracy, 96.69%, and MCC, 0.93376, was stackingClassifier. Additionally, it has great sensitivity (96.13% true positive rate) and specificity (97.23% true negative rate). The StackingClassifier is a prominent contender for consideration because it appears to perform well overall.

Table 12 | Performance of various classifiers utilizing (APAAC) features on Balanced training dataset.

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.63877909	0.27739478	0.63872047	0.64144454	0.63599640
RandomForestClassifier	0.97035573	0.94155664	0.96986458	0.99269131	0.94703770
SVC	0.89811155	0.80300945	0.89941649	0.83877901	0.96005386
XGBClassifier	0.95696091	0.91536464	0.95630742	0.98667239	0.92594254
DecisionTreeClassifier	0.88954765	0.77942283	0.88979434	0.87833190	0.90125673
KNeighborsClassifier	0.83772507	0.71578472	0.84114352	0.68228718	1
LGBMClassifier	0.95718050	0.91612028	0.95646572	0.98968185	0.92324955
StackingClassifier	0.97716293	0.95431278	0.97709569	0.98022355	0.97396768

The accuracy levels of the classifiers differ widely, from roughly 63.9% to 97.7%. This demonstrates a significant range in their capacity to anticipate the provided data set correctly. Among all the classifiers examined, the StackingClassifier had the best accuracy (97.7%) and the highest Matthews Correlation Coefficient (MCC), which was 0.954. A balanced trade-off between sensitivity (98.0%) and specificity (97.4%) was also shown. The StackingClassifier is successful in making accurate predictions and keeping a decent balance between true positive and true negative classifications, according to the high accuracy and MCC values. On the provided dataset, it exhibits strong discriminatory power and reliable performance.

Table 13 | Performance of various classifiers utilizing (CKSAAP) features on Balanced training dataset.

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.909188956	0.8183813732	0.9091930855	0.907996560	0.91038961
RandomForestClassifier	0.9594477998	0.9216574565	0.9593133149	0.998280309	0.92034632
SVC	0.9741156169	0.9489080811	0.9740497966	0.993121238	0.9549783
XGBClassifier	0.9559965487	0.9131609865	0.955908179	0.981513327	0.93030303
DecisionTreeClassifier	0.9344262295	0.8688508969	0.9344254484	0.934651762	0.93419913
KNeighborsClassifier	0.5383951682	0.2037406097	0.5399828031	0.079965606	1
LGBMClassifier	0.9594477998	0.9204645192	0.959346071	0.988822012	0.92987012
StackingClassifier	0.9751941329	0.9506709871	0.9751514035	0.987532244	0.96277056

CKSAAP decreases the dimensionality of the data by converting the variable-length protein sequences into a fixed-length representation. Classifiers may operate with a consistent input size thanks to this fixed-length format, which makes it simpler for them to learn from the data and generalize to new samples. This new representation helps the classifiers function better and allows them to predict protein class with more accuracy and better discrimination. According to the results and on the assumption that accuracy is the main parameter of interest, the StackingClassifier outperformed all other classifiers, achieving the highest accuracy of 97.5% and the highest Matthews Correlation Coefficient (MCC) of 0.951. It additionally showed a superb harmony between sensitivity (98.8%) and specificity (96.3%).

Table 14 | Performance of various classifiers utilizing (CTDC) features on Balanced training dataset.

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.6120430108	0.224290928	0.61205181	0.5915735168	0.6325301205
RandomForestClassifier	0.9623655914	0.924861835	0.96236919	0.9539982803	0.9707401033
SVC	0.7772043011	0.569560127	0.77725346	0.6629406707	0.8915662651
XGBClassifier	0.9582795699	0.916709356	0.95828344	0.9492691316	0.9672977625
DecisionTreeClassifier	0.8909677419	0.782390397	0.89097502	0.8740326741	0.9079173838
KNeighborsClassifier	0.8133333333	0.675052228	0.81341321	0.6276870163	0.9991394148
LGBMClassifier	0.9552688172	0.910766294	0.95527361	0.9441100602	0.9664371773
StackingClassifier	0.9705376344	0.941113876	0.97053957	0.9660361135	0.9750430293

The StackingClassifier appears to consistently perform well and achieve high accuracy based on the results shown in the tables for various feature extraction methods. The StackingClassifier exhibits excellent performance with an accuracy of 97.05%, indicating high accuracy in predicting class labels. The classifier appears to be effective in classifying cases, with high sensitivity (96.60%) and specificity (97.50%), as indicated by the MCC of 0.941 and AUC of 97.05% in CTDC performance table.

Table 15 | Performance of various classifiers utilizing (CTraid) feature on Balanced training dataset.

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.76518046	0.53424827	0.76432901	0.69518486	0.8334731
RandomForestClassifier	0.96857749	0.93852976	0.96890220	0.99527085	0.9425335
SVC	0.96857749	0.93858006	0.96822756	0.9398108	0.9966442
XGBClassifier	0.95074309	0.90323349	0.95111217	0.9810834	0.9211409
DecisionTreeClassifier	0.89978768	0.79974190	0.89961559	0.88564058	0.9135906
KNeighborsClassifier	0.79936305	0.65206097	0.79686156	0.59372312	1
LGBMClassifier	0.95265392	0.90703600	0.95302067	0.98280309	0.9232382
StackingClassifier	0.98365180	0.9673157	0.9836101	0.98022355	0.9869966

The classifiers' overall performance exhibits a wide range of accuracies, from roughly 76.52% to 98.37%, according to the accuracy metric in the table of CTraid. Area Under the Curve (AUC): The classifiers' AUC values fall between about 76.43% and 98.39%. With a 98.37% accuracy score, the StackingClassifier performs admirably, making a high percentage of accurate predictions.

Table 16 | Performance of various classifiers utilizing (DPC) features on Balanced training dataset.

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.783599574	0.56843481	0.7832419649	0.74419604	0.822287885
RandomForestClassifier	0.96996805	0.94138252	0.9702171971	0.99742046	0.943013929
SVC	0.98828541	0.97657061	0.9882746728	0.98710232	0.989447024
XGBClassifier	0.955697550	0.91164750	0.9557991824	0.96689595	0.944702406
DecisionTreeClassifier	0.928860489	0.85772907	0.9288155042	0.92390369	0.933727311
KNeighborsClassifier	0.561874334	0.24879870	0.5578245916	0.11564918	1
LGBMClassifier	0.962513312	0.92552752	0.9626584353	0.97850386	0.946813001
StackingClassifier	0.985942492	0.97188275	0.985937411	0.98538263	0.986492190

The performance of the classifiers based on the DPC (Dipeptide Composition) feature extraction in terms of accurately classifying cases varied. The classifiers' accuracy ranged from roughly 56.2% to 98.6%. Additionally, the classifiers' sensitivity, which gauges how well they can detect positive cases, varied from roughly 11.6% to 99.7%, demonstrating disparities in how well they can recognize actual positive examples. In every performance indicator, the SVC and StackingClassifier outperformed the competition in a close race for the supplied dataset. Both classifiers showed great prediction abilities and were appropriate for the task at hand thanks to their high accuracy, MCC, AUC, sensitivity, and specificity scores.

Table 17 | Performance of various classifiers utilizing (Moran) feature on Balanced training dataset.

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.5058746736	0.0084698175	0.5040380529	0.65477214	0.35330398
RandomForestClassifier	0.7680591819	0.5375278601	0.7684171651	0.739036973	0.79779735
SVC	0.568537859	0.1482907412	0.570449165	0.413585554	0.72731277
XGBClassifier	0.728024369	0.4571604341	0.7283445896	0.702063628	0.75462555
DecisionTreeClassifier	0.7338990426	0.4683872106	0.7341114238	0.716680997	0.75154185
KNeighborsClassifier	0.7563098346	0.5236093145	0.7574562218	0.663370593	0.85154185
LGBMClassifier	0.7343342037	0.4705207819	0.7347693759	0.699054170	0.77048458
StackingClassifier	0.7785030461	0.5603414635	0.7790853065	0.731298366	0.82687224

The effectiveness of the classifiers based on the Moran feature extraction varied greatly across several measures. The accuracy ranged between about 50.6% and 77.9%, while the Matthews Correlation Coefficient (MCC) showed a variety of findings, from about 0.008 to 0.560. Similar variations were seen in the AUC (Area Under the Curve) statistic, which ranged from roughly 50.4% to 77.9%. All classifiers performed poorly, as shown by the AUC, MCC, and accuracy ranges for the Moran feature extraction. The accuracy of the classifiers ranged from 50.6% to 77.9%, their MCC scores ranged from 0.008 to 0.560, and their AUC values ranged from 50.4% to 77.9%, respectively. This shows that none of the classifiers showed particularly significant prediction ability on the balanced dataset. As a result, for the given job, the overall performance of the classifiers based on Moran feature extraction is deemed inadequate.

Table 18 | Performance of various classifiers utilizing (PAAC) feature on Balanced training dataset.

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.6313284293	0.2627818701	0.6313803062	0.6233877902	0.63937282
RandomForestClassifier	0.9647338814	0.9307183944	0.9645624676	0.9909716251	0.93815331
SVC	0.9110774556	0.8305492162	0.9115301275	0.8417884781	0.98127177
XGBClassifier	0.9539160537	0.9093474539	0.953724523	0.9832330181	0.92421602
DecisionTreeClassifier	0.8885763739	0.7773048163	0.8886320671	0.8800515907	0.89721254
KNeighborsClassifier	0.8418433579	0.7211991727	0.8428632846	0.6857265692	1
LGBMClassifier	0.9562959758	0.9144815659	0.9560834799	0.988822012	0.92334494
StackingClassifier	0.976633492	0.9532659181	0.9766268901	0.9776440241	0.97560975

The performance of the classifiers based on the PAAC (Pseudo-Amino Acid Composition) feature extraction showed a great deal of variation. The classifiers' levels of accuracy, which ranged from roughly 63.1% to 97.7%, showed different percentages of accurate predictions. Similar disparities in the models' capacity to effectively capture actual positive occurrences can be seen in the sensitivity, which measures the ability to correctly detect positive examples. This ranged from roughly 62.3% to 99.1%. These findings emphasize the significance of model choice, since certain classifiers showed noticeably higher accuracy and sensitivity in comparison to others. The "StackingClassifier" seems to be the best classifier for the given dataset and research challenge based on the findings offered and the performance metrics provided. It had a maximum accuracy of almost 97.7%, while other measures showed positive results.

Table 19 | Performance of various classifiers utilizing (PseKRAAC) feature on Balanced training dataset.

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.5318464508	0.0640991971	0.5316642512	0.4544282029	0.6089002
RandomForestClassifier	0.7205661591	0.4481659497	0.7207694934	0.8069647463	0.6345724
SVC	0.6527986275	0.3196538342	0.6531371657	0.7966466036	0.5096277
XGBClassifier	0.7126313532	0.4307492382	0.7128139015	0.7901977644	0.6354300
DecisionTreeClassifier	0.7190649796	0.4457950824	0.7192779177	0.809544282	0.6290115
KNeighborsClassifier	0.6214883122	0.3259437651	0.6222688324	0.9531384351	0.2913992
LGBMClassifier	0.7083422689	0.4218342638	0.7085207461	0.7841788478	0.6328626
StackingClassifier	0.7186360712	0.4444927185	0.718842936	0.8065348237	0.6311510

We may analyze the performance of the classifiers based on the presented findings of various classifiers utilizing PseKRAAC (Pseudo K-tuple Reduced Amino Acid Composition) feature extraction, Classifiers utilizing PseKRAAC feature extraction have an accuracy range of roughly 53.2% to 72.1%. The Area Under the Curve (AUC) varies between approximately 0.532 and 0.721, while the Matthews Correlation Coefficient (MCC) ranges from 0.064 to 0.448. With an accuracy of 71.9%, an MCC of 0.444, and an AUC of 0.719, the StackingClassifier displayed the best overall performance. Its sensitivity and specificity of 63.1% and 80.7%, respectively, made it the best-performing classifier among the models that were tested.

Table 20 | Performance of various classifiers utilizing (TPC) features on Balanced training dataset.

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.9170773516	0.8420244516	0.9172948079	0.984952708	0.84963690
RandomForestClassifier	0.9558602957	0.9149089251	0.9559934445	0.997420464	0.91456642
SVC	0.9631454896	0.9280729561	0.9632442794	0.993981083	0.93250747
XGBClassifier	0.9511463467	0.9048916466	0.9512670504	0.98882201	0.91371208
DecisionTreeClassifier	0.9367902293	0.8735998131	0.9367999066	0.93981083	0.93378897
KNeighborsClassifier	0.5198200129	0.1366399744	0.5182717111	0.036543422	1
LGBMClassifier	0.948146561	0.8983231651	0.94825346	0.981513327	0.91499359
StackingClassifier	0.9700021427	0.9407425218	0.9700651918	0.989681857	0.95044852

There is a significant variation in the accuracy of the classifiers based on the TPC feature extraction, ranging from about 51.9% to 97.0%. The Matthews Correlation Coefficient (MCC) also shows a wide variety of class separation abilities among the classifiers, falling between 0.137 and 0.941. The "StackingClassifier," which attained an accuracy of about 97.0% and an MCC of about 0.941, is the best classifier for the specified dataset and research challenge based on the results and performance metrics supplied. Among the evaluated models, this classifier has the best overall performance. However, the "KNeighborsClassifier" did the poorest, with an MCC of about 0.137 and an accuracy of about 51.9%.

4.2 Performance of Different Classifiers on balanced training data set

I demonstrated how severely unbalanced the data set is in the section 3.5 Data Balance. To balance the uneven data in this job, I began utilizing Adasyn. An unbalanced dataset can have a considerably negative effect on predicting IL-13 and may result in bias in favor of the majority class, inaccurate predictions of the majority class, and erroneous assessment metrics. To balance the uneven data in this job, I began utilizing Adasyn. An oversampling method called ADASYN (Adaptive Synthetic Sampling) is used to

balance unbalanced datasets. By creating artificial samples for the minority class, it directly addresses the problem of class inequality.

Following the Adasyn data balancing Method, we applied ten feature extraction strategies. Among these techniques, the effectiveness of CKSAAP, CTDC, CTraid, and DPC stood out. Tables 13, 14, 15, and 16 present our findings, which demonstrate that the StackingClassifier outperformed other classifiers. The StackingClassifier's amazing sensitivity value in CKSAAP was 0.987, and its accuracy was 97.80%. With a sensitivity value of 0.983 and an accuracy of 98.00%, Table 15's findings were similar. These results highlight the exceptional performance of the StackingClassifier in combination with the CKSAAP, CTDC, CTraid, and DPC feature extraction techniques. With accuracy levels of about 95%, notable classifiers including LogisticRegression, RandomForestClassifier, and LGBMClassifier also demonstrated remarkable performance.

The findings show that the accuracy, AUC, sensitivity, and specificity of several classifiers were all significantly improved when balanced data were used.

TABLE 21 | Performance comparison of single features (CKSAAP) on independent Test dataset by using various Classifiers

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.9023255814	0	0.5	1	0
RandomForestClassifier	0.9069767442	0.2077707134	0.5238095238	1	0.0476190
SVC	0.9054263566	0.1695121097	0.5158730159		0.0317460
XGBClassifier	0.903875969	0.2515426701	0.5716331206	0.98453608	0.1587301
DecisionTreeClassifier	0.8604651163	0.0943936544	0.5405007364	0.93814432	0.1428571
KNeighborsClassifier	0.3023255814	0.1225584093	0.5850924562	0.23367697	0.9365079
LGBMClassifier	0.9023255814	0.2161425465	0.5566192113	0.98625429	0.1269841
StackingClassifier	0.9054263566	0.3042973657	0.6008018328	0.97938144	0.2222222

The classifiers employing feature extraction from CKSAAP_ILTest_Metrics display a variety of performance indicators. The accuracy lies between about 30.2% and 90.7%, the Matthews Correlation Coefficient (MCC) between about 0 and 0.30, and the Area Under the Curve (AUC) between about 0.50 and 0.60. The classifiers exhibit a wide variety of accuracy, MCC, and AUC values, indicating that some models outperform others greatly on this particular dataset and research challenge.

TABLE 22 | Performance comparison of single features (CTDC) on independent Test dataset by using various Classifiers

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.9023255814	0	0.5	1	0
RandomForestClassifier	0.9007751938	0.05426314164	0.5062182949	0.99656359	0.0158730
SVC	0.9023255814	0	0.5	1	0
XGBClassifier	0.9007751938	0.2066717234	0.5557601047	0.98453605	0.1269841
DecisionTreeClassifier	0.8387596899	0.08525609557	0.5426280478	0.9106521	0.1746031
KNeighborsClassifier	0.8914728682	0.06429870294	0.5152184585	0.98281784	0.0476190
LGBMClassifier	0.9085271318	0.2930335401	0.5812878416	0.98797256	0.1746031
StackingClassifier	0.9007751938	0.191787609	0.5486827033	0.98625425	0.1111111

There are many performance metrics displayed by the classifiers utilizing CTDC_ILTest_Metrics feature extraction. The Area Under the Curve (AUC) spans from roughly 0.506 to 0.581, and the accuracy falls between 83.9% and 90.9%. According on the variety of performance criteria, some classifiers are superior to others at predicting the classes for the given dataset. With its high accuracy, MCC, and AUC, the LGBMClassifier stands out as the best-performing model, giving it a great contender for precise predictions and potential real-world applications.

TABLE 23 | Performance comparison of single features (CTraid) on independent Test dataset by using various Classifiers

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.8961240	0.1301852628	0.5319505809	0.98453608	0.07936507
RandomForestClassifier	0.9054263	0.1695121097	0.5158730159	1	0.03174603
SVC	0.9023255	0	0.5	1	0
XGBClassifier	0.8961240	0.05750108894	0.5107183767	0.98969072	0.03174603
DecisionTreeClassifier	0.8573643	0.1551533656	0.5741695304	0.92611683	0.22222222
KNeighborsClassifier	0.9007751	0.1167506941	0.5203730977	0.99312714	0.04761904
LGBMClassifier	0.9054263	0.1990591132	0.5371052201	0.99484536	0.07936507
StackingClassifier	0.9085271	0.2462451213	0.5459008346	0.99656357	0.09523809

There are many performance metrics displayed by the classifiers utilizing CTraid_ILTest_Metrics feature extraction. The accuracy varies between about 85.7% and 90.9%. The maximum MCC, 0.246, and 0.546 AUC were all attained by the StackingClassifier, which also had the highest accuracy of 90.9%. This shows that the

StackingClassifier outperforms the other tested classifiers and produces reliable predictions. Contrarily, the DecisionTreeClassifier, with an accuracy of 85.7%, an MCC of 0.155, and an AUC of 0.574, performed the worst of the classifiers.

TABLE 24 | Performance comparison of single features (DPC) on independent Test dataset by using various Classifiers

Classifier	Accuracy	MCC	AUC	Sensitivity	Specificity
LogisticRegression	0.9023255814	0	0.5	1	0
RandomForestClassifier	0.9085271318	0.2401000213	0.5317460317	1	0.06349206349
SVC	0.903875969	0.1197700646	0.5079365079	1	0.01587301587
XGBClassifier	0.9131782946	0.3442011637	0.598019964	0.9896907216	0.2063492063
DecisionTreeClassifier	0.880620155	0.2449828286	0.6082883325	0.9467353952	0.2698412698
KNeighborsClassifier	0.6449612403	0.1967975813	0.6617165767	0.6408934708	0.6825396825
LGBMClassifier	0.9178294574	0.3840278336	0.6005972836	0.9948453608	0.2063492063
StackingClassifier	0.9178294574	0.3810544312	0.5935198822	0.9965635739	0.1904761905

A variety of performance indicators are displayed by the classifiers utilizing feature extraction from DPC_ILTest_Metrics. The accuracy varies between 64.5% and 91.8%, while the Matthews Correlation Coefficient (MCC) is between 0 and 0.384. The highest MCC of 0.381, the highest AUC of 0.59, and the highest accuracy of 91.8% were all attained by StackingClassifier.

4.3 Performance of Different Classifiers on Independent Test data

We use both balanced and unbalanced data to train our model, then test it using a test set of data. In Table 21, StackingClassifier's accuracy is 90.05% and its sensitivity is 0.97. The LGBMClassifier likewise did well in CKSAAP; its accuracy is 90.02% and sensitivity is 0.98.

According to Table 23, the accuracy ranges between approximately 85.7% and 90.9%. The StackingClassifier, which also had the best accuracy of 90.9%, achieved the highest MCC, 0.246, and 0.546 AUC.

Table 24 contrasts this with accuracy that ranges from 64.5% to 91.8% and a Matthews Correlation Coefficient (MCC) that ranges from 0 to 0.384. StackingClassifier, which has an accuracy rating of 91.7%, is the best classifier.

CHAPTER 5
CONCLUSION

We have contrasted the Balance & Imbalance data set with the Results in this paper. To compare the performance metrics with the conventional ML Model, we take into account ensemble approach based stacking models. Modern feature engineering techniques (Recursive Shapley Value) were employed. We used the Best Peptide Sequence extractor and displayed the outcomes of applying various algorithms separately. To achieve accurate IL-13 projection, the appraisal of entire performance is taken into account. We ultimately developed an effective StackIL13 model for the prediction of IL-13 inducing peptides after conducting several experiments. We have established that the StackIL13 prediction model we provide can actually identify IL-13, which is superior to existing IL-13 prediction models, by executing a large number of trials on the training dataset. In the identification of IL-13 peptides, machine learning approaches, including conventional algorithms, have yielded encouraging results. The selection of the dataset, feature extraction methods, and used algorithms have a big impact on the models' performance.

It is important to note that the research conducted utilizing a balanced training dataset supports the conclusions made in this paper. Additionally, cross-validation and generalization analyses should be performed to judge the stability and dependability of the suggested models in order to evaluate the performance of the classifiers on unbalanced datasets typically seen in real-world applications. In the context of the specific problem domain, this work enhances knowledge of feature selection and classifier performance. The findings demonstrate that when using our StackingClassifier (RandomForestClassifier, XGBClassifier, DecisionTreeClassifier, SVM, KNeighborsClassifier, LogisticRegression, LGBMClassifier), certain feature sets, such as CKSAAP, DPC, CTDC, and CTraid, may be employed to accurately classify data. These findings provide useful information for academics and professionals working in the categorization and feature selection domains. It is crucial to use new research feature representation techniques that can encode peptide sequences in an adaptable manner. a further optimization is to discover the IL-13 peptide using computational intelligence approaches and models.

Reference

1. Ningjing, Qian., Ying, Gao., Jian-an, Wang., Yaping, Wang. (2021). Emerging role of interleukin-13 in cardiovascular diseases: A ray of hope.. Journal of Cellular and Molecular Medicine, 25(12):5351-5357. doi: 10.1111/JCMM.16566

2. Ingram, J. L., & Kraft, M. (2012). IL-13 in asthma and allergic disease: asthma phenotypes and targeted therapies. Journal of Allergy and Clinical Immunology, 130(4), 829-842.

3. Defrance, T., Carayon, P., Billian, G., Guillemot, J. C., Minty, A., Caput, D., & Ferrara, P. (1994). Interleukin 13 is a B cell stimulating factor. The Journal of experimental medicine, 179(1), 135-143.

4. Yang, Y., Shen, C., Li, J., Yuan, J., Wei, J., Huang, F., ... & Liu, Y. (2020). Plasma IP-10 and MCP-3 levels are highly associated with disease severity and predict the progression of COVID-19. Journal of Allergy and Clinical Immunology, 146(1), 119-127.

5. Yuan, S. M. (2019). Interleukin-13 in the pathogenesis of pulmonary artery hypertension. Journal of Laboratory Medicine, 43(1), 5-11.

6. Dhall, A., Patiyal, S., Sharma, N., Devi, N. L., & Raghava, G. P. (2021). Computer-aided prediction of inhibitors against STAT3 for managing COVID-19 associated cytokine storm. *Computers in biology and medicine*, *137*, 104780.

7. McKenzie, A. N., Culpepper, J. A., de Waal Malefyt, R., Briere, F., Punnonen, J., Aversa, G., ... & Menon, S. (1993). Interleukin 13, a T-cell-derived cytokine that regulates human monocyte and B-cell function. *Proceedings of the National Academy of Sciences*, *90*(8), 3735-3739.

8. Minty, A., Chalon, P., Derocq, J. M., Dumont, X., Guillemot, J. C., Kaghad, M., ... & Caput, D. (1993). Interleukin-13 is a new human lymphokine regulating inflammatory and immune responses. *Nature*, *362*(6417), 248-250.

9. Skinnider, B. F., Kapp, U., & Mak, T. W. (2001). Interleukin 13: a growth factor in hodgkin lymphoma. *International archives of allergy and immunology*, *126*(4), 267-276.

10. Oshima, Y., & Puri, R. K. (2001). Suppression of an IL-13 autocrine growth loop in a human Hodgkin/Reed-Sternberg tumor cell line by a novel IL-13 antagonist. *Cellular Immunology*, *211*(1), 37-42.

11. Amit, U., Kain, D., Wagner, A., Sahu, A., Nevo-Caspi, Y., Gonen, N., ... & Leor, J. (2017). New role for interleukin-13 receptor α1 in myocardial homeostasis and heart failure. *Journal of the American Heart Association*, *6*(5), e005108.

12. Rael, E. L., & Lockey, R. F. (2011). Interleukin-13 signaling and its role in asthma. *World Allergy Organization Journal*, *4*(3), 54-64.

13. Corren, J. (2013). Role of interleukin-13 in asthma. *Current allergy and asthma reports*, *13*, 415-420.

14. Brombacher, F. (2000). The role of interleukin‐13 in infectious diseases and allergy. *Bioessays*, *22*(7), 646-656.

15. Tomasiak-Łozowska, M., Bodzenta-Łukaszyk, A., Tomasiak, M., Skiepko, R., & Ziętkowski, Z. (2010). The role of interleukin 13 and interleukin 5 in asthma. *Advances in Hygiene and Experimental Medicine*, *64*.

