Ensemble Approach for Antigenic Epitopes Prediction using Physicochemical Properties

A Thesis submitted for the award of degree of
Doctor of Philosophy

By

Divya Khanna

Regn No.: 901411010

under the Guidance of

Dr. Prashant Singh Rana

Assistant Professor

Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala -147004, INDIA
November, 2020

Certificate

I, *Divya Khanna*, Regn No. *901411010*, hereby declare that the work which is being presented in this thesis entitled, **"Ensemble Approach for Antigenic Epitopes Prediction using Physicochemical Properties"** in partial fulfillment of the requirement for the award of **"Doctor of Philosophy"** submitted in Computer Science and Engineering Department of TIET, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Prashant Singh Rana, refers other research works which have been duly listed in the reference section. The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Divya Khanna
(Divya Khanna)
Regn No. 901411010

This is to certify that the above statements made by the candidate is correct and true to the best of my knowledge.

Verified by:

(Dr. Prashant Singh Rana)
Computer Science and Engineering Department, TIET, Patiala-147001, India

Acknowledgements

I express my deep gratitude to my supervisor **Dr. Prashant Singh Rana** for their encouragement, support and advice at every stage of my PhD program. This thesis would not has been possible, without his support and belief in me. His contribution to this thesis goes well beyond their role as an academic supervisor and includes constant support on a personal level without which this journey may never has been completed. And for this, I am truly grateful. He is great mentor for my life as well.

I am thankful to our director **Prof. Prakash Gopalan** for providing the environment which helps me to achieve my stated objectives. My sincere thanks to Head of the Computer Science and Engineering Department **Prof. Maninder Singh** and my research committee members **Prof. R. K. Sharma**, **Dr. Shalini Batra** and **Dr. Mukesh Singh** for their constant guidance and motivation.

My special thanks to my grandparents Sh. Gian Chand Khanna, Smt. Pushpa Khanna and grandfather-in-law Sh. Kailash Naryan Seth for their blessings and love.

I would like to express my gratitude to my parents Sh. Pardeep Khanna, Smt. Asha Khanna and in-laws Sh. Pardeep Seth, Smt. Rachna Seth, for their love, care, encouragement and blessings. I am thankful to my siblings Mrs. Kanchan and Mr. Sachin for their invaluable support, love and continuous encouragement throughout my years of study. My little rising stars Ridhaydeep, Gurnoor, Heramb and Dhanashvi. I would like to thank my fellow Ph.D scholars Mrs. Isha Kansal, Mr. Arun Rana and Mr. Rajesh Kondabala for their advices.

I owe thanks to a very special person, my husband Mr. Mohit Seth for his continued and unfailing love, support and understanding during my pursuit of Ph.D degree that made the completion of thesis possible. You were always around at times I thought that it is impossible to continue, you helped me to keep things in perspective. I would like to special thank to my daughter Dhanashvi for giving me unlimited happiness, pleasure and love.

I thank the Almighty (babaji and maa) for giving me the strength and patience to work through all these years so that today I can stand proudly with my head held high.

.....dedicated to my spiritual masters and grandfather

Sh. Gian Chand Khanna

Abstract

Accurate and efficient prediction of antigenic epitopes are essential for the medical applications and immunologic research. The prediction of antigenic epitopes are challenging as compared to other bio-informatics issues. Because antigenic epitopes have many variabilities where an paratope which is a part of antibody binds to a given epitope with high accuracy. Although, continuous efforts are invested in this field for the improvement but the problem is still unsolved and attracts attention of the researchers. To improve the results of antigenic epitopes prediction, an adaptive system needs to be constructed by using machine learning techniques.

The pathogen or invader which is identifiable as a foreign substance by the adaptive immune system that is known as antigen. Normally, antigens are the structural proteins which include portion of bacterium cell membranes and spike proteins of viruses. Epitopes are the part of antigens which bind to the helper T-cells, Cytotoxic T-lymphocytes, B-cells, antibodies and antigenic molecule based upon the type of antigen. Therefore, to predict antigenic epitopes, analyze and predict diseases, to group similar genetic elements, and to find relationships or associations in biological data, machine learning techniques can be used to improve the results of such type of problems. There are many studies exist to predict antigenic epitopes. But these studies have some limitations including use of single model, fixed length of epitopes, lack of data preprocessing and fixed data partitioning approach to train the models. Because of such issues, the trained model may or may not produce a reliable and efficient prediction. Single model can be replaced with the ensemble model to predict antigenic epitopes.

Ensemble learning is a process of combining more than one model to solve a given computational intelligence problem. Generally, it is used to enhance the predictability as well as to improve the robustness of a model. Identification of T-cell or B-cell epitopes in the targeted antigen is the main goal in designing epitopes based vaccine, immune-diagnostic tests and antibody production. Therefore, three ensemble models have been developed to predict IgG and IgA antibodies antigenic epitopes, mycobacterium tuberculosis (M. tuberculosis) epitopes and B-cell epitopes.

A multilevel ensemble model has been proposed for the prediction of epitopes inducing IgG and IgA antibodies. Epitope length is important while training the model and it is efficient to use variable length of epitopes. In this ensemble approach, seven different machine learning models are combined

to predict variable length of epitopes (4 to 50-mers).

To predict T-cell M. tuberculosis epitopes, an ensemble model has been developed. The existing NetMHC 2.2, NetMHC 2.3, NetMHC 3.0 and NetMHC 4.0 etc estimate binding capacity of peptide. This is still a challenge for those servers to predict whether a given peptide is M.tuberculosis epitope or non-epitope. One of the servers, CTLpred works in this category but it is limited to peptide length of 9-mers. Therefore, a direct method of predicting M. tuberculosis epitope or non-epitope has been proposed which also overcomes the limitations of above servers. The proposed method is able to work with variable length epitopes having size even greater than 9-mers. The proposed ensemble model is designed by combining three models and is used to predict M. tuberculosis epitopes of variable length (7 to 40 mers).

The third hybrid model has been designed by using stacked generalization ensemble technique for prediction of linear B-cell epitopes. The goal of using stacked generalization ensemble approach is to refine predictions of base classifiers and to get rid of the worse predictions. In this ensemble model, six machine learning models are fused to predict variable length epitopes (6 to 49 mers).

The three proposed ensemble models contain different machine learning models. In the training process, other models are also trained on these datasets. To meet the objective of ensembling, i. e. combine weak models to improve their performance, thus we have selected the weak and strong models. The models whose performance is poor considered as weak models. On the other hand, strong models are ones which produce accurate predictions. We ensemble these models to get improved and robust results. In the ensembling process, there are multiple trained weak and strong models. There are various combinations to get proposed ensemble model. The best performer combination is selected as the final ensemble model for the antigenic epitopes prediction.

A data division approach has been proposed in which data is provided to each model in such a way that they can properly learn it. This approach enhances the predictability of the proposed model. For feature selection, different approaches are used. All the proposed models are efficient to predict variable length of epitopes. To check the consistency of proposed ensemble models' prediction, repeated k-fold cross validation has been performed. Each proposed ensemble model has been evaluated via evaluation parameters like Gini, area under the curve, accuracy, sensitivity and specificity. To check the improvement in the results, proposed models are compared with the existing systems.

List of Publications

1. **D Khanna** and PS Rana, Multilevel Ensemble Model for Prediction of IgA and IgG antibodies, Immunology Letters, Elsevier 184 (1) (2017) 51-60. [Impact Factor: 3.27]

2. R Rayal, **D Khanna**, J Kaur, N Hooda, and PS Rana, N-semble: neural network based ensemble approach, International Journal of Machine Learning and Cybernetics, Springer 10 (2) (2019) 337-345. [Impact Factor: 4.08]

3. **D Khanna** and PS Rana, Ensemble Technique for Prediction of T cell Mycobacterium Tuberculosis Epitopes, Interdisciplinary Sciences: Computational Life Sciences, Springer 11 (4) (2019) 611-627. [Impact Factor: 1.51]

4. M Goyal, **D Khanna**, PS Rana, T Khaibullin, E Martynova, S Khaiboullina, A Rizvanov and M Baranwal, Computational Intelligence Technique for Prediction of Multiple Sclerosis Based on Serum Cytokines, Frontiers in neurology, Frontiers 10 (2019) 781. [Impact Factor: 3.55]

5. N Verma, H Singh, **D Khanna**, PS Rana and SK Bhadada, Classification of drug molecules for oxidative stress signalling pathway, IET Systems Biology, IET 13 (5) (2019) 243-250. [Impact Factor: 1.13]

6. **D Khanna** and PS Rana, Improvement in Prediction of Antigenic Epitopes using Stacked Generalization: An Ensemble Approach, IET Systems Biology, IET 14 (1) (2019) 1-7. [Impact Factor: 1.13]

List of Abbreviations

APCs	Antigen-Presenting Cells
AUC	Area Under the Curve
avNNet	Averaged Neural Network
BCRs	B-cell Receptors
CL	Class of Peptides
CTL	Cytotoxic T-Lymphocyte
ELM	Extreme Learning Machine
GAM	Generalized Additive Model
GBM	Generalized Boosted Regression Modeling
GLM	Generalized Linear Model
HLA	Human Leukocyte Antigen
Ig	Immunoglobulin
M. tuberculosis	Mycobacterium Tuberculosis
MCC	Matthews Correlation Coefficient
MHC	Major Histocompatibility Complex
mIgs	Membrane-bound Immunoglobulins
ML	Machine Learning
NK	Natural Killer Cells
RF	Random Forest
ROC	Receiver Operating Characteristics
RRF	Regularized Random Forest
Sens	Sensitivity
SL	Sequence Length
SMOTE	Synthetic Minority Over sampling Technique
Spec	Specificity
SVM	Support Vector Machine
TB	Tuberculosis
TCRs	T-cell Receptors

Th	Helper T-Lymphocyte
TOPSIS	Technique for Order Preference by Similarity to an Ideal Solution

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this Chapter, the background of research, problem statement, research gaps and objectives are described. This Chapter introduces the human immune system, innate immune response, adaptive immune response, antigen recognition, Cytotoxic T-lymphocyte, Helper T-lymphocyte and B-cell. The definition of machine learning, categorization of machine learning which includes supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning are also described. The introduction to ensemble machine learning, various type of ensembling, need of ensemble machine learning and detailed description of models which are used in the prediction of antigenic epitopes are also presented.

1.1 Background

The antigen-determinant *i.e.* epitope is the portion of an antigen which is identified by the human immune system, especially by B-cells or cytotoxic T-cells. Recognition of the epitopes is an important step in the development of vaccine, immunodiagnostic tests and antibody production because an epitope plays an essential role in the activation of the immune system.

Traditionally, epitopes are recognised by synthesizing full length peptides and then immunological observations are performed. Without computer interference, biologists identify epitopes by doing experiments in the wet labs. While doing experiments, they have to test all the peptides individually to get the epitopes. This makes their task tedious in terms of efforts, cost and time. To make biologist's task easy, an accurate statistical model is required which can classify whether a given peptide is an epitope or a non-epitope.

Currently, one of the effective approaches for the epitope prediction is machine learning approach. It is an application of artificial intelligence which facilitates the system to learn automatically. Machine learning technique is beneficial because it facilitates the computer to understands the hidden patterns within the dataset and produces predictions on the unknown data without human interference.

Therefore, with the help of machine learning techniques only those samples which are filtered by these techniques are used in the wet labs for further analysis like in experiments, peptide based vaccines, epitope based antibodies and diagnostic tools.

In immunologic research and biological applications, an accurate prediction model for antigenic epitopes are required but it is still a problem in bioinformatics. Accurate functioning of the human immune system depends upon humoral and cellular immune response because they assist or activate each other depending upon the existence of foreign agent. Thus, there are various techniques to predict B-cell and T-cell epitopes which minimizes the cost and human efforts.

1.2 Problem Statement

Accurate and efficient prediction of antigenic epitopes are essential for the medical applications and immunologic research. The prediction of antigenic epitopes are challenging as compared to other bio-informatics issues. Because antigenic epitopes have many variabilities where an paratope which is a part of antibody binds to a given epitope with high accuracy. Although, continuous efforts are invested in this field for the improvement but the problem is still unsolved and attracts attention of the researchers. To improve the results of antigenic epitopes prediction, an adaptive system needs to be constructed by using machine learning techniques.

Machine learning is an application of artificial intelligence which facilitates the system to learn automatically and correct from experiences without any human interference. It focuses on designing a computer program which can use data and learn from it. And then, trained models are used to predict the unknown data. The pathogen or invader which is identifiable as a foreign substance by the adaptive immune system that is known as antigen. Normally, antigens are the structural proteins which includes portion of bacterium cell membranes and spike proteins of viruses. Epitopes are the part of antigens which bind to the helper T-cells, Cytotoxic T-lymphocytes, B-cells, antibodies and antigenic molecule based upon the type of antigen.

Therefore, to predict antigenic epitopes, analyze and predict diseases, to group similar genetic elements, and to find relationships or associations in biological data, machine learning techniques can be used to improve the results of such type of problems. There are many studies exist to predict antigenic epitopes. But these studies have some limitations including use of single model, fixed length of epitopes, lack of data preprocessing and fixed data partitioning approach to train the models.

Because of such issues, the trained model may or may not produce a reliable and efficient prediction. Single model can be replaced with the ensemble model to predict antigenic epitopes.

Ensemble learning is a process of combining more than one model to solve a given computational intelligence problem. Generally, it is used to enhance the predictability as well as to improve the robustness of a model. The ensemble approach is used because it is capable of boosting the weak learners. The ensemble approach uses divide and conquer method in which a complex problem is divided into multiple chunks that are easy to analyse and solve. This approach has advantage that the ensemble model can adapt any diversity in the data more correctly as compared to single model. It suggests that the ensemble approach is more efficient than the single model.

Therefore, three ensemble models have been developed to predict antigenic epitopes. A data division approach has been proposed in which data is provided to each model in such a way that they can properly learn it. This approach enhances the predictability of the proposed model. For feature selection, different approaches are used. All the proposed models are efficient to predict variable length of epitopes. The proposed models produce improved predictions as compared to existing models.

1.3 Research Gaps and Research Objective

Prediction of antigenic epitopes are essential because the information of peptide's epitopes has an important role in production of epitope-based vaccines, antibodies and diagnosis. These are injected into the recipient to induce immune response. Hence, after exhaustive review, this section presents the various research gaps and the research objectives.

1.3.1 Research Gaps

1. In the literature, SVM, RF, NN are mostly used. To get effective predictions, more models need to be explored for the epitope prediction [1–5].

2. It is essential to select those features in the data which are most relevant to the problem domain, this process is known as feature selection. It's main purpose is to reduce the complexity of the dataset. Random forest has in-built property to select an important feature. Instead of RF, there are multiple techniques need to be analyzed like RF-Gini, correlation matrix

filters, principal component analysis (PCA). Moreover, many papers surveyed lack the feature importance module itself.

3. In most of the papers, K-fold cross-validation has been used [1–5]. But, there are many more validation types which can be used to check the robustness of the model like re-substitution validation, hold-out validation, leave-one-out validation and repeated k-fold cross validation.

4. When an antibody binds to an invader antigen, it attaches with the portion of that antigen rather than the whole full-length antigen protein. A regular full-length protein sequence actually contains many different epitopes against which antibodies can bind. Thus, selecting variable length epitope is an important approach for efficient prediction.

1.3.2 Objectives

The following objectives are set for the research proposal:

1. To study and review existing machine learning techniques to predict Antigenic Epitopes.

2. To propose an ensemble machine learning framework that predicts Antigenic Epitopes efficiently using physicochemical properties.

3. To test the proposed framework using various parameters like total time, accuracy, area under the curve.

4. To validate the performance of proposed framework by comparing with the existing schemes.

1.4 Human Immune System

The human immune system is mechanism of biological components which defends the organism from foreign invaders. It is capable of detecting different invaders such as fungi, bacteria, virus and other toxic agents. The defense system consists of various types of cells and all of them have different functions to perform.

The immune system is categorised into two parts which are innate immune system and adaptive immune system. The humans are defended by layered lines of immune system. The first defense line is innate immune system which comprises of physical boundaries like protein, skin and variety of

the white blood cells. If a pathogen ruptures the innate immune system then second line of defense *i.e.* adaptive immune system deals with it. Response rate of adaptive immune system is slower as compared to innate immune system. But the response of adaptive immune system is more focused and efficient. The system has immunological memory which facilitates the adaptive immune system to respond fast and more effectively when it recognises the same pathogen [6]. These two defense lines do the functions differently but interact with each other to defend the human body. For instance, some components of the adaptive immune system can trigger or assist the innate immune system and vice versa. Figure 1.1 describes the human immune system with its classification.

The immune system has ability to differentiate the self and non-self which is important to defend the human body from foreign pathogens. The another feature of immune system to protect human body is to find the defected cells which can be infectious by the cancer cells and the viruses. Sometimes, the immune system loses it's feature to differentiate the self and non-self. In this case, it kills normal cells which results in the autoimmune diseases [7].

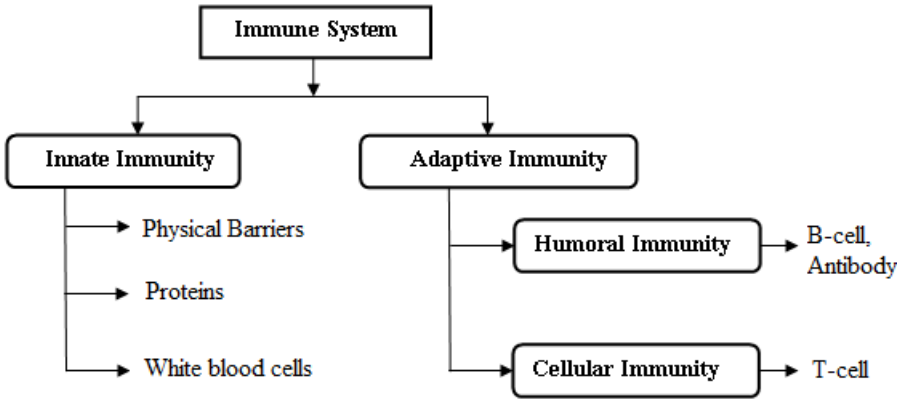

Figure 1.1: Pictorial view of human immune system

The human immune system is further divided into two subsystems which work directly or indirectly with each other to deal with the pathogen. The two subsystems are innate immune system and adaptive immune system which are described in the upcoming section.

1.4.1 Innate Immune System

The innate immunity is available in the humans as well as all types of plants and animals. It is a first defense line against the foreign invaders [6]. It identifies and reacts to the invaders in a non-specific or generic manner. It means that every time the response rate is constant while treating the invaders. This first defense line reacts rapidly against the infections. The main difference between the innate and

adaptive immunity is that there is no enhancement and long term protection by the innate immunity [8].

The innate immune system consists of humoral components, anatomical barriers and cellular components. Anatomical barriers includes protection system in gastrointestinal pathways, skin, eyes and respiratory pathways. If the invader raptures these barriers then acute inflammation which is an another mechanism of innate immunity deals with it. Humoral components work simultaneously in the inflammation. Such components are present in the serum or formulate in the infectious places.

The innate immunity has its main humoral component *i.e.* complement system. This system has the number of chemical reactions which boosts the capability of phagocytic cells and antibodies to discard the invaders. There are small proteins in the complement system which resides in the blood circulation [9]. Other humoral components except complement system include transferrin, lactoferrin, lysozyme, Interleukin-1 and interferon. All these components play an important role in the innate immune system [9]. Leukocytes are the part of cellular components which are the one of the types of white blood cells. Leukocytes are not specifically related to any tissue or organ and are different from all of the body cells. Like, single cell organism, leukocytes are self-sufficient and also capable to move easily in the body. They have the capacity to discard the invaders and capture the foreign fragments which they collected from the body including the lymphatic system and blood [10]. Leukocytes contain mast cells, natural killer cells (NK), basophils, phagocytic and eosinophils cells. Macrophages, dendritic and neutrophils cells come under phagocytic cells. These cells do the phagocytosis process which is used to kill the invaders. During this process, invaders are engulfed by the cell membrane to form an internal phagosome. Afterwards, phagosome combines with either granule or a lysosome to break out the invaders [11].

1.4.2 Adaptive Immune System

The adaptive or acquired immunity is able to identify particular types of invaders and also has the ability to memorize the invaders for the faster future responses. It means that the adaptive immune system doesn't work adequately when it finds the invaders first time. Its first response rate is slow and requires upto three weeks to heal the infection. The experience from the first response is used to create the memory for a particular type of invader. When the immune system recognizes the same invader again in the body, then it's secondary reaction towards the invader will be faster and more effective.

The secondary reaction of the system is fast enough to discard the invader before it causes any serious harm to the body. This feature of adaptive immunity to memorizes the invader *i.e.* immunogenic memory helps the immune system to provide long time protection to the body.

The adaptive immune system consists of lymphocytes which are a specific type of white blood cells. Like leukocytes, lymphocytes also can travel in the whole body through lymph system and the blood. The main lymphocytes in the adaptive immunity are B-cells and T-cells which are generated by stem cells in the bone marrow [9]. The T-cells are further divided in two parts which are cytotoxic T-lymphocyte (CTL) and helper T-lymphocyte (Th). Antigen recognition is done by the B-cell, CTL and Th cells to identify the invaders and are explained in the upcoming section.

1.4.2.1 Antigen Recognition

The pathogen or invader which is identified as a foreign substance by the adaptive immune system is known as antigen. Normally, antigens are the structural proteins which include portion of bacterium cell membranes and spike proteins of viruses.

Antigenic molecules are huge biological polymers. Various surface and molecular attributes are introduced by these polymers. Such attributes are used as the sites of interaction between Th cells, CTLs, B-cells, antibodies and antigenic molecule. The binding sites are known as epitopes. Generally, a single antigen has various epitopes which are identified by the specific antibodies. Mostly, Th cells and T-cell receptors (TCRs) identify epitopes on the surface of antigen-presenting cells (APCs). On the other hand, B-cell receptors (BCRs) identify epitopes in the extracellular fluid [6, 12]. These APCs include dendritic cells and macrophages which absorb invader or pathogen by phagocytosis and break out the antigen into short peptides. A few of these peptides are epitopes. Such epitopes are transited to the APCs' membrane and facilitated to the T-cells by using major histocompatibility complex (MHCs) molecules. MHCs are categorized in three subclasses including class I, II and III. MHC genes are extremely polymorphic and have various types. MHC class I discovers on each nucleated cells and shows epitopes to CTLs. MHC class II shows epitopes to Th cells and generally, available on the APCs which include B-cells, dendritic cell and macrophages. MHC is also known as human leukocyte antigen (HLA) [13] in humans.

1.4.2.2 Cytotoxic T-lymphocyte

The CTLs have authority to discard the virus infected cells or invaders to stop the infection process. Further, CTLs also have the ability to discover and kill cancer cells as well as dysfunctional cells. Antigen presentation on MHC I activates the CTLs and then CTLs releases cytotoxins to create pores on the membrane of targeted cell. The pores allow water and ions to enter into the infected cell which cause cell lysis.

Furthermore, CTLs emit serine proteases named granzymes to flow in the cells through pores and activate apoptosis process which is programmed cell death [9]. Most of the CTLs are expired after removing the infection but some of them survive to become memory cells. When the same antigen again encounters in the body, then the response rate will be faster because of such memory cells.

1.4.2.3 Helper T-lymphocyte

In the adaptive immune response, Th cells plays an important role. However, Th cells don't have the ability of phagocytic or cytotoxic. They act as intermediators to manage all other responses of the immune system. Antigen presentation on MHC II activates the Th cells to identify epitopes. After activation, they transmit signals in the form of cytokines which trigger other cells including macrophages, B-cell and CTLs [9]. The Th cells are subdivided into two groups including Th1 and Th2. Th1 cells emit Interferon $-\gamma$ for the activation of the bactericidal features of macrophages and the opsonizing of complement fixing antibodies on B-cells. Th2 cells emit interleukin 4, 5, 6, 10, and 13 to stimulate antibody generation process of B-cells.

Antibody is the fundamental component of humoral immune system. Usually, Th1 responses are efficient to fight against invaders which are within the cells *i.e.* intracellular invaders. On the other hand, Th2 responses are efficient for extracellular invaders such as helminths and toxins [9]. Like CTLs, some of the Th cells will be expired after removing the infection but few of them survive to become memory cells.

1.4.2.4 B-cell

B-cells are essential in the antibody production because they are the main components of humoral immune system. The antibody can be referred as immunoglobulin (Ig). Antibodies are represented as a Y-shaped protein shown in Figure 1.2. The antibodies are divided into five groups such as IgA, IgE,

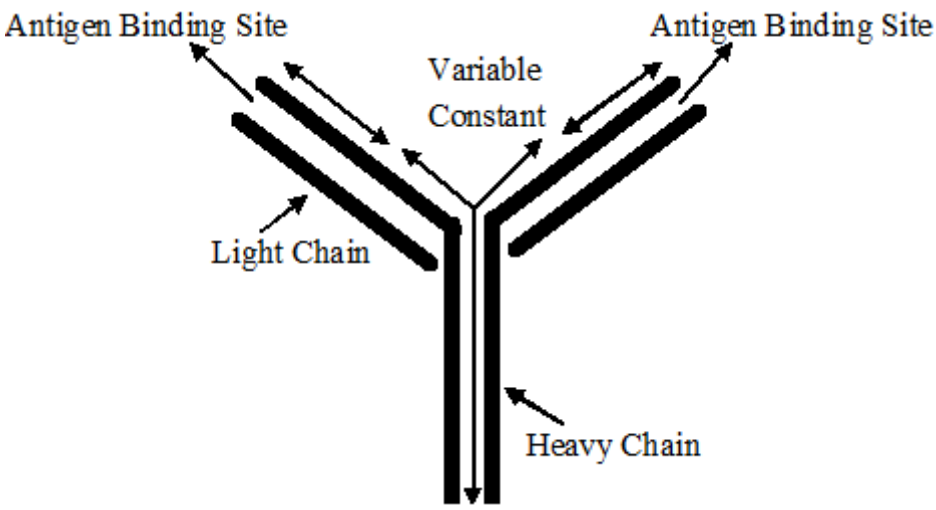

Figure 1.2: View of antibody as Y-shaped.

IgD, IgM and IgG. All these groups of antibody have their own different biological features and can fight against various kinds of antigens [14].

In the immune system, antibodies do functions in three different ways. Firstly, it bind with invaders or pathogens to restrict them from invading or destroying the cells. Secondly, invaders covered with antibody simulate the phagocytosis feature of macrophages. In the Figure 1.2, antibody has two paratopes which means that the two invaders can be bound together. Many antibodies can tied to group of many cells or parts of invaders. This process helps the macrophages to eliminate many cells or parts of invaders simultaneously. Thirdly, complement system and other immune responses are stimulated by antibodies which lead to the complete elimination of the invaders [15].

When B-cells are activated and grow into plasma B-cells only then antibodies can be produced. To activate B-cells, there are two ways including T-cell dependent and T-cell independent activation. In T-cell dependent activation, antigen is recognized by APCs through MHC II, then Th2 cells discharge interleukin 4, 5, 6, 10 and 13 to stimulate B-cells. In T-cell independent activation, B-cell receptor gets directly attached to the antigens and then B-cells are activated. The produced plasma B-cells live in the body for two to three days. Out of these, plasma cells only 10% of them survive as memory B-cells and remain in the body for long time. When the same antigens again encounter in the future, the memory B-cell will become the plasma b-cells to generate antibodies [9].

The collective working of immune system's components defend the body from infections, viruses or any other disease. Both the defense lines *i.e.* innate immunity and adaptive immunity can activate each other. Further, component of immune system can boost many activities. For instance, Th1 and Th2 cells transmit signal to stimulate macrophages and B-cells, respectively.

1.5 Machine Learning

The formal definition of machine learning (ML) is given as:
Tom Mitchell said ”A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

In machine learning, the role of dataset and the algorithms or models are essential. The quantity and quality of dataset affect the learning process and the prediction performance of models. The models are used to identify and learn the properties of data. Models learn from the examples or experiences. It means to design a computer program which is able to determine the output of a task by examining the examples. In other words, the process of learning from examples makes a model capable of producing effective predictions.

1.5.1 Categorization of Machine Learning Tasks

The machine learning problems can be typically divided into four subgroups which include supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning as shown in Figure 1.3 and all are explained in the upcoming section.

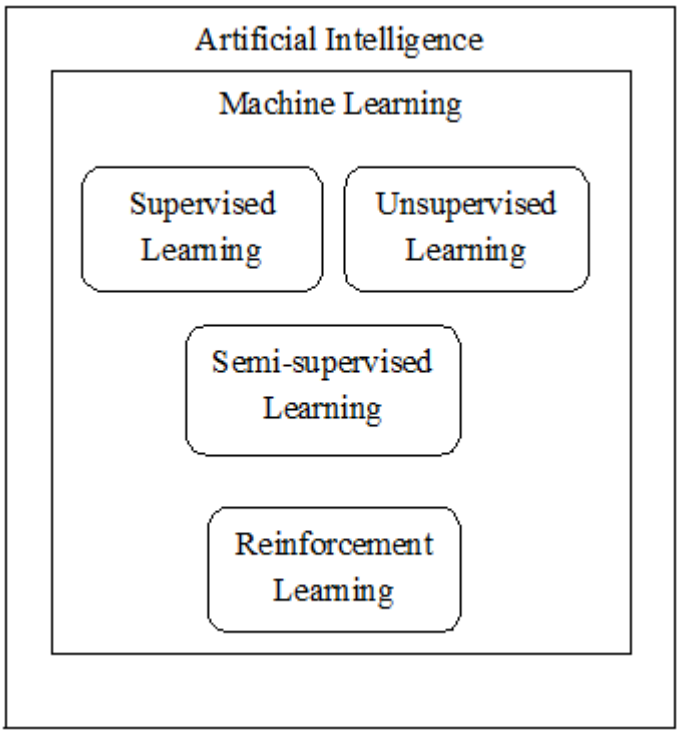

Figure 1.3: Categorization of machine learning tasks.

1.5.1.1 Supervised Learning

Supervised learning is a learning which has well labeled data that means correct output of the data is associate with them. This labelled dataset is used to train the models and is represented in Figure 1.4. The new dataset is provided to this trained model to produce predictions.

Basically, in supervised learning models, the relationship and dependency between the input features and the target outputs are analyzed. These learned relationships facilitate the model to predict new data. Commonly used supervised models include decision tree, support vector machine, random forest and neural networks etc.

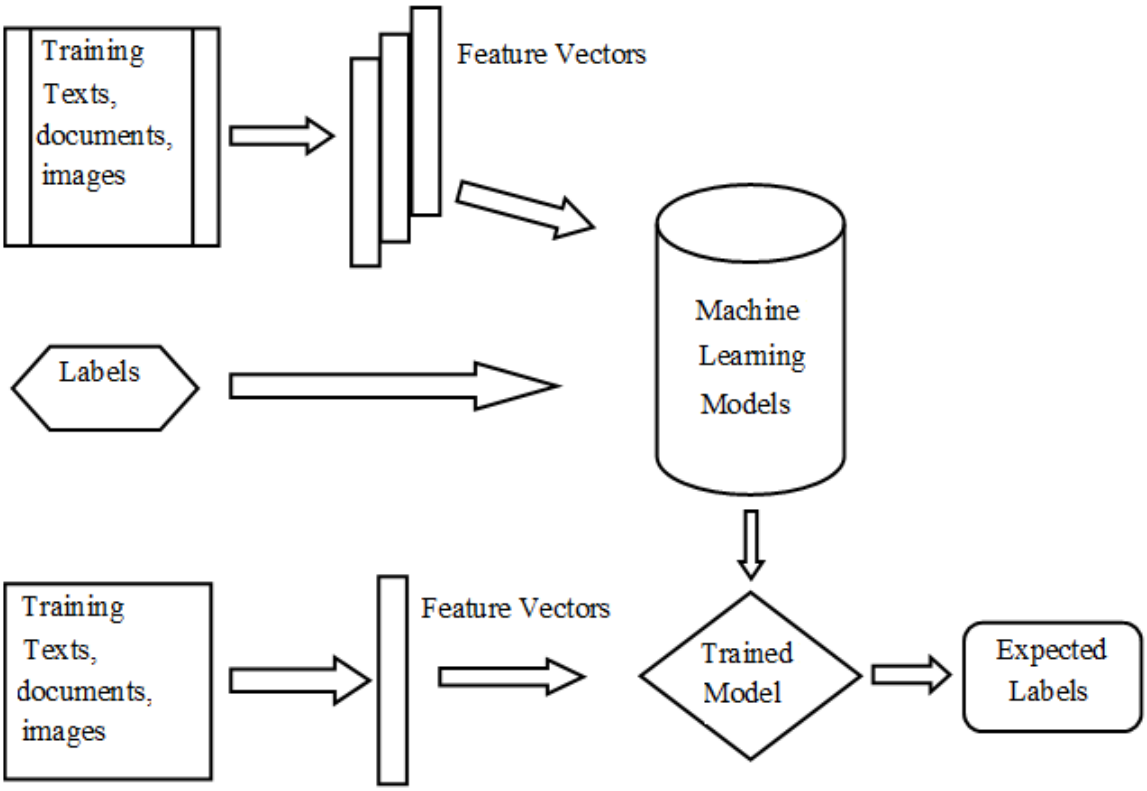

Figure 1.4: Supervised learning.

1.5.1.2 Unsupervised Learning

Unsupervised learning means train the model with dataset which is not classified or labeled and allow the model to analyse it without any guidance and is represented in Figure 1.5. In this learning process, model needs to group shuffled information based upon their patterns, similarities and dissimilarities without knowing any training data in advance. Here, the dataset is in unstructured form which contains unknown data, noisy data and missing values etc.

Contrary to supervised learning, no instructor is available in unsupervised learning which means the model will not get any labelled training dataset. Thus, models have to find out the hidden pattern with in the unlabeled data by its own.

When model finds the hidden pattern in the dataset then it creates clusters of them. Once the clusters are created, new dataset or unknown data is provided to the model to find out its cluster. The unsupervised models are specifically used in the scenario where the human expert has not any prior knowledge what to find in the data. Commonly used unsupervised models include k-means clustering and association rules.

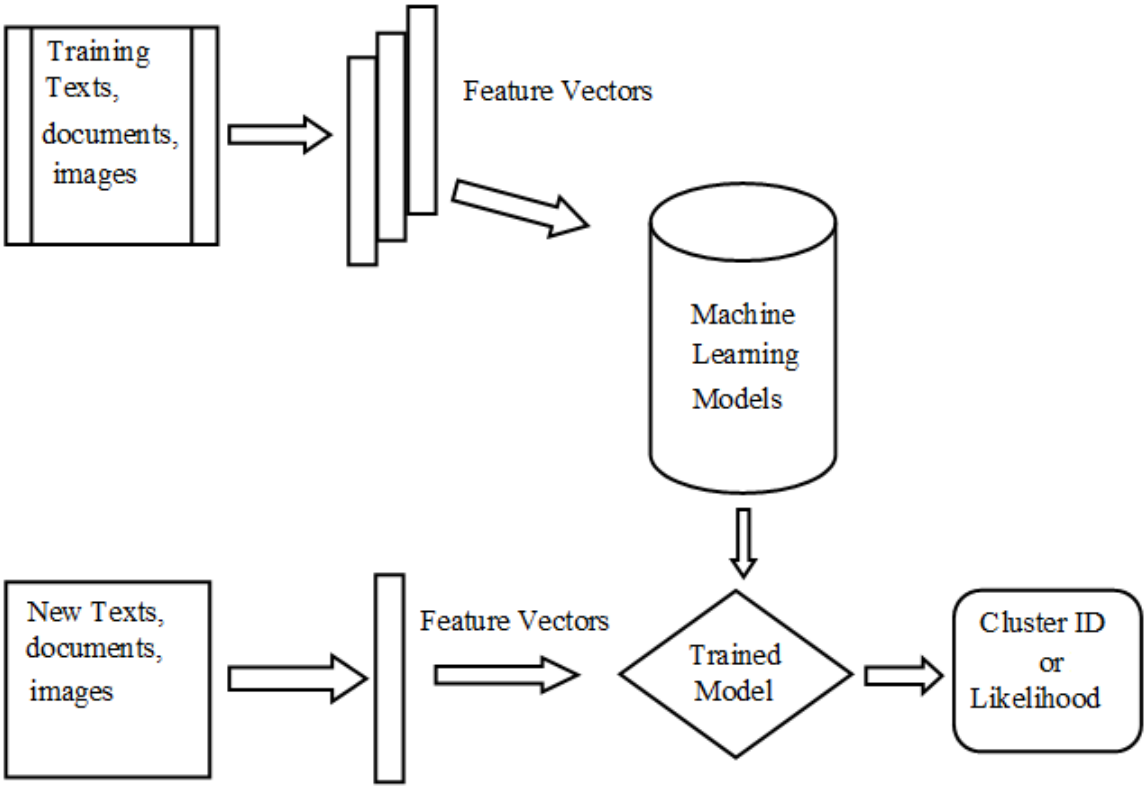

Figure 1.5: Unsupervised learning.

1.5.1.3 Semi-supervised Learning

In the above mentioned two types of learning, either whole the records of dataset are labelled or not labelled. The semi-supervised learning comes between supervised learning (labels in dataset exist) and unsupervised learning (labels in dataset don't exist) as shown in Figure 1.6. Mostly, it is expensive to do labelling of each record in the dataset and needs human expert to do this task. When some of the records are labelled in the dataset, semi-supervised learning is the effective approach to build the models.

Many researchers have analysed that when unlabeled and labeled data are used together, it can provide improved learning with less human efforts and costs as compare to unsupervised learning and supervised learning. This learning approach implies that without knowing the group membership of unlabeled data, such data has essential information regarding the group features.

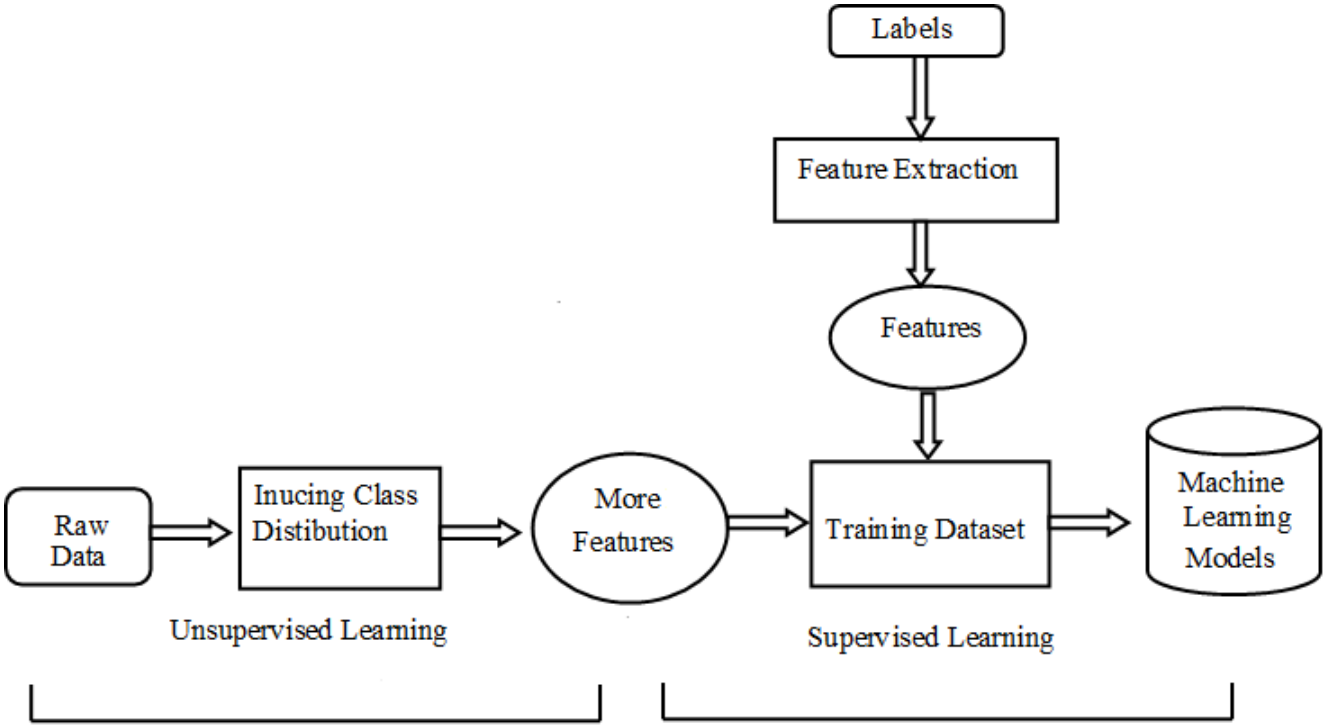

Figure 1.6: Semi-supervised learning.

1.5.1.4 Reinforcement Learning

Reinforcement learning algorithm analyses the environment repetitively before taking the actions. It facilitates the machines and software agents to automatically decide the best action to take within a specific situation. This approach aims at using experiences which are collected by interacting with the environment that would maximize the benefits or minimize the risk. The agent needs reward as a feedback to analyse its actions, this process is called as reinforcement signal as shown in Figure 1.7. The agent has authority to decide the perfect action to take in his present state and uses hit and trial method. The agent has the count of his correct and wrong answers, which is required to reward or penalize him. The commonly used reinforced algorithms are Q-Learning, temporal difference (TD) and deep adversarial networks.

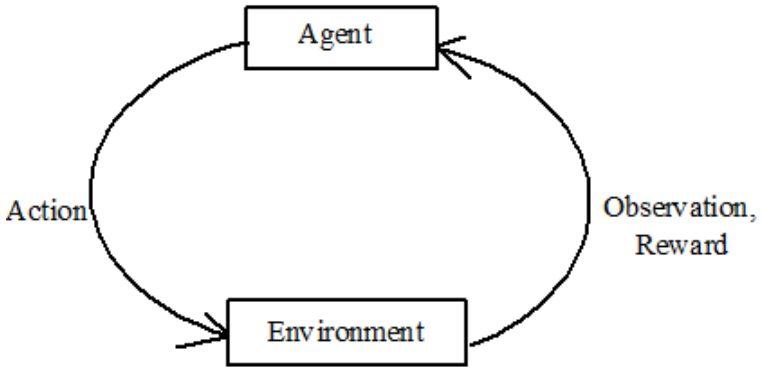

Figure 1.7: Reinforcement learning.

1.5.2 Ensemble Machine Learning

Ensemble learning is a process of combining more than one model to solve a given computational intelligence problem. Generally, it is used to enhance the predictability as well as to improve the robustness of a model. The combination of learners create an ensemble model, the learners are known as base learners. The ensemble model is generally much more stronger than these base learners. The ensemble approach is used because it is capable of boosting the weak learners [16]. The significant improvement in the prediction world by the use of ensemble approach, encourages the researchers to solve the problems of different fields. There are many more benefits of ensemble approach which include effective prediction results, selection of relevant features, combination of data, incremental learning and correction of errors.

The ensemble approach uses divide and conquer method in which a complex problem is divided into multiple chunks that are easy to analyse and solve. This approach has advantage that is the ensemble model can adapt any diversity in the data more correctly as compared to single model [17]. It suggests that the ensemble approach is more efficient than the single model [18]. The progress of ensemble approach depends upon the diversity in the individual model corresponding to misclassified instances [19].

Polikar [20] stated that there are four ways to attain this diversity. Firstly, train individual model with different data chunk. Secondly, use different training parameters. Thirdly, use different properties to train the model and finally, combine different types of models.

According to Dietterich [21], there are three reasons which conclude that the ensemble model is efficient than the single model. The first is that the required information to select one correct hypothesis is not always facilitated by the training dataset. The second is that the weak models are

not properly trained. The third is that the hypothesis space being searched might not get the accurate target function while an ensemble model can produce a good approximation.

1.5.2.1 Need of Ensemble Machine Learning

The process of ensemble the models, is an effective approach to achieve highly accurate model by combining less accurate ones [18]. To solve different problems or all the cases of a given problem, there isn't one best machine learning model [22]. To improve the performance of machine learning models, many techniques are used which include efficient preprocessing of the data, collect large number of features, perform feature selection task to get relevant ones, explore different machine learning models and if results are not desirable, combine the less accurate models. In ensemble model, more than one opinions are there for single instance. Thus, if one model fails to predict the correct output, there is chance that the other models predict it correctly [23].

There are two errors in the trained model including bias error and variance [16]. Bias error quantifies that on an average the predicted values are differ from the targeted values. High bias represents that the model is under performing which means it has missed some essential trends. On the other hand, variance is used to quantify that the prediction produced on same observation differ from each other. High variance means model is over fitted and will produce adverse predictions on instances except training dataset.

To deal with these errors, ensemble approach is an efficient way [16]. There are three methods to combine the models which include bagging, boosting and stack generalization. The ensemble model is a meta-algorithm which is combination of different models and in order to minimize the variance bagging ensemble approach is used, to minimize the bias boosting ensemble approach is used, or to enhance the predictions stack generalization approach is used.

1.5.2.2 Techniques of Ensemble Machine Learning

Ensemble approach is beneficial to enhance the model's performance. There are three ways to combine the different models and are explained below:

1. **Bagging:** Bagging means bootstrap aggregation which is a simple and successful method to ensemble the models. It is used to improve the unstable classification problems. For instance, weak models like decision tree can fluctuate when any training point changes its position and

may become a different tree. This ensemble method can be applied to other models as well. Bagging method is beneficial for the huge and high dimensional datasets. It is introduced by Leo Breiman [24] to minimize the variance of the model. In bagging, outputs of n models are aggregated which are generated by using N bootstrap sets as shown in Figure 1.8. These sets are generated by using complete dataset via feature selection and random method with replacement. The parallel training of each model is possible because the training of each model is independent. In the end, averaging of outputs is performed where outputs are produced by each bootstrap set.

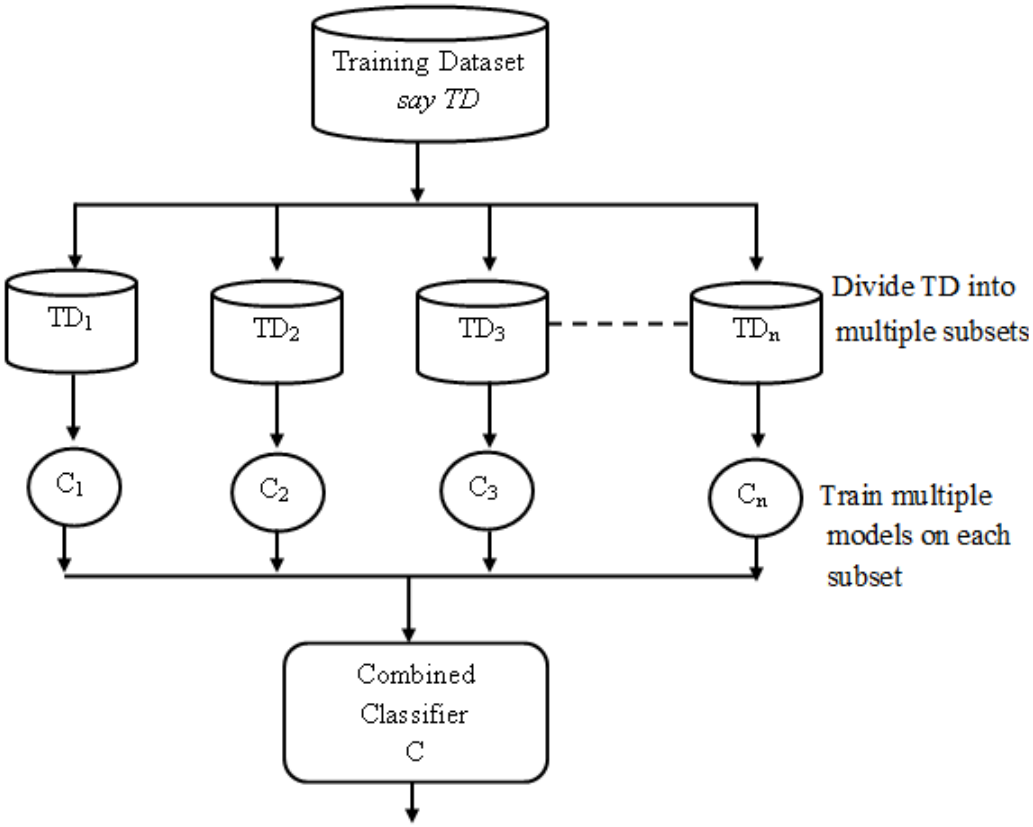

Figure 1.8: Bagging ensemble method.

2. **Boosting:** Boosting is introduced by Schapire [25] which is an ensemble technique to boost the performance of weak models and then group into a strong model. It facilitates the sequential training of the models. First model is trained on the complete dataset while other models get trained by using training sets. These sets are based upon the output of the previous ones. The incorrect instances are extracted to increase their weights. So that, these instances have high chance of appearing in the training dataset which is used by next model. By using this approach, different models are well trained on different sets of the data which helps the ensemble model

to produce enhanced results [26].

3. **Stacking Generalization:** Stacked generalization is a different approach of combining the multiple models. It is used to combine different models like neural network, decision tree etc. It mainly composed of two levels, where level 0 has base learners and level 1 has other models as represented in Figure 1.9. Different models are used in Level 0 which are trained on the dataset. The predictions of each models are combined to create a new dataset. Newly generated dataset has the original values of each instance that it is suppose to predict. This dataset with original values of instances are used by level 1 models and provide final output [26].

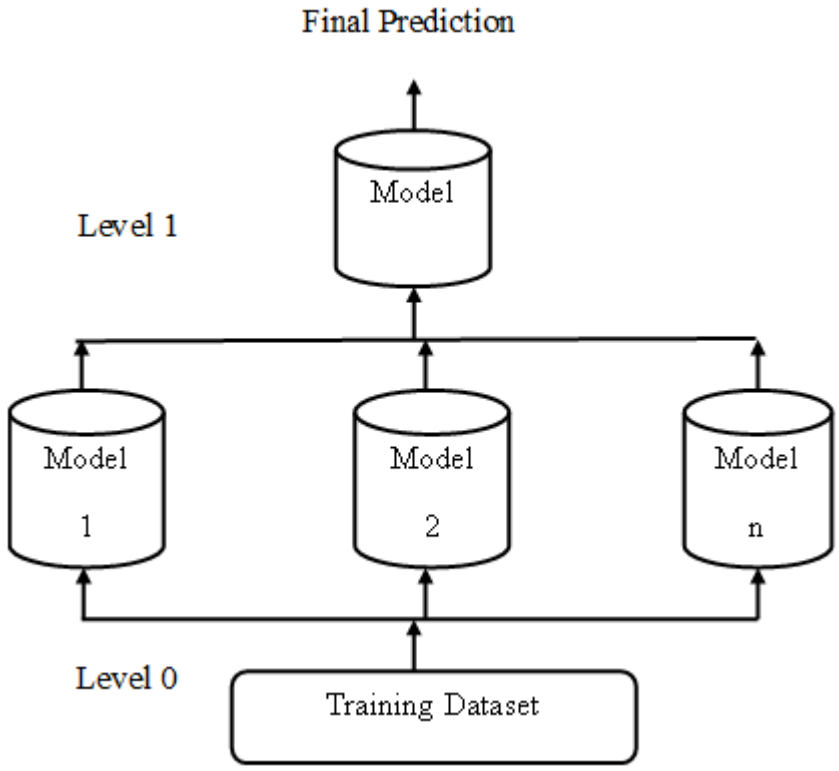

Figure 1.9: Stacking generalization ensemble method.

1.6 Machine Learning Models used in this Thesis

The brief detail of models which are used in proposed ensemble models are given below:

1. **Decision Tree:** Decision tree is one of the supervised learning algorithms which is used to solve the classification problems. It can be used for categorical and continuous input and output variables. While constructing the decision tree top-down approach is considered. Each branch node in decision tree shows a preferred attribute from the given attributes and each

leaf node shows a decision or output. To select the nodes, entropy and information gain are calculated which are discussed below:

- **Entropy:** It shows the degree of disorganization in the data. In other words, it measures the randomness in data. If the number of positive and negative instances are equal then entropy is 1. Otherwise, it is between 0 and 1. For example, while tossing a coin, the probability of getting head is 0.5 and probability of getting tail is also 0.5. Here, entropy is high and there is no alternative to find what will be the output. On the other hand, assume heads on both the sides of a coin, now entropy of this event is predictable and it's output is heads. The entropy of this event is zero because there is no randomness in the data. 1.1 is used to calculate the entropy:

$$Entropy = \sum_i -p_i \times \log_2 \times p_i \tag{1.1}$$

  p_i is the probability of class i. Calculate it as the proportion of class i in the set.

- **Information Gain:** It measures the relative modification in the entropy with respect to the input or independent attributes. In other words, it measures the expected decrease in entropy. It is used to determine the decision node from the given attributes. The best option to decrease the depth of decision tree is to delete attribute which has repeated decrease in entropy. To find the root node from the given attributes, the information gain of each attribute is computed. The attribute which has highest information gain is used as the root node. 1.2 is used to calculate the information gain of each attribute:

$$Information\,Gain = Entropy\,(Entire\,Set) - [Average\,(Entropy\,(Each\,Split))] \tag{1.2}$$

2. **Averaged Neural Network (avNNet) model:** The neural network model has layered structure. Each layer has number of nodes known as neurons. Bottom layer contains the number of inputs or predictors and the top most layer contains the outputs or dependent variable. In between these two layer, there might be intermediate layers known as hidden layers. AvNNet model is

the combination of neural networks in which many neural networks are averaged. The idea behind combining the neural networks can be related to the random forest model which is obtained by averaging of many decision trees. AvNNet model trains many neural networks on the same dataset and final output is obtained by averaging the predictions from all the trained neural networks. The trained neural networks can be different from each other due to random number seeds which is used to initialize the neural network or by training the models by using bootstrapping. In classification problems, the class probabilities are averaged to get the final prediction.

3. **Regularized random forest model (RRF):** The RRF model implements tree regularization framework to random forest and can select a compact feature subset. In other words, regularization limits the depth of the trees to avoid overfitting. To prune the trees, regularization is used. If it is not used then the tree will continue to fit each feature (data point) in different leaf of the tree and this will lead to overfitting. Thus, to generalize the tree, a stopping criteria is required that at which node splitting should be stopped. This can be achieved by mentioning the minimum data points required at each node for splitting. In other words, if a feature d is used for splitting in a tree and information gain of feature d and feature m are same, then the regularization will penalize feature m and will continue with the previously selected feature d. If the regularization is applied on every tree in the forest then this process will reduce the number of features in the forest and it will reduce the dimensionality.

4. **Random Forest (RF):** Random forest generates multiple decision trees and combines them to achieve an accurate prediction. It can be used to regression as well as for classification problems. It has almost same hyper-parameters as a decision tree. While generating trees, it adds the randomness to the model by growing every decision tree with a random subset of features. While splitting a node, it selects random subset of features and searches for the best feature from them instead of looking for the most essential feature. Therefore, a random subset of features is considered by it to split a node. To produce final prediction, it takes an average of all the generated decision trees predictions. RF can be used to select important features.

5. **Support Vector Machine (SVM):** SVM generates the hyperplane which divides the whole data

into classes. It is an algorithm that receive data as input and as an output a line is produced which divides these classes. The points close to the hyperplane from both the classes are known as support vectors. After this, distance between this hyperplane and support vectors are computed and is called as margin. The main objective of the SVM is to maximize this margin. The hyperplane is the optimal hyperplane by which margin is maximised. Therefore, SVM creates a decision boundary in such a way which separates the two classes as far as possible.

6. **Neural Network (NN):** NN works on the basic architecture and behaviour of the human brain. There are neurons in the human brain which process and transmit the information to each other. Dendrites are there to receive the inputs and on these input, an output is produced which is transmitted to other neuron with the help of axon. In NN, network contains artificial neurons known as nodes which process the information and does operations. NN contains three layers which includes input layer, hidden layer and output layer. Input layer takes huge amount of input data such as text, audio, image pixels, numbers, etc. In hidden layer, pattern analysis, mathematical operations, feature extraction etc are performed on the input data. There can have more than one hidden layer in the network. The output is generated by the output layer. NN has many parameters and hyper-parameters which generates the output. These parameters include biases, number of neurons, weights, learning rate, etc. Every node in a network has weights with it and transfer function calculates the weighted sum of the inputs and also adds bias into it. These results are act as an input to the activation function which will further decide the nodes to get fired. The selection of activation function type depends upon the required output. Some of the activation functions are used in the hidden layer or in output layer or in both layers.

7. **Extreme Learning Machine (ELM):** ELM model consists of number of hidden neurons where input weight is given randomly. It uses concept of randomness and perceptron model to solve particular type of problem. It can have single or multiple layers. It is a feed-forward neural network which means that the data travels only in one way via series of layers. In ELM, there is not any requirement of tuning parameters. ELMs are useful in clustering, classification and regression problems.

8. **Blackboost:** Blackboost model creates regression trees as the base classifiers. It is a boosting algorithm which optimizes the loss functions by using gradient boosting.

9. **Generalized Additive Models (GAM):** GAM is an extension of generalized linear models (GLMs) [27]. To improve the prediction quality of the dependent variable from different distributions, is one of the motive of GAM. This is done by approximate the non-parametric functions of predictor variables where these are linked with dependent variable through link function.

10. **GAMBoost :** Generalized additive model is applied by using boosting approach which is based upon the component-wise base classifiers. It is well-suited for the models with many predictors which are possibly non-linear.

11. **Gradient Boosting Machine (GBM):** GBM builds additive model in forward step-wise procedure by applying gradient descent. It can be used for classification and regression problems. The advantage of GBM is its boosting feature which means that it is used to optimize a given cost function.

12. **GLMBoost :** Generalized linear model (GLM) is applied by using a boosting approach which based upon component-wise univariate linear models. GLM is the extension of general linear models. Generalization means that rather finding linear relationship between dependent and the independent variables, it facilitates the dependent variable to be connected with independent variable via link function.

1.7 Performance Evaluation Parameters

To analyze the performance of the proposed ensemble models and each individual model, evaluation parameters like Gini, accuracy, area under the curve, specificity and sensitivity are used. To get an optimized result which is based upon the combination of these evaluation parameters, TOPSIS a multiple criteria decision making method has been used. It generates score by using these evaluation parameters and rank each model according to this score. In this study, models are evaluated on all these parameters which are explained in upcoming section:

1.7.1 Gini Coefficient

Gini coefficient is measured to calculate inequality in the distribution. It can be derived from area under curve (AUC) receiver operating characteristics (ROC) number. Gini coefficient is a ratio between area in between the ROC curve and diagonal line and the area of complete triangle. ROC curve plot represents two evaluation parameters that are sensitivity (True Prediction Rate) and 1-specificity (False Positive Rate). Thus, Gini coefficient shows the inequality between these two evaluation parameters. If the number of negatives is large, then there is an issue in the efficiency of model. Its value lies between 0 and 1. Value 1 means inequality and value 0 means equality. For example, if a model scores Gini value 60% then it is considered as a good model.

$$Gini = 2 \times AUC - 1 \tag{1.3}$$

1.7.2 AUC

To check the quality of the model, AUC is calculated. Basically, AUC is the complete 2D area under the ROC curve which is from (0,0) to (1,1). High AUC value depicts the good quality of the model. Its value lies between 0 and 1. The model has AUC value near to 1 means its quality is good.

1.7.3 Accuracy

Accuracy (ACC) is a metric to evaluate the machine learning models. It is used to determine which model is correctly learn the patterns and relationships between features in a dataset based upon the training dataset. In simple words, It measures the correct predictability of the model. A high accuracy model doesn't mean that the model is predicting all the instances correctly because it can be misleading. For instance, in class imbalanced problem, a model may predict the instances of majority class for all the predictions and scores high accuracy. Therefore, to check the model performance other parameters should be considered. The accuracy of the model is calculated as follows:

$$ACC = \frac{TP + TN}{TotalData} \times 100 \tag{1.4}$$

1.7.4 Sensitivity

Sensitivity (Sens) or recall has been calculated to check the true prediction rate of proposed ensemble models. It is the proportion of actual positives which are correctly identified as positives by the model and is computed as follows:

$$Sens = \frac{TP}{TP + FN} \tag{1.5}$$

1.7.5 Specificity

Specificity (Spec) is the ability of model to identify negative results. It has been calculated to check the true negative rate of the proposed ensemble models and is computed as follows:

$$Spec = \frac{TN}{TN + FP} \tag{1.6}$$

TN: Non-epitopes are classified as Non-epitopes *i.e.* true negative,

TP: epitopes are classified as epitopes *i.e.* true positive,

FP: Non-epitopes are classified as epitopes *i.e.* False positive,

FN: epitopes are classified as Non-epitopes *i.e.* False negative.

1.7.6 Technique for Order Preference by Similarity to an Ideal Solution

Technique for order preference by similarity to an ideal solution (TOPSIS) [28, 29] is one of the multiple criteria decision making methods. This technique is useful for decision makers to structure the problems to be solved, conduct analyses, comparisons and ranking of the alternatives. In other words, it is used to find out the combined solution which involves multiple criteria. In this study, a R package named TOPSIS is used to get optimized result by using evaluation parameters. Rather giving importance to one evaluation parameter, all the evaluation parameters are considered to generate the TOPSIS score which is used to rank all the individual and proposed ensemble models.

1.8 Thesis Organization

This thesis contributes in the field of antigenic epitope prediction by designing and implementing the ensemble models. B-cell, antibody and T-cell are used separately to predict antigenic epitopes. Different ensemble models are proposed for the predictions which are explained in Chapter 3, Chapter 4 and Chapter 5. The complete Chapter-wise summary of the thesis work is given as follows:

Chapter 1 : This Chapter introduces background, problem statements, research gaps and research objectives. The human immune system, innate immune response, adaptive immune response, antigen recognition, Cytotoxic T-lymphocyte, Helper T-lymphocyte and B-cell are described in detail. The definition of machine learning, categorization of machine learning which includes supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning are described in this Chapter. The introduction to ensemble machine learning, various type of ensembling, need of ensemble machine learning and its requirement in prediction of antigenic epitopes are also presented.

Chapter 2 : It reviews the research work in antigenic epitopes which includes humoral immune response, immunoglobulins and cellular immune response. This Chapter also contains the literature in prediction of antigenic epitopes using machine learning which comprises of prediction models for B-cell epitopes, immunoglobulins and T-cell epitopes.

Chapter 3 : It presents the multilevel ensemble model to predict IgA and IgG antibodies. This Chapter contributes in the prediction of immunoglobulins (IgA and IgG). It describes the methodology to develop the multilevel ensemble model, feature extraction approach and feature selection technique. The regularized random forest is used to find out the important features. The ensemble model is a combination of seven different machine learning models and is able to predict variable length of epitopes (4 to 50-mers). Repeated 10-fold cross validation is performed to check the consistency of it's predictions. The proposed model is compared with the IgPred model [30] which is an existing server. It is important to predict antibody specific class to test immune system, find out the infection, recognition of allergy and other illnesses in the body.

Chapter 4 : It describes the ensemble model to predict T-cell mycobacterium tuberculosis

(M.tuberculosis) epitopes. Available servers like NetMHC 2.2 [31], NetMHC 2.3 [32], NetMHC 3.0 [33] and NetMHC 4.0 [34] etc predict binding capacity of peptides. Challenge for the above servers is to classify whether a peptide is a M. tuberculosis epitope or a non-epitope. The existing CTLpred [35] server does the classification task but it is bounded to the peptide length of 9-mers. Thus, these limitations of the servers are overcame by using proposed model which is a direct method to predict M. tuberculosis epitope or non-epitope. The proposed model is capable to predict variable length of epitopes. In this Chapter, computational methods are used to classify T-cell M. tuberculosis epitopes. The set of important features are filtered out by using caret package. The proposed ensemble model is a combination of three different models which is used to predict M. tuberculosis epitopes of variable length (7 to 40 mers). The robustness of proposed model is analysed by doing repeated k-fold cross validation. This Chapter also demonstrates the validation and comparison of proposed model with existing servers.

Chapter 5 : This Chapter explains the improved ensemble prediction model of antigenic epitopes. It is desirable to develop a reliable model with significant improvement in prediction models. To extract the important features, boruta is used. In this Chapter, a hybrid model has been proposed by using stacked generalization ensemble technique for prediction of linear B-cell epitopes. The goal of applying stacked generalization ensemble approach is to refine predictions of base classifiers and to get rid of the worse predictions. The proposed model is combination of six different machine learning models and is used to predict variable length of epitopes (6 to 49 mers). The trained ensemble model has been tested on the benchmark dataset and compared with existing sequential B-cell epitope prediction techniques including APCpred [36], ABCpred [5], BCpred [37] and AAP_{BCPred} [38].

Chapter 6 : This Chapter concludes the major findings and prime contributions of the thesis and describes the feasible future research directions.

1.9 Thesis Contribution

In this thesis, ensemble models have been developed to predict the antigenic epitopes. The major contributions of the thesis are mentioned below:

1. Prediction of the antigenic epitopes can be achieved by using B-cells and T-cells. In this thesis, B-cells, antibody IgG and IgA and T-cells are used separately for the prediction of antigenic epitopes.

2. The preprocessing of the data which includes data cleansing, class balancing, feature extraction and feature selection is performed.

3. The feature selection process is performed by using different techniques which includes regularized random forest, caret and boruta.

4. A multilevel ensemble model is combination of seven different models and is used to predict antibodies IgG and IgA epitopes.

5. To predict T-cell mycobacterium tuberculosis epitopes, an ensemble model is proposed which classifies a peptide is an epitope or a non-epitope rather predicting its binding capacity. While training and testing the models, data partitioning is performed in such a way that all the models are able to access the whole data. Here, the ensemble model is a combination of three different models.

6. Prediction of B-cell epitopes is done by the proposed ensemble model which is combination of six different models.

7. The three different ensemble models are efficient to predict variable and fixed length of epitopes.

8. The existing tools, prediction of antigenic epitopes is dependent upon single model's outcome but in this thesis, three different ensemble models are proposed. In each ensemble model, different machine learning models are used. Here, boosting and stacked generalization techniques are used to develop the different ensemble models for prediction of antigenic epitopes.

Chapter 2

Literature Review

This Chapter reviews the research work in antigenic epitopes which includes humoral immune response, immunoglobulins and cellular immune response. This Chapter also contains the literature in prediction of antigenic epitopes using machine learning which comprises of prediction models for B-cell epitopes, immunoglobulins and T-cell epitopes.

2.1 Antigenic Eptiopes

The two major mechanisms of the adaptive immune response are humoral and cellular immunity which are mediated by B-lymphocytes and T-lymphocytes respectively [39]. These two immune responses work differently to identify the antigens. B-cell receptor consists of immunoglobulin and can bind to soluble antigen in a similar way to many well-defined receptor-ligand systems.

On the other hand, the antigen receptor of T-cell can commonly identify antigen only in the association of cell surface molecules encoded by the MHC. This is called as MHC restriction which ensures the activation of T-cell occurs only in an relevant cellular situation. Thus, activation of T-cell for specific antigen results from the creation of ternary complex which includes nominal antigen, TCR and MHC I or MHC II molecules [40, 41].

The presence of an antigen initiates the activation of B-cell or T-cell. Antigens are generally peptides, proteins or polysaccharides. To create complex antigens, nucleic acids and lipids can attach to these molecules. An epitope or antigenic determinant is a particular portion of an antigen to which antibody gets attached. On the other hand, the portion of an antibody which binds to an epitope is known as paratope and is shown in Figure 2.1. When an antigen provokes the antibody response in the body, it doesn't attach to the whole protein, but only to the epitope. There is a possibility that an antigen has more than one epitope with which different antibodies will bind. Antigens are categorized as endogenous *i.e.* generated within cells, exogenous *i.e.* entering from outside, an auto-antigen, a native antigen or a tumor antigen. The host cells are able to identify an antigen via antigenic

specificity, like the bond between paratopes and epitopes. To protect the body, humoral and cellular immunities of adaptive immune system are required which are described in the upcoming section.

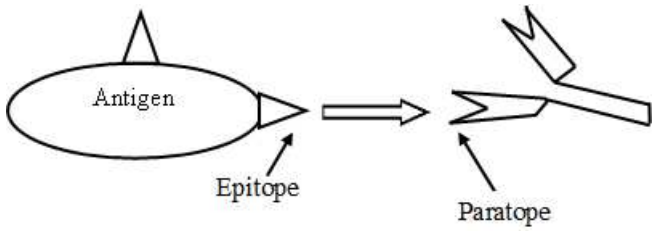

Figure 2.1: Representation of epitope and paratope.

2.1.1 B-cell - Humoral Immune Response

Membrane-bound immunoglobulins (mIgs) are the specific receptors by which B-cell epitopes bind on the surface of B-cells. Such interactions cause to a number of events which stimulate the growth of the specific B-cell population. Activation of B-cells lead to the generation of antibodies having similar binding features as those of the B-cell receptors. It is known as the humoral immune response and is shown in Figure 2.2.

Fundamentally, soluble molecule which can bind to mlgs are potential B-cell antigen. However, features defined via B-cell receptors decide the ability of these potential B-cell epitopes [42]. On the B-cell surface, B-cell epitopes can directly interact with mIgs. There isn't any requirement of antigen processing for these interactions. Therefore, B-cell receptor can bind to antigen in its native structure. This is the reason of identifying the discontinuous epitopes by the antibodies corresponding to the native proteins. The discontinuous epitopes are those epitopes which have amino acids in non-linear fashion but are near in the folded structure. Antibodies corresponding to the synthetic peptide antigen or denatured proteins have ability to identify continuous epitopes. But can't identify the corresponding protein in its native structure.

Antigen which can evoke the B-cell responses are entirely different. It can be lipids, proteins, carbohydrates or small molecules. It is opposite to the T-cell epitopes which are generally small peptide fragments. The diversity of B-cell antigen makes it challenging to predict B-cell epitopes. When B-cell sends a signal, mIgs bind to antigen which causes the activation of B-cell. The presence of T-cells is not always required in the activation of B-cells. T-dependent or Thymus-dependent antigens are those antigens which require T-cells to activate B-cells. On the other hand, T-independent

or Thymus-independent antigens don't require the T-cells for the stimulation of B-cells. In most of the cases, antigens are T-dependent and T-independent antigens are mostly polymeric molecules which have repeated epitopes. For instance, thymus independent antigens contain microbial products including polymeric proteins and polysaccharides.

The thymus dependent B-cell antigen, induce humoral response only in the presence of T-cells. B-cells take many days to respond these antigens, however antibodies produced have a high affinity and are more versatile in functioning as compared to thymus independent activation. On the other hand, thymus independent B-cell antigens, stimulate B-cells via cross linking of surface receptors. In the absence of T-cells contribution, B-cells don't do immunoglobulin gene class diversion and this results in the generation of IgM antibodies. On the other hand, thymus dependent B-cell antigens are generally mature to contain large portion of IgG antibody.

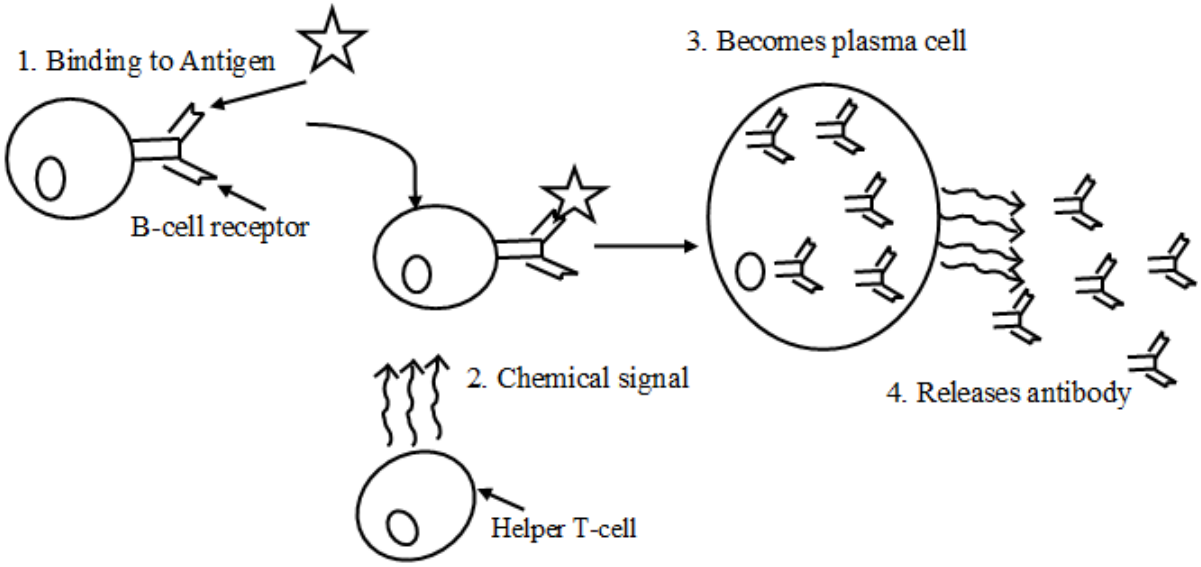

Figure 2.2: Representation of humoral immune response.

2.1.1.1 Immunoglobulins

The paratope is a part of Ig which binds with the part of antigen *i.e.* epitope, this process is known as immunoglobulin antigen interaction. In the soluble form, Igs are generated in vivo process against the complete antigens. In this manner, surface epitopes are recognized which describes their conformational structures and these are non-linear in the primary sequence of antigens. The benefit to recognize the portions of antigen independently makes possible for the B-cells to differentiate between two closely linked antigen. These are considered as group of epitopes. This allows the

same antibody to attach with the different antigens which have same epitopes, it is called as cross reactivity [43]. The structures of all these Igs are different from each other as represented in Figure 2.3. There are five classes of immunoglobulin and are explained below:

1. **IgM:** IgM is primary Ig which is generated during development of B-cells [43]. The first response of immune system is the generation of IgMs which are usually considered to identify acute exposure to pathogen. IgM antibodies are more poly reactive as compared to other isotypes. Thus, this facilitates the IgM-bearing B-cells to react instantly to the different antigens. These are also known as natural antibodies because of its low affinity property. Some natural antibodies contribute as a first defense line and also participate in immuno-regulation process. These antibodies may respond to auto-antigens, but are infrequently responsible for auto-immune diseases.

2. **IgD:** IgD is antibody which is generated in the secreted form and are present in blood serum. It presents in a small amount. This consists of two heavy chains of class delta and two light chains. IgD antibodies participate in the immune response by passing signal for the activation of B-cells. These activated B-cells are then ready to participate in the defense system.

3. **IgG:** IgG antibody is the most important isotype in the body. It circulates in blood and is in huge amount as compared to other isotypes. These are created and triggered by the plasma B-cells. These are broadly researched isotype of immunoglobulins. It has four subgroups including IgG1, IgG2, IgG3 and IgG4 and all have different functions. For protein antigens, IgG1 and IgG3 are usually induced and for polysaccharide antigens, IgG2 and IgG4 are activated. IgG participates directly in immune system by neutralizing toxins and viruses.

4. **IgA:** IgA is antibody whose amount is large as compared to IgM but lower than IgG. At mucosal surfaces the level of IgA is higher than IgG, like in breast milk and saliva [44]. It has two subgroups including IgA1 and IgA2. The protection from viruses, toxins and bacterias at mucosal surfaces is important and is done by intracellular IgA.

5. **IgE:** IgE are the antibodies which are created by the immune system to defend the body from virus, bacteria and allergy. It is generally in small amount and is present in the blood. The large amount of it indicates that the body is overacting to the allergens which lead to allergic reaction.

It's level is also high when body fights with the infections. Currently, anti-IgE antibodies are developed for the treatment of asthma and allergy [45].

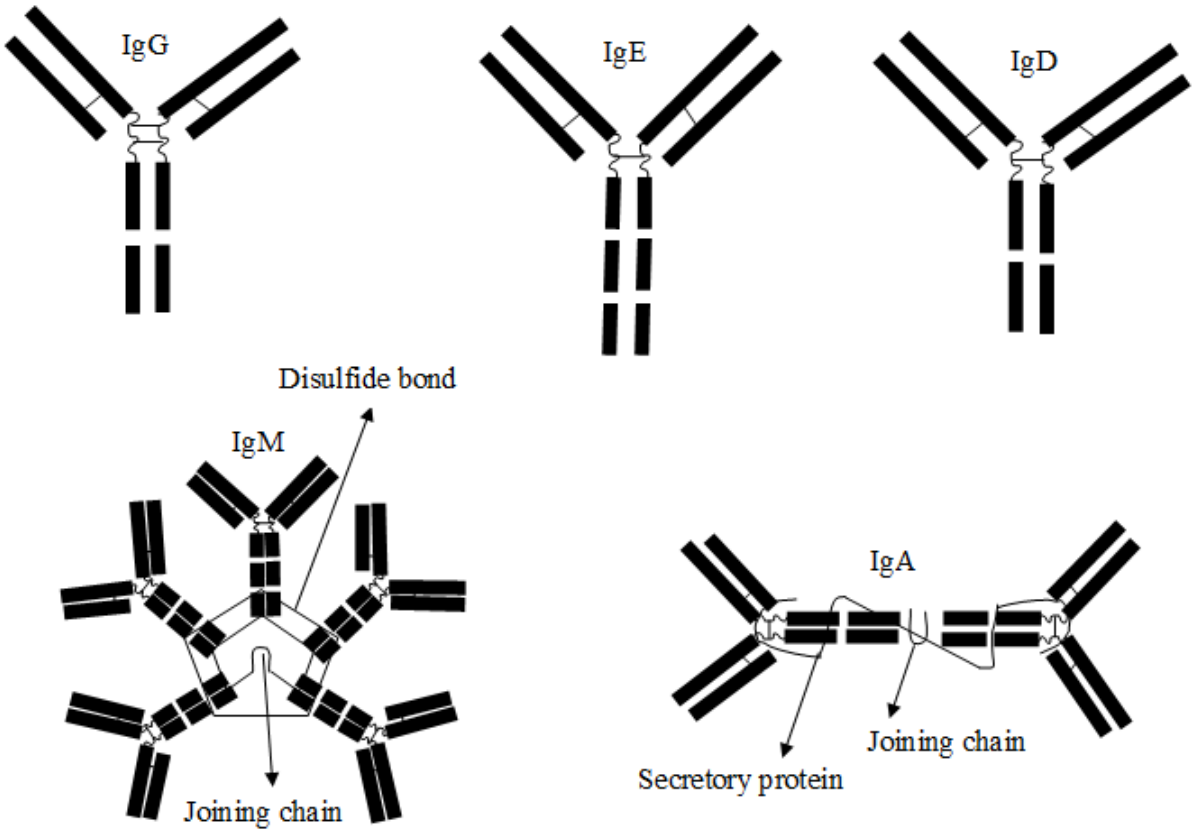

Figure 2.3: Structures of immunoglobulins classes.

2.1.2 T-cell - Cellular Immune Response

T-cells epitopes bind with MHC molecules and TCRs. Thus, epitopes of T-cells are determined by specificity of TCRs and MHC. MHC I and MHC II have particular binding regions for the small peptide fragments. The processing of protein antigens is performed by intracellular proteases before these gets attach with mIgs and antibodies. TCRs has complementarity determining regions (CDRs) which are physically connected to the peptide MHC complexes. Cytokine mediated mechanism activates the T-cells which is a result of tripartite interaction including MHC molecule, TCR and peptide. The output of T-cell activation is dependent upon the class of MHC molecules and the type of T-cells. The peptides bind to MHC I generally communicate with the TCRs on the cytotoxic cell's surface. Such interactions lead to the killing of cells via activation and expansion of cytotoxic T-cells, this process is known as cellular immune response. The peptides bind to MHC II, which

are present on the APCs including B-cells, dendritic cells and macrophages. These peptides interact with the TCRs on the helper T-cell's surface and results in the activation of helper T-cells. Then, it releases cytokines to activate macrophage and/or differentiation of B-cells into plasma cells which will produce antibodies [42].

Antigenic epitopes of T-cells are available on the surface of APC and bind with MHC to generate immune response. Both the MHC I and MHC II bind to different length of peptides. The peptides length presented by the MHC I are between 8 to 11 amino acids and MHC II presents peptides length of 12 to 25 amino acids. Oligopeptide (peptide contains small number of amino-acid residues) fragments which are acquired via proteolysis of invader antigens bind to the MHC II molecules. They show them at the surface of cell which is identified by the helper T-cells or $CD4^{+}$ T-cells.

Figure 2.4 represents the T-cells with MHC I and MHC II molecules. T-cell produces the adaptive immune response corresponding to the invader, when sufficient amount of epitopes are showcased. MHC II are expressed APCs. MHC I are available on each nucleated cells in the body. The identification of antigenic epitopes via T-cells and the production of immune response have important role in the immune system of every individual. The minute changes from its original functioning can affect seriously to the organism. In auto-immune diseases, T-cells identify the native peptide of cell as pathogen. Thus, it attacks on it and finally kills the own tissues of organisms.

Therefore, information about epitopes of the peptides has an important role in manufacturing vaccines based on epitopes which then injected into the patient that will generate immune response [46].

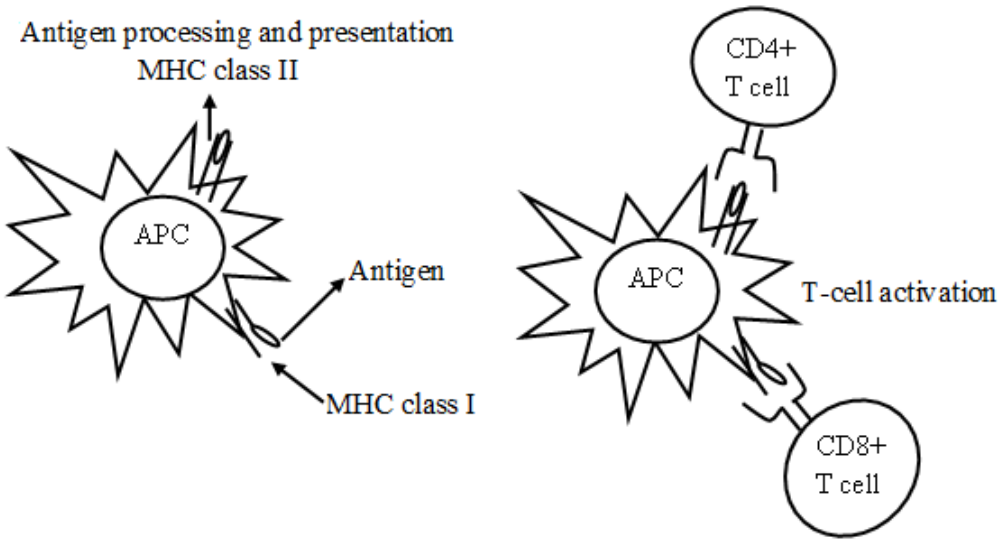

Figure 2.4: Representation of cellular immune response.

2.2 Prediction of Antigenic Epitopes using Machine Learning

Antigenic determinants are majorally concentrated by the clinical and biomedical researchers because they play essential role in the designing of vaccine, prevention of disease, diagnosis and treatment. Bioinformatics is an area of science where many different disciplines including computing, information technology and biology are combined to arrange and store huge amount of biological data. Such data is generated by the molecular biology, biotechnology and genetics [47] etc. The major objectives of bioinformatics is to well-organise and describe the data which is collected from the transcriptome, proteome and/or genome [48]. The aim of such discipline is to improve health benefits which includes field of vaccines, antibodies production, diagnosis tests etc.

2.2.1 Prediction Models for B-cell Epitopes

Antigenic epitopes of B-cells are identifies by antibodies or B-cell receptors in their native form. The prediction of T-cells epitopes are similar to the B-cell epitopes. Because these both are based upon the features of amino acids like charge, hydrophilic, hydrophobic, secondary structure and exposed surface area. For prediction of discontinuous B-cell epitopes, 3D structure of the invader antigen is required [49–51].

The propensity scales such as hydrophilicity [52], antigenicity [53] and surface accessibility [54] were used to predict sequential B-cell epitopes. Traditionally, single property of amino acid was used to describe the information of sequence. Later on, more than one physicochemical properties had been employed in the methods like PREDITOP [55], PEOPLE [56], BEPITOPE [57] and BcePred [58].

To predict linear B-cell epitopes, the Bcepred tool was based on physicochemical properties such as hydrophilicity, flexibility, polarity, and exposed surface on a non-redundant dataset. The dataset consists of 1029 B-cell epitopes obtained from Bcipep database and an equal number of non-epitopes obtained randomly from Swiss-Prot database. The prediction accuracy for models based on these properties varies from 52.92% to 57.53%. The performance of models were enhanced by using physicochemical properties in contrast to the techniques those used single property.

ABCpred server, which was based on NNs, had an estimated accuracy of 65.93% [5]. These networks were trained and tested on a clean dataset, which consists of 700 non-redundant B-cell epitopes and equal number of non-epitopes. They had used fixed length of epitopes (20-mers) to

train the models. BepiPred predicted the location of linear B-cell epitopes using a combination of a hidden markov model and a propensity scale method [59]. These servers are easy to use and properly organized.

Chen [38] used 872 epitopes with a same length of 20 residues and 872 non-epitope to prepare support vector machine (SVM) with 400 features and their model achieved accuracy of 71.09%. BCPred used SVM and a string kernel [37] to predict linear epitopes. To train the model, 701 linear B-cell epitopes and 701 non-epitopes were used. The model scored AUC value 0.758.

COBEpro [60] which utilized a two-stage design where SVM was trained on novel sequence similarity scores as inputs and predicted variable-size peptides in the first stage. In second stage, these fragments were combined to predict epitopes in full chains. The COBEpro was extended to predict conformational epitopes via its second stage. BayesB method [4] which predicted epitopes of diverse lengths (12 to 20-mers) by using position specific scoring matrix (PSSM).

BEST [61] was composed of two-stage scheme which predicted conformational and linear epitopes from the antigen chains based on accurate predictions of linear epitopes from the first stage. They used dataset of 20-mers epitopes to train the SVM model for prediction and attained AUC values 0.81 and 0.85. The SVM based model [3] predicted antigenic epitopes by using tri-peptide similarity and propensity of amino acid. It attained AUC value 0.702.

Huang [1] used RF model to predict the linear B-cell epitopes and scored accuracy of 78.31%. To train the model, 2479 continuous epitopes from over 1000 antigenic proteins were used. DMN-LBE [62] used a sequence-based linear B-cell epitope predictor which used deep maxout network (DMN) and dropout training approaches. To minimize the training time of the classifier, graphics processing unit (GPU) was used. It achieved accuracy of 68.33% with AUC 0.743. For linear B-cell epitope prediction, Weike Shen [36] had proposed APCpred method, which used amino acid anchoring pair composition (APC). The SVM model of 20-mers epitopes achieved accuracy of 68.43%. Existing servers to predict continuous B-cell epitopes are available on web which are mentioned in Table 2.1.

2.2.1.1 Prediction Models for Immunoglobulin class

Numerous methods have been developed for predicting antigenic regions or B-cell epitopes which can induce B-cell response. Algpred [63] predicted the allergenic proteins based on four different approaches and scored accuracy of 85.02%. First approach, SVM was used to predict allergens based

Table 2.1: List of linear B-cell epitope prediction servers.

Server Name	Web Server	Models Used for Prediction
BepiPred [59]	http://www.cbs.dtu.dk/services/BepiPred/	Hidden Markov Model
ABCpred [5]	http://www.imtech.res.in/raghava/abcpred/	NN
BCPred [37]	Not Available	SVM
BEST [61]	Not Available	SVM
SVMTriP [3]	http://sysbio.unl.edu/SVMTriP/	SVM
APCpred [36]	http:/ccb.bmi.ac.cn/APCpred/	SVM
COBEpro [60]	Not Available	SVM
RF classifier [1]	Not Available	RF
DMN-LBE [62]	Not Available	Deep Maxout Network

on amino acid and dipeptide composition of proteins. Second approach, motif based technique was used to predict allergens by using software MEME/MAST [64]. Third approach, segment similarity technique was used. If segment was similar to allergen representative proteins (ARPs) [65], the protein was assigned allergen. Fourth approach, if protein had segment similar to known IgE epitopes then that protein was considered as allergen.

SVM based model and pseudo-amino acid composition were used [66] to predict allergenic proteins (IgE). The efficiency of model had been evaluated by using five-fold cross validation and results were better than the previous work by scoring accuracy of 91.20%. IgPred [30] predicted antigens of specific type of antibodies which utilized the amino acid sequence information for prediction. The model had been used to identify the specific class of antibodies in the antigen by using features like binary profiles, dipeptide composition and amino acid composition. From these features, dipeptide composition-based SVM achieved accuracy of 70.72%, 82.7% and 72.07% for IgG, IgE and IgA specific epitopes respectively. These models had been evaluated using five-fold cross validation.

2.2.2 Prediction Models for T-cell

The pathogenic mycobacterium which causes TB survives within the cells. Thus T-cells are required to fight against bacteria rather than antibodies [67,68]. For correct functioning of the immune system, both humoral and cell mediated immunity work together directly or indirectly depending upon the occurrence of foreign invader. Therefore, there are numerous methods for predicting T-cell and B-cell epitopes which help in designing of epitope based vaccines, immune-diagnostic and antibody production.

For prediction of T-cell epitopes with MHC type I restricted T-cell clone, SVM was used with cross validation and scored sensitivity 0.763 [69]. Neural network with other methods like binding motifs, molecular modelling, quantitative matrices and hidden markov models was used for recognition and prediction of T-cell epitopes and MHC-binding peptides [70]. CTLpred [35] method was trained and tested on T-cell epitopes and non-epitopes including 1137 experimentally proven MHC class I restricted T-cell epitopes. This method was based on quantitative matrix (QM) and machine learning techniques such as SVM and NN. This scores an accuracy of 70.0, 72.2 and 75.2%. An improved NN model [71] was used to predict T-cell class I epitopes which combined several neural networks and hidden markov model to get more accurate prediction. A SVM based [72] (SVMHC) prediction of peptides binding to MHC class I molecules was used which scored matthew correlation coefficient (MCC) 0.85 with four-fold cross validation.

Although, many studies exist for predicting binding capacity [73] of M. tuberculosis like interferon-gamma inducing MHC II binders [74] in which they used SVM for prediction, linear B-cell epitopes were used for prediction of M. tuberculosis epitopes [75], NetMHC 2.2 [31] server used NN to predict peptide binding capacity of MHC class II , NetMHC 2.3 [32] server was an improved version of NetMHC 2.2 server which predicted the binding capacity with HLA class II. This server was constructed by using an enlarged dataset of quantitative MHC peptide binding affinity data extracted from the Immune Epitope Database including HLA-DR, HLA-DQ, HLA-DP and H-2 mouse molecules. In this, it was demonstrated that the training with enlarged dataset increased the performance of peptide binding prediction, NetMHC 3.0 [33] server predicted the binding capacity of HLA class I using NN, NetMHC 4.0 [34] overcame the limitation of NN-Align method [76] that it could detect only fixed length motifs. In NetMHC 4.0, NN was trained on variable length peptides. It was compared with the fixed length methods and had proved to be an effective method and NetMHCpan 3.0 [77] server predicted the binding capacity of HLA class I using NN and hence improved the accuracy for prediction of peptide binding and recognition of MHC ligands. These servers provide weak and strong binders but are not capable of classifying M. tuberculosis epitope or a non-epitope. Different T-cell eptitopes prediction servers, model used for the predictions and output of predictions are listed in the Table 2.2 .

Table 2.2: List of T-cell epitope prediction servers.

Server Name	**Web Server**	**Models Used for Prediction**	**Prediction Target**
SVMHC [72]	Not Available	SVM	MHC I binding peptides
Hidden markov based model [70]	Not Available	NN	MHC-binding peptides and T-cell epitopes
CTLPred [35]	www.imtech.res.in/ raghava/ctlpred/	SVM and NN	T-cell epitopes
Improved NN [71]	Not Available	NN	MHC I epitopes
SVM based model [74]	crdd.osdd.net/ raghava/ifnepitope/	SVM	MHC II peptide binding
NN-align [76]	www.cbs.dtu.dk/ services/NetMHCII-2.0	NN	MHC II binding peptides
NetMHC 2.3 [32]	www.cbs.dtu.dk/ services/NetMHCII-2.3	NN	MHC II binding peptides
NetMHC 3.0 [33]	www.cbs.dtu.dk/ services/NetMHC-3.0/	NN	MHC I binding peptides
NetMHC 4.0 [34]	www.cbs.dtu.dk/ services/NetMHC/	NN	MHC II binding peptides
NetMHCPan 3.0 [77]	www.cbs.dtu.dk/ services/NetMHCpan-3.0/	NN	MHC I binding peptides

Chapter 3

Multilevel Ensemble Model for Prediction of IgA and IgG antibodies

Identification of the antigen for inducing specific class of antibody is prime objective in the peptide based vaccine designs, immunodiagnosis tests, and antibody productions. It's urge to introduce a reliable system with high accuracy and efficiency for the prediction of epitopes inducing IgA and IgG antibodies. In this Chapter, a novel multilevel ensemble model has been proposed for the prediction of epitopes inducing IgG and IgA antibodies. Epitope length is important while training the model and it is efficient to use variable length of epitopes. In this ensemble approach, seven different machine learning models are combined to predict variable length of epitopes (4 to 50-mers). The proposed model of IgG specific epitopes achieves 94.43% of accuracy and IgA specific epitopes achieves 97.56% of accuracy with repeated 10-fold cross validation. The proposed model has been compared with IgPred server and has outperformed the existing system. [†]

3.1 Introduction

The human immune system has an essential mechanism to protect the body which is known as immunoglobulins. It recognizes the specific antigen and blinds with it to protect the body. Antigens can be toxins, bacteria and viruses. When immune system doesn't work properly or not able to generate required antibodies, this is known as immunodeficiency. It can be caused by the side effects of medicines, diseases, infections or lack of proper nutrition. The analysis of particular immunoglobulins' presence in the blood is useful to identify the infections or different illnesses or to do the diagnoses. The types of antibodies are IgM, IgE, IgD, IgA and IgG which are generated by the immune system to fight against the invaders. The amount of Immunoglobulin G (IgG) is huge, followed by Immunoglobulin M (IgM) and Immunoglobulin A (IgA) [78]. The amount of

[†]D Khanna and PS Rana, Multilevel Ensemble Model for Prediction of IgA and IgG antibodies, Immunology Letters, Elsevier 184 (1) (2017) 51-60.

Immunoglobulin D (IgD) is less than IgA, IgG, IgM but higher than Immunoglobulin E (IgE) [79]. IgM and IgG preserve from infections in the internal body tissues, blood and organs. IgA [80] is available in blood, the greater part of the IgA in the body is in the secretions of the mucosal surfaces which encompass respiratory, saliva, tears and gastrointestinal secretions. The IgA antibodies in the secretions play a major role to protect these areas from infections. IgG and IgM are also found in secretions but not equal to the amount of IgA. The human mucosal surfaces are protected by IgA, this shield is very important for the human body and the deficiency of IgA may lead to cancer [81]. IgG the most abundant type of antibody, is found in all the body fluids and protects against bacterial and viral infections [78]. In experimental designs, immunodiagnostic tests and vaccines production [82], the disclosure of continuous epitopes still play a key role because most of the B-cell epitopes are discontinuous.

In the late years, many computational strategies had been produced to predict B-cell epitopes. The propensity scales such as hydrophilicity [52], surface accessibility [54] and antigenicity [53] were used to predict linear B-cell epitopes. Later on, more than one physicochemical properties of epitopes were used to develop the methods such as PREDITOP [55], PEOPLE [56], BEPITOPE [57] and BcePred [58] for the prediction of linear B-cell epitopes. The quality of prediction is enhanced by using physicochemical property as contrast with the techniques which used the single property. ABCPred [5], Chen et al. [38], BCPred [37], SVMTrip [3], Huang [1], Lian Yao [62] and APCpred [36] used machine learning algorithms to predict linear B-cell epitopes. All these models were used to predict B-cell epitopes, not for identifying specific class of antibodies in the antigen.

Numerous methods had been developed for predicting antigenic regions or B-cell epitopes which can induce B-cell response. The researchers used machine learning models for the prediction of antibodies like IgA, IgE and IgG. Like, Algpred [63], SVM based model [66], IgPred [30] predicted specific class of antibodies.

Prediction of antibody specific class is important for testing immune system, finding out the allergy, infection and any other illness. Influenced from the outcome of machine learning approaches and the urge to find an accurate method, a multilevel ensemble model has been proposed to predict B-cell epitopes which can inducing a specific class of antibody (like IgA, IgG). The physicochemical properties of amino acid are used to the train machine learning models which will classify the IgG and IgA inducing epitopes as explained in Section 3.3.

The Chapter has been organized as follows: An overview of the dataset, features measurement, feature selection, machine learning models and benchmark dataset have been presented in Section 3.2. The methodology and proposed model are explained in Section 3.3. Model evaluation has been described in Section 3.4. Section 3.5 narrates experiments, result analysis, comparison and discussion. Finally, conclusion and future work have been presented in Section 3.6.

3.2 Materials and Methods

This section describes the dataset, extraction of features, feature importance, machine learning models and the benchmark dataset.

3.2.1 Dataset and its Features

The balanced dataset consists of IgA and IgG inducing epitopes has been extracted from *http://crdd.osdd.net/raghava/IgPred/*. Total number of IgG epitopes sequences are 16,067, in which number of IgG epitopes are 7,575 and non-IgG are 7,673; IgA epitopes are 403 and non-IgA are 416. All the epitopes have variable length (4 to 50-mers). The glimpse of datasets are presented in Table 3.1 and Table 3.2.

Table 3.1: Sample dataset of IgG epitopes.

SL	**F**$_a$	**F**$_b$	**F**$_c$	**F**$_d$	———	**F**$_z$	**F**$_{aa}$	**F**$_{ab}$	**F**$_{ac}$	**CL**$_{IgG}$
7	111.43	2.61	0.41	71.34	———	-0.03	-0.07	834.89	6.50	1
11	26.36	3.29	0.25	37.40	———	-0.15	-0.25	1246.21	3.23	1
15	90.67	0.09	0.13	54.03	———	0.31	-0.67	1498.66	7.54	1
12	98.33	-0.05	0.34	-4.98	———	0.30	-1.00	1088.23	6.41	0
6	131.67	0.46	0.45	8.33	———	0.22	-0.93	571.68	10.55	0

Table 3.2: Sample dataset of IgA epitopes.

SL	**F**$_a$	**F**$_b$	**F**$_c$	**F**$_d$	———	**F**$_z$	**F**$_{aa}$	**F**$_{ab}$	**F**$_{ac}$	**CL**$_{IgA}$
10	20.00	3.73	0.22	47.87	———	-0.60	-0.03	1176.35	9.44	1
5	118.00	-0.82	0.35	8.00	———	0.44	-0.59	533.64	3.85	1
20	29.50	2.61	0.40	53.96	———	-0.33	-0.08	2267.58	8.52	1
40	66.00	1.30	0.33	28.29	———	0.07	-0.47	4212.79	8.24	0
12	137.50	0.32	0.31	11.07	———	0.19	-0.39	1275.56	11.65	0

3.2.2 Feature Measurement

The Table 3.3 describes the physicochemical properties of amino acid which are used in this Chapter. To extract the properties, R an open source software is used which is licensed under GNU GPL is used.

Table 3.3: Physicochemical properties of amino acid.

SN	Property	Description	R Package	Function	Notations used in present study
1	Aliphatic index	The relative volume occupied by aliphatic side chains (alanine, valine, isoleucine, and leucine) is known as aliphatic index of a protein.	Peptides [83]	aindex	F_a
2	Potential protein interaction index	Based upon amino acid sequence of a protein, the potential protein interaction index is computed which is introduced by Boman [84].	Peptides	boman	F_b
3	Hydrophobic moment	It is computed for an amino acid sequence of N residues and their corresponding hydrophobicities	Peptides	hmoment	F_c
4	Instability index	The stability of protein in a test tube is estimated by instability index.	Peptides	instaindex	F_d
5	Number of possible neighbours	It describes neighbors of degree one for a group of peptide sequences.	Peptider [85]	getNofNeighbors	F_e
6	Tiny	Number of amino acid in the sequence which comes under tiny class.	Peptides	aacomp	F_f
7	Small	Number of amino acid in the sequence which comes under small class.	Peptides	aacomp	F_g
8	Aliphatic	Number of amino acid in the sequence which comes under aliphatic class.	Peptides	aacomp	F_h
9	Aromatic	Number of amino acid in the sequence which comes under aromatic class.	Peptides	aacomp	F_i
10	Nonpolar	Number of amino acid in the sequence which comes under nonpolar class.	Peptides	aacomp	F_j
11	Polar	Number of amino acid in the sequence which comes under polar class.	Peptides	aacomp	F_k

to be cont'd on next page

Table 3.3: Physicochemical properties of amino acid. (cont.)

SN	Property	Description	R Package	Function	Notations used in present study
12	Charged	Number of amino acid in the sequence which comes under charged class.	Peptides	aacomp	F_l
13	Basic	Number of amino acid in the sequence which comes under basic class.	Peptides	aacomp	F_m
14	Acidic	Number of amino acid in the sequence which comes under acidic class.	Peptides	aacomp	F_n
15	Percentage of tiny	Percentage of tiny amino acid in the given sequence.	Peptides	aacomp	F_o
16	Percentage of small	Percentage of small amino acid in the given sequence.	Peptides	aacomp	F_p
17	Percentage of aliphatic	Percentage of aliphatic amino acid in the given sequence.	Peptides	aacomp	F_q
18	Percentage of aromatic	Percentage of aromatic amino acid in the given sequence.	Peptides	aacomp	F_r
19	Percentage of nonpolar	Percentage of nonpolar amino acid in the given sequence.	Peptides	aacomp	F_s
20	Percentage of polar	Percentage of polar amino acid in the given sequence.	Peptides	aacomp	F_t
21	Percentage of charged	Percentage of charged amino acid in the given sequence.	Peptides	aacomp	F_u
22	Percentage of basic	Percentage of basic amino acid in the given sequence.	Peptides	aacomp	F_v
23	Percentage of acidic	Percentage of acidic amino acid in the given sequence.	Peptides	aacomp	F_w
24	Charge of protein sequence	The net charge can be computed at defined pH by using pKa scales.	Peptides	charge	F_x
25	Hydrophobicity	It computes the hydrophobicity index of an amino acids sequence.	Peptides	hydrophobicity	F_y
26	Kidera factor	The kidera factors is derived by applying multivariate analysis to 188 physical features of the 20 amino acids and using dimension reduction techniques.	Peptides	kiderafactor	F_z
27	Molecular Weight	This function calculates the molecular weight of a protein sequence.	Peptides	mw	F_{aa}
28	Isoelectric point	Isoelectric point (pI) is the pH at which a particular molecule or surface doesn't carry electrical charge.	Peptides	pI	F_{ab}
29	Sequence length	Number of amino acids in a sequence.	Peptides	lengthpep	SL

3.2.3 Feature Importance using Regularized Trees

While building the model, feature selection is an essential process which filters the correlated variables, biases and unwanted noise from the dataset. It selects important features which may improve the model performance. In this Chapter, RRF model [86] has been used for the feature selection task. The regularized random forest (RRF) uses one ensemble instead of multiple ensembles. A feature with the highest regularized information gain is inserted at a node based on the instances only at that node. If more than one feature has same regularized information gain, then one of these features is selected arbitrary. RRF model has been implemented in R, python and do the tedious task in an easy way. It gives node impurity which is measured by the Gini index.

According to the RRF algorithm, features F_i, F_j, F_l, F_m, F_n, F_o and F_x are least important for the IgG dataset. When target is changed then it will effect the subset of important features. Thus, the important features for both the datasets are different. For IgA dataset, features F_i, F_j, F_k, F_l, F_m, F_n and F_o are less important. Table 3.4 shows the ranking of features based on node purity which has been computed by RRF. High rank is assigned to the feature which has high node purity. Seven least ranked (24 to 30) features aren't considered to train the proposed model as well as the individual models. The feature selection effects the model performance which is shown in Table 3.6.

3.2.4 Machine Learning Methods

To get the better results, parameters of models are need to be tuned. The models used in this Chapter is described in Table 3.5 with required R packages and their tuning parameters.

3.2.5 Benchmark of the Proposed Model Correctness

For the benchmarking of model correctness, the performance of proposed model has been compared with existing IgPred [30] model. The proposed model is based on seven different models and IgPred is based on single model *i.e.* SVM. The independent dataset of 44 IgG, 44 IgA and 44 non-IgG, 44 non-IgA epitopes have been collected from the immune epitope database (IEDB). The epitopes in benchmark dataset are unique and they are not present in the training of IgPred and proposed model.

IgPred prediction on the benchmark dataset has been recorded by using its web server (*http://crdd.osdd.net/raghava/IgPred/*). Benchmark dataset of epitopes are provided in Table 3.14 with predictions from IgPred and the proposed model. The predictions from IgPred and the proposed

Table 3.4: Feature importance for IgG and IgA epitopes.

Rank	**Features**	**IncNodePurity**$_{IgG}$	**Features**	**IncNodePurity**$_{IgA}$
1	F_{ab}	177.33	F_{d}	11.19
2	F_{e}	160.75	F_{aa}	7.79
3	F_{c}	155.96	F_{ab}	7.49
4	F_{d}	153.55	F_{e}	6.74
5	SL	143.7	F_{c}	6.41
6	F_{z}	133.5	F_{q}	6.11
7	F_{b}	133.33	SL	5.94
8	F_{aa}	132.63	F_{v}	5.81
9	F_{y}	117.86	F_{p}	5.53
10	F_{f}	112.47	F_{y}	5.31
11	F_{a}	100.2	F_{b}	5.00
12	F_{ac}	97.5	F_{z}	4.65
13	F_{q}	71.98	F_{a}	4.60
14	F_{p}	68.15	F_{h}	4.47
15	F_{r}	63.06	F_{f}	4.11
16	F_{h}	56.14	F_{ac}	4.03
17	F_{v}	52.89	F_{x}	3.97
18	F_{s}	51.95	F_{t}	3.87
19	F_{u}	50.66	F_{w}	3.80
20	F_{t}	50.58	F_{r}	3.56
21	F_{k}	48.63	F_{s}	3.46
22	F_{w}	44.13	F_{u}	3.43
23	F_{g}	44.11	F_{g}	3.12
24	F_{x}	41.17	F_{l}	2.79
25	F_{i}	36.19	F_{k}	2.62
26	F_{l}	34.13	F_{m}	2.42
27	F_{m}	29.67	F_{i}	1.89
28	F_{j}	27.8	F_{j}	1.44
29	F_{n}	21.61	F_{o}	1.34
30	F_{o}	21	F_{n}	1.22

model on these epitopes are used to compare them through various parameters including Gini, AUC, accuracy, MCC, specificity and sensitivity as described in Table 3.13. The results shows that the proposed model is outperforming the existing model.

3.3 Methodology

The methodology has been represented in Figure 3.1. Initially, peptide sequences are collected from *http://crdd.osdd.net/raghava/IgPred/*. Dataset contains peptide sequences which has negative and positive epitopes inducing IgA and IgG antibodies with variable length from 4 to 50-mers. The feature measurement has been performed in the second step and explained in Section 3.2.2. In the

Table 3.5: Machine learning models considered for ensembling; their respective R packages, methods and tuning parameters.

Model	Method	Required Package	Tuning Parameter
RF [87]	rf	random forest	mtry=2, ntree=500
SVM [88]	ksvm	kernlab	kernel=”rbfdot”, type=”C-svc”
Decision Tree [89]	rpart	None	usesurrogate=0, maxsurrogate=0
NN [90]	nnet	nnet	size=10
ELM [91]	elmtrain	elmNN	nhid=10
Avnnet [92]	avNNet	caret	size, linout, trace
RRF [86]	RRF	RRF	None

third step, RRF [86] is used to get the subset of important features. This process reduces the space complexity, time complexity as well as increases the accuracy of model. In the fourth step, the dataset has been used to train the models, with their optimum tuning parameters. The used machine learning models are presented in Table 3.5. The models have been combined to get the proposed multilevel ensemble model as shown in Figure 3.3 and also explained in Section 3.3.2. Finally, performance of the proposed model has been evaluated on various parameters such as AUC, specificity, sensitivity, Gini and accuracy. To check the consistency of the proposed model’s prediction, repeated k-fold cross validation has been performed. The flow of proposed scheme is presented in Figure 3.2.

3.3.1 Flow of Proposed Scheme

The Figure 3.2 describes the prediction of IgG, IgA epitopes and their physicochemical properties (as mentioned in Section 3.2) are used to train the machine learning models. Seven different models have been combined to get the proposed model and is explained in Section 3.3.2. The final prediction is produced by majority voting from the seven models which are used in the proposed ensemble model. The proposed model is separately trained for IgA and IgG inducing epitopes.

3.3.2 Proposed Multilevel Ensemble Model

Ensemble is used to deal with the worst case of model prediction. In this Chapter, the focus is on the refinement of false prediction as well as true prediction of the model. Seven models which include Decision tree, RF, SVM, ELM, NN, RRF and avNNet are combined to get better accuracy as mentioned in Figure 3.3. All the models has been trained on 70% of the dataset and 30% has been used to test them. The proposed model is divided into three phases and all the phases are explained below:

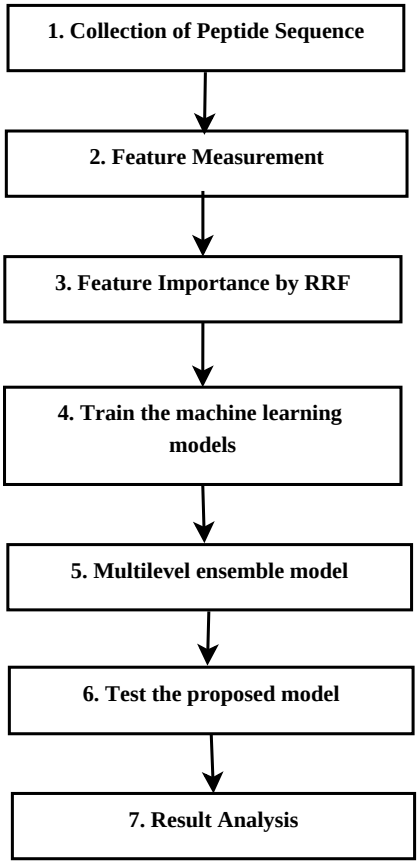

Figure 3.1: Methodology of the proposed model.

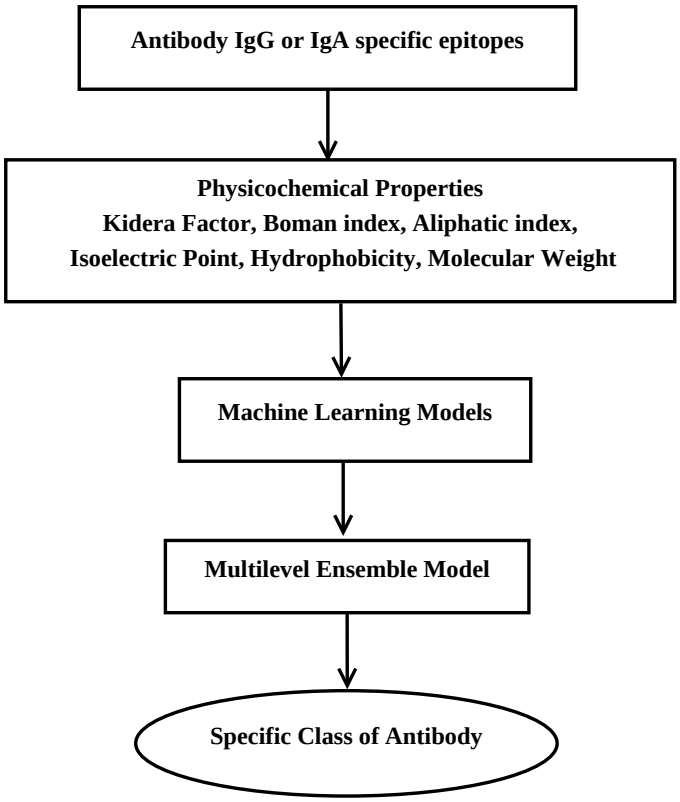

Figure 3.2: Flow of the proposed scheme.

Phase I : The decision tree, ELM, NN, SVM models have been trained with 70% of dataset and generate predictions on the rest 30% of dataset.

Phase II : The false predictions of two models (decision tree and ELM) from Phase I are used to train the RF model. The false predictions of two models (NN and SVM) from Phase I have been used to train the avNNet model.

Phase III : The false predictions from Phase II and true predictions from Phase I have been combined. This combined new dataset has been used to train the RRF model which provides the final predictions.

In this approach, true predictions as well as false predictions are refined to get accurate proposed model. These true and false predictions are attained by testing the models with the same data *i.e.* training dataset. The purpose of using true prediction as the input of other models is to deal with false positive results (Non-antigenic epitope is considered as antigenic epitope). The data is travelled through seven models because of this models perfectly learn the data to provide reliable and accurate results. At the end, test dataset has been used to test the ensemble model. The final predictions have been attained by considering votes from all the models which are in the ensemble model.

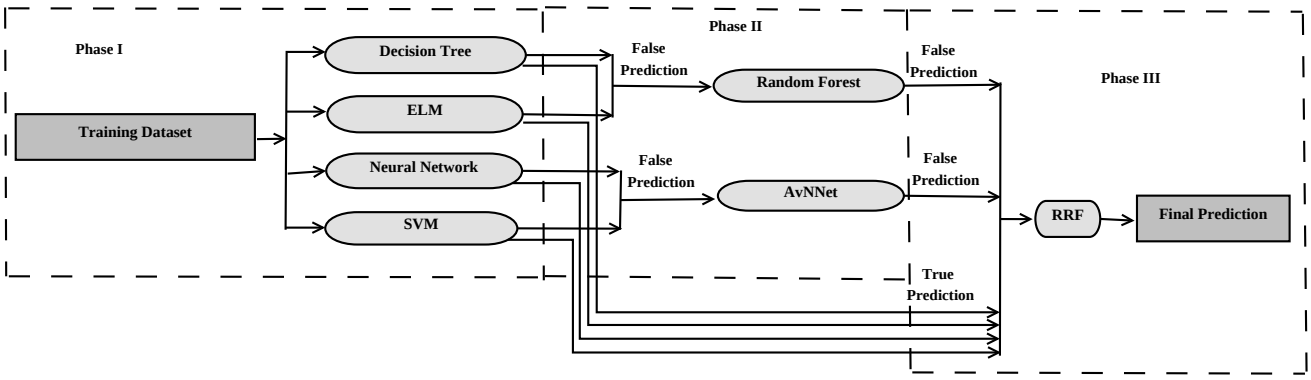

Figure 3.3: Multilevel ensemble model.

3.4 Model Evaluation

Various parameters such as Gini, accuracy, AUC, specificity and sensitivity have been calculated to evaluate the performance of proposed model and individual models which are explained in Chapter 1. Repeated k-fold cross validation has been performed to test the robustness of proposed model. According to RRF algorithm, seven features are discarded from both the datasets (IgA and IgG) as

mentioned in Section 3.2.3. Based on rest of the features, formula for IgG antibody prediction has been formulated which consists of important features and target class as mentioned below:

$$\mathrm{CL}_{IgG} \sim f(F_a, F_b, F_c, F_d, F_e, F_f, F_g, F_h, F_k, F_p, F_q, F_r, F_s, F_t, F_u, F_v, F_w, F_y, F_z, F_{aa}, F_{ab}, F_{ac}) \quad (3.1)$$

For IgA antibody prediction, formula is given below:

$$\mathrm{CL}_{IgA} \sim f(F_a, F_b, F_c, F_d, F_e, F_f, F_g, F_h, F_p, F_q, F_r, F_s, F_t, F_u, F_v, F_w, F_x, F_y, F_z, F_{aa}, F_{ab}, F_{ac}) \quad (3.2)$$

3.4.1 Repeated K-Fold Cross Validation

The large number of comparisons are always preferred, to comparison the performance of model. While performing cross validation, random data is provided in each fold to do the comparisons. In k-fold cross validation, only k comparisons are acquired. Thus, to run k-fold cross validation multiple time or increase the number of comparisons, repeated k-fold cross validation is an efficient way of cross validation. Here, 10-fold cross validation has been repeated for 3 times.

3.5 Result Analysis, Comparison and Discussion

Epitope length is important while training the model and it is efficient to use variable length of epitopes. The machine learning models have been trained with fixed length of IgG, IgA epitopes and evaluated on various parameters as mentioned in Table 3.7 and Table 3.8. On the other hand, models trained with variable length of epitopes have been evaluated via evaluation parameters. Results describe that these models are performing better as compared to models trained with fixed length of epitopes as shown in Table 3.9 and Table 3.10.

The models may get biased while training, to handle this issue SMOTE algorithm can be used. Another problem in trained models is overfitting/underfitting, to deal with such issues, the model should be cross validated and tested on benchmark dataset, if performance is found to be consistent then models are not affected from such issues. A model is overfitted when it learns too much. In contrast, a model is underfitted when it learned too less. In cross validation, models are executed n times and accuracy is recorded if accuracy is highly fluctuating then that model

is overfitted/underfitted/biased. Repeated k-fold cross validation has been used to describe the consistency in the accuracy which also concludes that the proposed model is not affected from such problems. For validation of the proposed model, benchmark dataset is used and compared with the existing model by using various parameters such as Gini, AUC, accuracy, MCC, specificity and sensitivity as described in Table 3.13. The result concludes two things about the proposed model. First, the proposed model is free from overfitted/underfitted/biased issues. Second, the outcome of proposed model has been improved as compared to the existing technique.

Table 3.6 depicts the accuracy on various subset of features. The seven least ranked features are discarded because they are affecting the accuracy of the models. To balance the number of features and accuracy, 23 feature (F_{ab} - F_g in IgG antibody and F_d - F_g in IgA antibody) have been considered to train the models.

The proposed and individual models have been evaluated on various parameters as mentioned in Table 3.9 and Table 3.10. From the results, it is concluded that the accuracy of proposed model is increased as compared to the single model accuracy.

Table 3.11 describes the accuracy of the proposed model while performing cross validation. The accuracy has been recorded by applying 10-fold cross validation 3 times. Here, 70% of dataset is used for training and 30% is used for testing. The Figure 3.4 and Figure 3.5 describe the accuracy of proposed model 3 times in 10 runs and shows the consistency in the accuracy of proposed model.

From the results and comparison, it is concluded that predictability of the proposed ensemble model has been improved significantly as compared to the individual models and existing model.

3.5.1 Performance Comparison on Benchmark Dataset

The dataset has been extracted from IgPred [30]. The IgA, IgG incuding epitopes and non-IgA, non-IgG epitopes of length 4 to 50-mers are available on *http://crdd.osdd.net/raghava/IgPred/* which have been used to train the proposed model. This is a balanced dataset which means equal number of both the target class. IgPred uses features like binary profiles, dipeptide composition and amino acid composition. The outcome of dipeptide composition based model is better than other models in IgPred. In this Chapter, results of dipeptide composition based model has been compared with the proposed model. The results recommend that the proposed model outperforms the existing server. The Table 3.12 describes the performance of IgPred and the proposed model (IgG, IgA).

Table 3.6: Impact of features on accuracy for IgG and IgA epitopes.

Number of features	**Features**	**Accuracy$_{IgG}$**	**Features**	**Accuracy$_{IgA}$**
10	F_{ab} - F_f	96.6	F_d - F_y	98.5
11	F_{ab} - F_a	96.6	F_d - F_b	98.4
12	F_{ab} - F_{ac}	96.1	F_d - F_z	98.2
13	F_{ab} - F_q	95.5	F_d - F_a	98.5
14	F_{ab} - F_p	95.6	F_d - F_h	97.9
15	F_{ab} - F_r	95.0	F_d - F_f	97.2
16	F_{ab} - F_h	94.9	F_d - F_{ac}	97.5
17	F_{ab} - F_v	94.6	F_d - F_x	97.5
18	F_{ab} - F_s	93.0	F_d - F_t	97.5
19	F_{ab} - F_u	88.1	F_d - F_w	97.5
20	F_{ab} - F_t	90.2	F_d - F_r	97.5
21	F_{ab} - F_k	91.8	F_d - F_s	97.5
22	F_{ab} - F_w	94.4	F_d - F_u	97.5
23	**F_{ab} - F_g**	**94.4**	**F_d - F_g**	**97.5**
24	F_{ab} - F_x	93.5	F_d - F_l	96.2
25	F_{ab} - F_i	92.4	F_d - F_k	95.2
26	F_{ab} - F_l	84.2	F_d - F_m	95.4
27	F_{ab} - F_m	72.3	F_d - F_i	94.9
28	F_{ab} - F_j	77.4	F_d - F_j	94.6
29	F_{ab} - F_n	73.0	F_d - F_o	93.8
30	F_{ab} - F_o	70.2	F_d - F_n	93.2

Table 3.7: Performance evaluation of machine learning models for fixed length of IgG epitopes.

SN	**Model Name**	**Gini**	**ACC**	**AUC**	**Spec**	**Sens**
1	RF	0.16	58.45	0.58	0.51	0.51
2	AvNNet	0.06	51.47	0.53	0.50	0.45
3	Decision Tree	0.16	49.97	0.42	0.40	0.46
4	RRF	0.19	59.84	0.59	0.52	0.52
5	NN	0.08	54.20	0.54	0.48	0.48
6	ELM	0.02	49.43	0.51	0.45	0.46
7	SVM	0.08	54.25	0.54	0.48	0.47
8	**Proposed model**	**0.19**	**67.02**	**0.59**	**0.53**	**0.50**

To validate, the existing model and proposed model have been evaluated on various parameters such as Gini, AUC, accuracy, MCC, specificity and sensitivity. The benchmark dataset is also used as mentioned in Section 3.2.5. These epitopes prediction has been calculated from IgPred server and compared with outcome of the proposed model as shown in Table 3.13. The result shows that the proposed model outperforms the existing model.

Table 3.8: Performance evaluation of machine learning models for fixed length of IgA epitopes.

SN	Model Name	Gini	ACC	AUC	Spec	Sens
1	RF	0.32	63.35	0.66	0.68	0.43
2	AvNNet	0.34	44.1	0.67	0.71	0.40
3	Decision Tree	0.23	60.87	0.61	0.61	0.45
4	RRF	0.30	62.73	0.65	0.66	0.44
5	NN	0.14	57.14	0.57	0.56	0.44
6	ELM	0.05	44.1	0.52	0.44	0.50
7	SVM	0.33	61.49	0.66	0.70	0.41
8	**Proposed model**	**0.34**	**61.40**	**0.67**	**0.57**	**0.41**

Table 3.9: Performance evaluation of machine learning models for variable length of IgG epitopes.

SN	Model Name	Gini	ACC	AUC	Spec	Sens
1	RF	0.31	65.3	0.65	0.56	0.56
2	AvNNet	0.21	60.6	0.61	0.52	0.54
3	Decision Tree	0.23	61.1	0.62	0.51	0.56
4	RRF	0.31	65.8	0.66	0.56	0.56
5	NN	0.21	60.6	0.61	0.53	0.53
6	ELM	0.19	50.8	0.59	0.43	0.61
7	SVM	0.28	63.9	0.64	0.54	0.56
8	**Proposed model**	**0.86**	**94.43**	**0.93**	**0.99**	**0.98**

Table 3.10: Performance evaluation of machine learning models for variable length of IgA epitopes.

SN	Model Name	Gini	ACC	AUC	Spec	Sens
1	RF	0.38	69.11	0.69	0.59	0.57
2	AvNNet	0.31	65.85	0.66	0.59	0.53
3	Decision Tree	0.28	64.23	0.64	0.57	0.53
4	RRF	0.39	69.51	0.69	0.59	0.58
5	NN	0.34	67.07	0.67	0.58	0.55
6	ELM	0.19	58.94	0.59	0.51	0.53
7	SVM	0.36	68.29	0.68	0.59	0.55
8	**Proposed model**	**0.91**	**97.56**	**0.95**	**0.97**	**0.99**

Table 3.11: Repeated 10-fold cross validation of IgG and IgA proposed model.

	IgG Proposed Model			IgA Proposed Model		
Folds	**Run 1**	**Run 2**	**Run 3**	**Run 1**	**Run 2**	**Run 3**
1	93.10	92.41	93.36	93.17	95.61	96.10
2	93.07	93.48	93.22	94.15	96.10	96.10
3	93.56	93.02	92.99	92.20	97.07	95.12
4	92.93	93.62	93.28	95.12	93.66	96.10
5	92.41	93.36	93.19	94.15	95.12	94.63
6	92.41	92.24	93.13	93.66	96.10	97.07
7	93.82	93.05	93.53	92.68	97.56	92.20
8	93.79	92.76	92.39	94.63	90.73	94.63
9	93.91	93.45	93.45	95.12	97.07	93.66
10	93.25	93.05	93.51	92.68	93.66	94.63

Table 3.12: Performance comparison with existing model and the proposed model.

	IgG Epitopes		IgA Epitopes	
Parameters	**IgPred**	**Proposed Model**	**IgPred**	**Proposed Model**
ACC(%)	70.42	**94.43**	72.07	**97.56**
MCC	0.41	**0.86**	0.44	**0.89**
AUC	0.76	**0.93**	0.78	**0.95**

Table 3.13: Performance comparison on benchmark dataset with existing model and the proposed model.

	IgG Epitopes		IgA Epitopes	
Parameters	**IgPred**	**Proposed Model**	**IgPred**	**Proposed Model**
ACC(%)	73.86	86.36	54.55	77.27
AUC	0.77	0.86	0.57	0.77
Gini	0.54	0.72	0.14	0.54
Spec	0.86	0.90	0.62	0.74
Sens	0.67	0.83	0.52	0.81
MCC	0.50	0.73	0.11	0.55

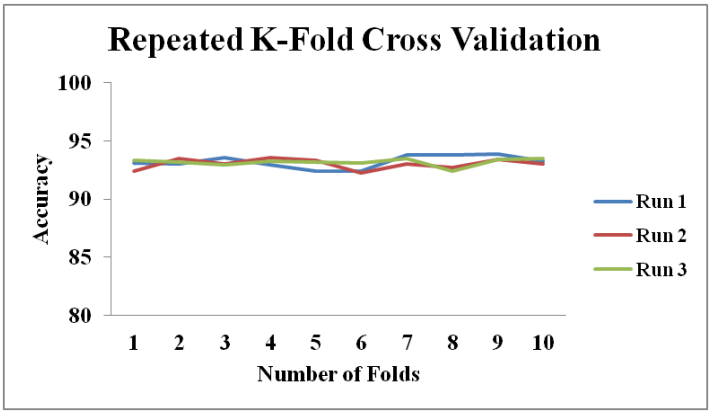

Figure 3.4: Repeated k-fold cross validation of the IgG proposed model.

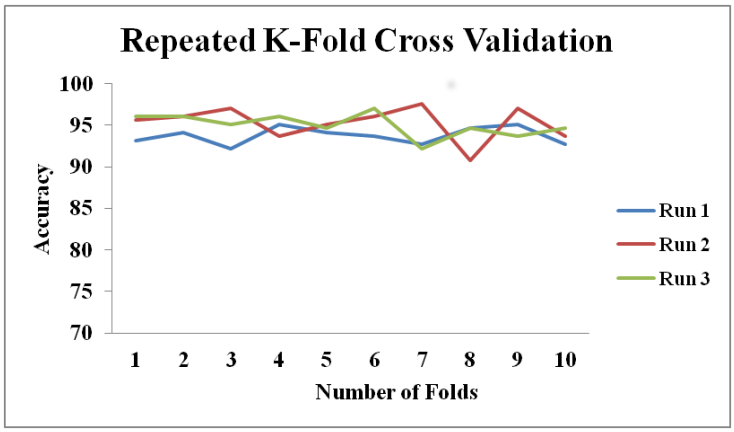

Figure 3.5: Repeated k-fold cross validation of the IgA proposed model.

3.6 Conclusion

The proposed model increases the prediction accuracy of IgG and IgA antibodies as compared to the existing technique. In this Chapter, seven different models including decision tree, ELM, RF, NN, SVM, avnnet and RRF have been used to create multilevel ensemble model. A novel multilevel ensemble model has been developed for prediction and it produces high accuracy, Gini, AUC, specificity and sensitivity with variable length of epitopes. The multilevel ensemble model is divided into three phases. In this approach, true predictions and false predictions are refined to get accurate proposed model. The benefit of using true prediction as the input of other models is to deal with false positive results. The data has been travelled through seven models in such a way that each model is able to learn the data perfectly. This technique provides reliable and accurate results. The proposed model has been compared with existing IgPred model and validated on benchmark dataset. To check the robustness of proposed model, repeated k-fold cross validation has been performed.

Table 3.14: Benchmark dataset of IgG and IgA epitopes.

SN	IgG_Sequence	Actual	Proposed Model	IgPred Model	IgA_Sequence	Actual	Proposed Model	IgPred Model
1	AATGAATAAA	1	1	0	AEVLKDAIKDLVMTKPAPTC	0	0	1
2	AEFYLNPDQSGEFMFDFDGDEIF	1	1	1	AGKREMVIIT	1	1	1
3	AFYGVWPLL	0	1	0	AHGRSQVLQQSTYQLLQELCC	0	0	1
4	AILDMIAGAHWGVLAGIA	0	0	0	AQGSVQPQQLPQFEEIRNLAL	0	0	1
5	AKAPPAPKPAPQPGP	1	1	1	ASLKMADPNRFR	1	1	1
6	AKQELE	1	1	1	ASREAK	1	1	1
7	ATRKTSER	0	0	0	ASREAKKQVEKALE	1	1	0
8	AYALYGVWPL	0	0	0	DAEFRHDSGYEVHHQKLVFFAED VGSNKGAIIGLMVGGVVIA	1	0	0
9	CFHQGKEYAPG	1	1	1	DEKGIMRTGLISFENNNYYFNENGE	1	1	1
10	CITQYERESQAYY	1	1	1	EQSRCQAIHN	1	1	1
11	CNRLGGLFNFGPKQKI	1	1	1	ERMKDTLRIT	1	1	1
12	DLEDRDRSELSPLLLTTT	0	0	1	ESMAGKREMV	1	0	1
13	DLLVGSATLCSALYVGDL	0	0	0	FFQP	0	1	0
14	DPNAVCETDKWKYENPCKKM	1	1	1	FPPQLPYPQP	0	0	1
15	DVKFPGGG	0	0	1	FTTKVIGKDSRDFDISPKV	1	1	1
16	EGHLIDLKRV	1	1	1	GGSGGRGRGGSGGRGRGGSG	0	0	1
17	EKPHFP	1	1	1	GIAKAFRAPSIREVSPG FGTLTQGGASIMYGN	1	0	0
18	EPWFLHGLGLARTYWRDTNTG	1	1	1	GKREMVIITF	1	0	1
19	ESDEAPFMFSENKFL	1	0	1	GSENKRTGALGNLKN	1	1	1
20	EVNQIVETNVRLRQQW	1	1	1	GSPPRRPPPGRRPFFHPVGE	0	0	0
21	FKNPHAKKQDVV	1	1	1	IDKLCVWNNK	1	1	1
22	FSPRRHWTTQGCNCSIYP	0	0	0	IILHQQHH	0	1	0
23	GCNCSIYPGHITGHRMAW	0	1	0	IPEQ	0	0	0
24	GGQIVGGV	0	0	1	ISIKYDPRKDSEVFA	1	1	1
25	GIGAVLKVLTTGLPALISWIKR	1	1	1	ITFKSGATFQ	1	0	1
26	GNASRCWVAMTPTVATRD	0	0	0	ITIMDNGNIDTELLVGTLTLGGY	1	1	1
27	GSAGHTVSGFVSLLAPGA	0	1	1	KASITEIKADKT	1	1	1
28	GVDAETHVTGGSAGHTVS	0	0	1	KGINLIDDIKYYFDEKGIMRTGLIS	1	0	1
29	GVRATRKT	0	0	1	KREMVIITFK	1	0	1

to be cont'd on next page

Table 3.14: Benchmark dataset of IgG and IgA epitopes. (cont.)

SN	IgG_Sequence	Actual	Proposed Model	IgPred Model	IgA_Sequence	Actual	Proposed Model	IgPred Model
30	GYGAGVAGAL	0	0	0	LALLAIVATTATTAVRVPVPQ	0	1	0
31	HADPAPASAENVKEIIELLKGL DLRLQTVEGKVDKILA	1	1	1	LCCQHLWQIPEQSQCQAIHNV	0	0	1
32	HYAPRPCGI	0	0	0	LCVWNNKTPN	1	1	1
33	HYKLFLARL	0	0	0	LQQQLIPCRD	1	1	1
34	KNQVEGEVQIVSTATQTFLA	0	0	0	LRITYLTETK	1	0	1
35	KSGNFKHLREFVFK NKDGFLYVYKGYQPIDV	1	1	1	LRLQTAGNVDHVGLGT	1	1	1
36	KTNTPADVFIVFTDNETFAG	1	1	1	MKTFLILALLAIVATTATTAV	0	0	0
37	KTSERSQP	0	0	1	MSDGAVQPDGGQPAVRNERATG	1	1	0
38	LEGAARQ	1	1	1	MVIITFKSGA	1	1	1
39	LGVRATRK	0	0	1	PGEGPSTGPRGQGDGGRRKK	0	1	1
40	LPATQLRRHIDLLVGSAT	0	0	0	PLKP	1	1	1
41	MMNWSPTTALVMAQLLRI	0	1	1	PLLGCIGSTCAEDGN	1	1	1
42	MYVGGVEHRL	0	0	1	PPDQLVNLHDFRSD EIEHLVVEE	1	1	0
43	NPDPNPN	1	1	1	PQQPISQQQQQQQQQQQQQ	0	0	1
44	NQVYYRPMDEYSN	1	1	1	QEQKQQLQQQ	0	0	0
45	NRRPQDVK	0	0	1	QFLGQQQPFPPQQPYPQPQPF	0	0	1
46	NSTNSGIN	1	1	1	QGSFRPSQQNPQAQGSVQPQQ	0	0	1
47	NTHVTGAVQGHGAF TLTSLFQPGASQKIQLV	1	1	1	QLVKDKNIDISIKYDPRKDSE	1	0	1
48	PGGGQIVG	0	0	0	QPFPPQLPYP	0	0	1
49	PIPKARRP	0	0	0	QPFPQPQLPYSQPQPFRPQQP	0	0	1
50	PLATQPPVLAL	1	1	1	QPFPSQQPYLQLQPFPQPQLP	0	0	1
51	PPAYEK	1	1	1	QPQEQVPLVQQQQFLGQQQPF	0	0	1
52	PPGEFLQVSIQDTRNAVRAC	1	1	1	QPQYSQPQQPISQQQQQQQQ	0	0	1
53	PQDVKFPG	0	0	1	QPYLQLQPFPQPQLPYSQPQP	0	0	1
54	PRRGPRLGVRAPRKTS	0	1	1	QPYPQPQPFPSQQPYLQLQPF	0	0	1
55	QDVKFPGG	0	0	1	QQLQQQQQQQ	0	0	1

to be cont'd on next page

Table 3.14: Benchmark dataset of IgG and IgA epitopes. (cont.)

SN	IgG_Sequence	Actual	Proposed Model	IgPred Model	IgA_Sequence	Actual	Proposed Model	IgPred Model
56	QDVKFPGGGQIVGG VYLLPRRGPRL	0	0	1	QQQQLQQQQQ	0	0	1
57	QKTRTSRRAKPPQRPKQQPAAP	1	1	1	QQQQQQQQQQQQQILQQILQQ	0	0	1
58	QPIPKARR	0	0	0	QQQQQQQQQQQQQQQQQQQIL	0	0	1
59	RATRKTSE	0	0	0	QQYPLGQGSFRPSQQNPQAQG	0	0	1
60	REQAPNLVY	1	1	1	REMVIITFKS	1	0	1
61	RGQRTKTNARTRKGPRKPIKK	1	1	1	RGRGRGEKRPRSPSSQSSSS	0	1	0
62	RKTSERSQ	0	0	1	RRPFFHPVGEADYFEYHQEG	0	0	0
63	RLGVRATR	0	0	1	SATAIFDTTTLNPTIAGAGDVKASAE GQLG	1	1	1
64	RPEPKKPWSGVWNASTY	1	1	1	SEEEND	1	1	1
65	RSQPRGRR	0	0	0	SELLSLINDMPITNDQKKLMSNNV	1	0	1
66	SAFVFPTKD	1	0	1	SFQQPLQQYPLGQGSFRPSQQ	0	0	1
67	SAQSGTSGTSAQSGT	1	0	1	SGPRHRDGVRRPQKRPSCIG	0	1	1
68	SCLTVPASAYQVRNSTGL	0	1	0	SLLTEVETPIRNEWGCRCNDSSD	1	1	0
69	SMVGNWAKVLVVLLLFAG	0	1	0	SMAGKREMVI	1	0	1
70	SQWN	1	1	1	SQQNPQAQGSVQPQQLPQFEE	0	0	1
71	SSKYGDTSTNNVRG DLQVLAQKAERTLP	1	1	1	SQVLQESTYQ	0	0	1
72	TCSMFVYGGC	1	1	1	STYQLVQQLC	0	1	1
73	TISSLQS	1	1	0	TAVRVPVPQLQPQNPSQQQPQ	0	0	1
74	TPGCVPCVREGNASRCWV	0	1	0	TETKIDKLCV	1	1	1
75	TRKTSERS	0	0	0	TFKSGATFQV	1	1	1
76	TSGTSGTSGTSPSSR	1	0	1	TINKPKGYVGKE	1	0	1
77	TWGENETDVLLLNNTRPPQ	1	1	1	TLRITYLTET	1	1	1
78	VDPLPSGYQFNPEATKAASP	1	1	1	TRLSRTIGYTVK	1	1	1
79	VENGLISRVLDGLV	1	1	0	TSQDGNNHQFT	1	1	1
80	VFVGLILLTL	0	0	0	VATTATTAVRVPVPQLQPQNP	0	0	1
81	VKEFLESSPNTQWELRAFMA	1	1	1	VETEDTKEPGVLMG GQSESVEFTKDTQTGM	1	1	1
82	VKTIGDKRTLTLNTTANYT	1	1	0	VKAETRLNPDLQPTE	1	1	1

to be cont'd on next page

Table 3.14: Benchmark dataset of IgG and IgA epitopes. (cont.)

SN	IgG_Sequence	Actual	Proposed Model	IgPred Model	IgA_Sequence	Actual	Proposed Model	IgPred Model
83	VRATRKTS	0	0	1	VLQQSTYQLLQELCCQHLWQI	0	0	1
84	VTSVSAVASGHYLHR	1	1	1	VVLQQHNIAH	0	0	1
85	VYEAADAILHTPGCVPCV	0	0	1	WQIPEQSQCQAIHNVVHAIIL	0	0	1
86	WGVLAGIAYFSMVGNWAK	0	0	0	YLLPRRGPRL	0	0	0
87	WHLNSTALNCNDSLNTGW	0	0	0	YPQPQPQYSQ	0	0	1
88	YWPPPQGRRRF	1	1	1	YQLLQELCCQHLWQIPEQSQC	0	0	1

1 represents IgG or IgA epitopes, 0 represents Non-IgG or Non-IgA epitopes

Chapter 4

Ensemble Technique for Prediction of T-cell Mycobacterium Tuberculosis Epitopes

Development of an effective machine learning model for T-cell mycobacterium tuberculosis (M. tuberculosis) epitopes is beneficial for saving biologist's time and effort for identifying epitope in a targeted antigen. Existing NetMHC 2.2, NetMHC 2.3, NetMHC 3.0 and NetMHC 4.0 etc estimate binding capacity of peptide. This is still a challenge for those servers to predict whether a given peptide is M.tuberculosis epitope or non-epitope. One of the servers, CTLpred works in this category but it is limited to peptide length of 9-mers. Therefore, in this Chapter direct method of predicting M. tuberculosis epitope or non-epitope has been proposed which also overcomes the limitations of above servers. The proposed method is able to work with variable length epitopes having size even greater than 9-mers. Identification of T-cell or B-cell epitopes in the targeted antigen is the main goal in designing epitopes based vaccine, immune-diagnostic tests and antibody production. Therefore, it is important to introduce a reliable system which may help in the diagnosis of M. tuberculosis. In this Chapter, computational intelligence methods are used to classify T-cell M. tuberculosis epitopes. The caret feature selection approach is used to find out the set of relevant features. The ensemble model is designed by combining three models and is used to predict M. tuberculosis epitopes of variable length (7 to 40 mers). The proposed ensemble model achieves 82.0% accuracy, 0.89 specificity, 0.77 sensitivity with repeated k-fold cross validation having average accuracy of 80.61%. The proposed ensemble model has been validated and compared with NetMHC 2.3, NetMHC 4.0 servers and CTLpred T-cell prediction server. [†]

[†]D Khanna and PS Rana, Ensemble Technique for Prediction of T cell Mycobacterium Tuberculosis Epitopes, Interdisciplinary Sciences: Computational Life Sciences, Springer 11 (4) (2019) 611-627.

4.1 Introduction

Tuberculosis (TB) is a destructive global health communicable disease which is caused by mycobacterium tuberculosis. According to the survey of WHO in 2015 [93], approximately 10.4 million TB cases worldwide were found in which 5.9 million men, 3.5 million female and 1.0 million children. The patients who were suffering from HIV (1.2 million) were also affected by TB. Approximate death rate of TB patients was 1.4 million and 0.4 million for both HIV and TB affected patients. However, the death rate was decreased from 2000 to 2015, even then TB is one of the top 10 death causing disease worldwide. Generally, lungs in the human body get affected from TB disease but it may harm other organs as well. This disease gets transmitted as the antigen released in environment when the infected person coughs or sneezes. Approximately 2-3 billion TB infected patients spread TB disease during their lifespan [93]. This issue has boosted the research field for identifying better vaccines and antibodies for curing tuberculosis. According to [94], Bacillus Calmette-Guerin (BCG) is the only licensed TB vaccine but it is limited in its efficacy and applicability. Thus, a universal TB vaccine was derived by using advanced computational procedures [94].

The pathogenic mycobacterium which causes TB survives within the cells. Thus T-cells are required to fight against bacteria rather than antibodies [67,68]. For correct functioning of the immune system, both humoral and cell mediated immunity work together directly or indirectly depending upon the occurrence of foreign invader. Therefore, there are numerous methods for predicting T-cell and B-cell epitopes which help in designing of epitope based vaccines, immune-diagnostic and antibody production.

CTLpred [35] method was trained and tested on T-cell epitopes and non-epitopes. To predict binding capacity of T-cell epitopes with MHC type I and M.tuberculosis, many servers exist including, SVM based model [69], NN based model [70], improved NN model [71], SVMHC [72], NetMHC 2.2 [31] server, NetMHC 3.0 [33] server, NN-Align method [76], NetMHC 4.0 [34] server and NetMHCpan 3.0 [77] server. To predict MHC II peptide binding capacity [73], the existing servers include SVM based model [74], linear B-cell epitopes [75], NetMHC 2.2 [31] server and NetMHC 2.3 [32] server.

Machine learning models can be used to reduce the number of wet lab experiments and suggest

some focused experiments out of them. They are in trend of recognising T-cell epitopes, B-cell epitopes and many more areas of biology. Instead of predicting binding capacity of an epitope to determine T-cell M. tuberculosis, there is an another way which is called as classification of M. tuberculosis epitopes. In this Chapter, targeted epitopes contain both MHC class I and II T-cell epitopes.

Inspired from the contribution of machine learning techniques in biology and increasing spread rate of M. tuberculosis disease, there is need to find an accurate model which can classify T-cell epitopes of M. tuberculosis and non-epitopes. A hybrid model has been designed in which physicochemical properties of amino acids are used to train the machine learning models. These models have been fused to design ensemble model which will classify T-cell M. tuberculosis epitope or non-epitope.

A brief overview of the Chapter is as follows: the dataset, feature extraction, machine learning model, aim of the proposed study are detailed in Section 4.2. The proposed methodology, feature selection approach, data partitioning approach, proposed ensemble model and analysis of improved results are explained in Section 4.3. Model evaluation on various parameters, repeated k-fold cross validation, benchmark dataset and comparison with existing systems are detailed in Section 4.4. Evaluation of Results has been narrated in Section 4.5. Discussion has been detailed in Section 4.6. Conclusion and future work is described in Section 4.7.

4.2 Materials and Methods

The section contains the details of dataset, alleles, feature extraction, machine learning models and aim of the proposed study.

4.2.1 Dataset and its Features

The T-cell epitopes of M. tuberculosis and non-epitopes are collected from IEDB. Total 4045 epitopes of T-cell M. tuberculosis are extracted, from which 1804 epitopes are left after removing similar epitopes and 1804 are non-epitopes. Non-epitopes vary from 8 to 20 mers and M. tuberculosis epitopes vary from 7 to 40 mers which means epitopes with variable length are considered. Thus, the proposed ensemble model is capable of predicting epitopes of variable length. The glimpse of dataset is presented in Table 4.1.

Table 4.1: Sample dataset of T-cell M. tuberculosis.

Sequence	**SL**[!]	$\mathbf{F}_a$	$\mathbf{F}_b$	———	$\mathbf{F}_{aa}$	$\mathbf{F}_{ab}$	$\mathbf{F}_{ac}$	**CL**[#]
DMWEHAFYL	9	54.44	0.66	———	33.33	11.11	22.22	1
VLMGGVPGVE	10	126.00	-1.54	———	10.00	0.00	10.00	1
STEGNVTGMFA	11	35.45	0.81	———	9.09	0.00	9.09	1
SEFAYGSFVRTVSL	14	76.43	0.92	———	14.29	7.14	7.14	1
TDAATLAQEAGNFER	15	52.67	2.57	———	26.67	6.67	20.00	1
MTEQQWNFAGIEAAAS	16	49.38	1.03	———	12.50	0.00	12.50	1
DAATAQTLQAFLHWAITDGN	20	83.50	0.87	———	15.00	5.00	10.00	1
LLAFTNPTV	9	130.00	-0.77	———	0.00	0.00	0.00	0
GLSTHEGALL	10	127.00	-0.10	———	20.00	10.00	10.00	0
LAQEAGNFERISGDL	15	91.33	1.97	———	26.67	6.67	20.00	0
KTDAATLAQEAGNFE	15	52.67	1.94	———	26.67	6.67	20.00	0
GEAWTGGGSDKALAAATP	18	49.44	0.53	———	16.67	5.56	11.11	0
PAIAAGLNAPRRNRVGRQ	18	81.67	3.08	———	22.22	22.22	0.00	0
MQLVDRVRGAVTGMSRRLVV	20	116.50	2.07	———	25.00	20.00	5.00	0

[!]SL represents sequence length; [#]CL represents class label *i.e.* 1 means M. tuberculosis epitope and 0 means non M. tuberculosis epitope

4.2.2 Alleles used in this Chapter

In this Chapter, epitopes with their corresponding alleles are extracted from IEDB. MHC class I alleles in the present work are : HLA-A * 01:01, HLA-A * 02:01, HLA-A * 02:05, HLA-A * 03:01, HLA-A * 11:01, HLA-A * 24:02, HLA-A * 30:01, HLA-A * 30:02, HLA-A * 68:01, HLA-A * 68:02, HLA-A2, HLA-A24, HLA-B * 07:02, HLA-B * 08:01, HLA-B * 15:01, HLA-B * 15:02, HLA-B * 27:05, HLA-B * 35:01, HLA-B * 35:14, HLA-B * 39:01, HLA-B * 39:05, HLA-B * 40:01, HLA-B * 41:02, HLA-B * 45:01, HLA-B * 56:01, HLA-B * 57:01, HLA-B * 58:01, HLA-B14, HLA-B44, HLA-B52, HLA-C * 06:02, HLA-C * 07:01, HLA-C * 07:02, HLA-C * 12:02.

MHC II class alleles in the present work are : HLA-DP, HLA-DPw4, HLA-DQB1 * 03:02, HLA-DR, HLA-DR1, HLA-DR2, HLA-DR3, HLA-DR4, HLA-DR52, HLA-DR7, HLA-DRB1, HLA-DRB1 * 01:01, HLA-DRB1 * 03:01, HLA-DRB1 * 04:01, HLA-DRB1 * 08:18, HLA-DRB1 * 11:01, HLA-DRB1 * 15:01, HLA-E * 01:03.

4.2.3 Feature Extraction

To enhance the performance and effectiveness of the model, feature extraction is an essential phase. It helps to search out the most informative group of features which can effectively depict the area of interest. The physicochemical properties used here are explained in Chapter 3 and the Table contains brief explanation, related R packages, functions and short names of properties which have been used

in this Chapter. To extract the features, R is used with default parameters of functions. R is an open source software licensed under GNU GPL.

4.2.4 Machine Learning Methods

Machine learning models facilitate to build programs with adaptive nature which are capable to adjust automatically based upon the data they are receiving. Adaptiveness of models allow them to improve their performances without being programmed. In this Chapter, ten models are explored as mentioned in Table 4.2 with required packages and their tuning parameters. To get better results, models may be tuned but here, default parameters have been used.

Table 4.2: Machine learning models considered for ensembling; their respective R packages, methods and tuning parameters.

Model	Method	Required Package	Tuning Parameter
RRF [86]	RRF	RRF	None
Decision Tree [89]	rpart	rpart	parms=list(split="information"), control=rpart.control(usesurrogate=0, maxsurrogate=0)
Avnnet [92]	avNNet	caret	size=10
Blackboost [95]	blackboost	mboost	None
ELM [91]	elmtrain	elmNN	nhid=10, actfun="sig"
GAM [96]	gam	gam	None
Gamboost [95]	gamboost	mboost	None
Glmboost [95]	glmboost	glmboost	None
NN [90]	nnet	nnet	size=10
SVM [97]	ksvm	kernlab	kernel="rbfdot", type="C-svc"

4.2.5 Aim of the Proposed Study

In this section, the aim of the proposed study has been discussed. Existing NetMHC 2.2, NetMHC 2.3, NetMHC 3.0, NetMHC 4.0 etc estimate binding capacity of peptide. This is still a challenge for those servers to predict whether a given peptide is M.tuberculosis epitope or non-epitope. One of the servers, CTLpred works in this category but it is limited to peptide length of 9-mers. Therefore, in this work direct method of predicting M. tuberculosis epitope or non-epitope has been proposed which also overcomes the limitations of above servers. The proposed method is able to work with variable length epitopes having size even greater than 9-mers. It can be seen in Figure 4.1 that when a peptide is given as an input to existing servers and the proposed ensemble model, the predictions made by

them vary in terms of their outcomes. NetMHC 2.2, NetMHC 2.3, NetMHC 3.0 and NetMHC 4.0 servers provide binding capacity of the M. tuberculosis peptide. It cannot be concluded from the output of these servers that the given peptide is an epitope or a non-epitope.

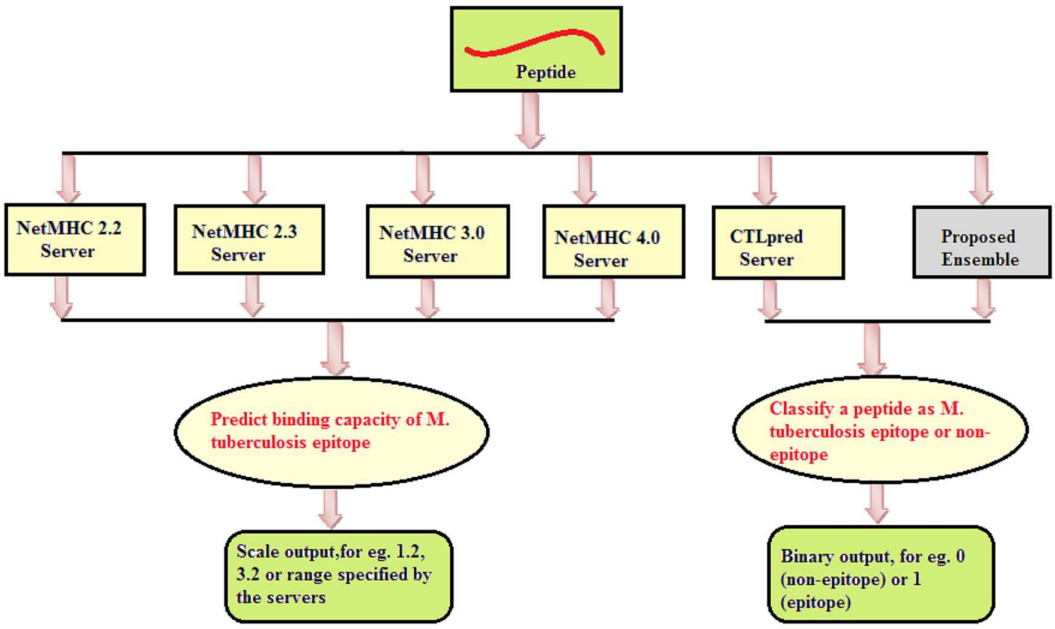

Figure 4.1: Prediction outcomes of existing servers and the proposed ensemble model.

4.3 Methodology

In machine learning, algorithms repetitively learn from data and allow computer to search for the solution of problem without human interference. To perform this task efficiently, an effective methodology is required and has been proposed as described in Figure 4.2. In methodology of this Chapter, following steps are considered:

Step 1 : Initially, 4045 T-cell epitopes of M. tuberculosis (positive) and non-epitopes (negative) are collected from IEDB. In dataset, negative and positive epitopes with variable length have been taken. Length of non-epitopes vary from 8 to 20 mers and M. tuberculosis epitopes vary from 7 to 40 mers.

Step 2 : In this step, feature extraction is done which means that the data from step 1 is preprocessed in order to maximize the information related to the epitopes and non-epitopes. To do this step, R which is an open source software is used as mentioned in Section 4.2.3.

Step 3 : Now, discard redundant epitopes from the dataset.

Step 4 : Then importance of each feature is calculated by using R package caret. This process enhances the accuracy of prediction as explained in Section 4.3.1.

Step 5 : After Step 4, complete dataset ready which is used to train the models. The models used in this Chapter are described in Section 4.2.4 and their required R packages, functions and tuning parameters are mentioned in Table 4.2.

Step 6 : In the this step, three models are fused to get proposed model as discussed in Section 4.3.2.

Step 7 : After execution of the above steps, performance of the proposed ensemble model has been evaluated on various parameters such as specificity, AUC, sensitivity, Gini and accuracy. To analyze the robustness of the model, repeated k-fold cross validation is used. The benchmark dataset has been used to validate and compare the proposed ensemble model with existing systems. Finally, results have been analysed to conclude the effectiveness of the proposed ensemble model.

Final outcome of above stated seven steps is to classify whether an epitope is of M.tuberculosis or not.

4.3.1 Feature Selection

It is required to filter the important features to enhance the performance of models and to reduce computational time and space complexity. In this Chapter, caret a R package is used to perform the feature selection task. Caret package includes the varImp() function to calculate the importance of each features. Here, generalized linear model (glm) has been trained on the complete dataset. To get importance of each feature, trained model is given as an object to the varImp() function. This function provides the weights corresponding to each feature as shown in Table 4.4. All the features are ranked according to these weights. Lower rank shows that the feature is highly important. In table, varImp output ranks F_a to be the most important feature followed by F_r and F_z.

After finding importance of each feature, subset of features are generated as shown in Table 4.3. This table shows that the subsets of 10, 15, 20, 25 and 30 features give accuracy of 81.98%, 80.99%, 80.37%, 82.0% and 78.66% respectively. According to the evaluation parameters include accuracy, specificity and sensitivity, one subset of 25 features (F_a - F_l) is selected. Features F_{ab}, F_t, F_n and F_n are discarded and rest are considered as important ones.

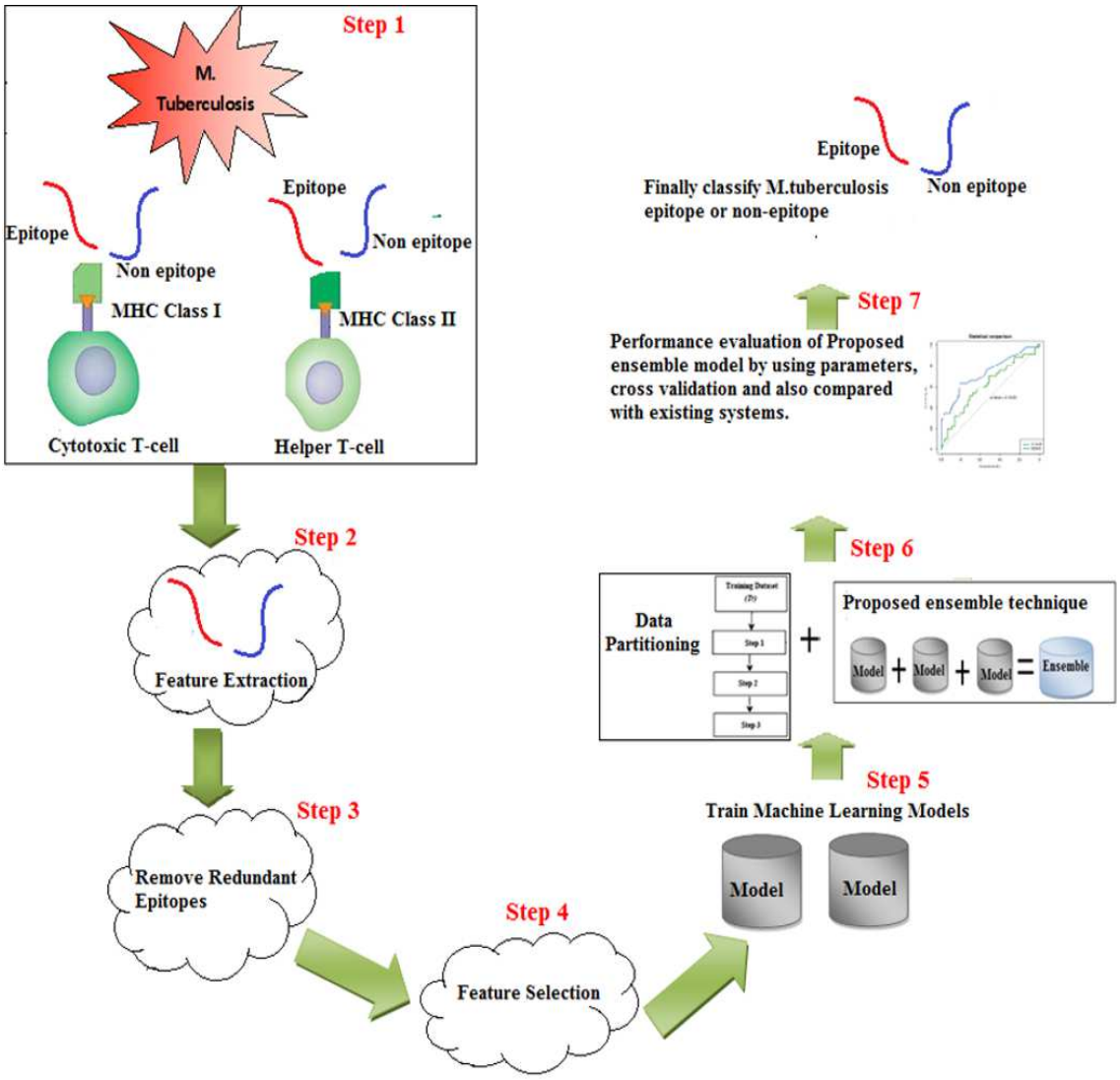

Figure 4.2: Graphical view of the proposed work.

Table 4.3: Features Subsets and their impact on the performance of proposed ensemble model.

Number of features	Features	ACC%	Spec	Sens
10	F_a - F_g	81.98	0.86	0.77
15	F_a - F_u	80.99	0.85	0.76
20	F_a - F_f	80.37	0.88	0.75
25	**F_a - F_l**	**82.0**	**0.89**	**0.77**
30	F_a - F_n	78.66	0.86	0.70

4.3.2 Proposed Ensemble Model

Ensemble modeling is a powerful approach to enhance the performance of a given machine learning model. In this Chapter, the major goal is to enhance a model's performance by rechecking it's false

Table 4.4: Importance of each feature according to the caret package.

Rank	Features	Varimp	Rank	Features	Varimp
1	F_a	3.478	16	F_v	0.654
2	F_r	2.241	17	SL	0.609
3	F_z	2.086	18	F_q	0.567
4	F_b	1.918	19	F_m	0.551
5	F_x	1.794	20	F_f	0.497
6	F_e	1.637	21	F_h	0.307
7	F_c	1.631	22	F_y	0.296
8	F_j	1.533	23	F_p	0.115
9	F_s	0.961	24	F_o	0.070
10	F_g	0.893	25	F_l	0.064
11	F_{aa}	0.712	26	F_{ab}	0.049
12	F_d	0.708	27	F_t	0.047
13	F_i	0.685	28	F_k	0.042
14	F_w	0.655	29	F_n	0.030
15	F_u	0.654			

predictions. While training the models, data splitting is important because if the models get sufficient and effective data then they will be able to predict the unknown data efficiently. Thus, in the present, the focus is on the data splitting as well as fusion of models to improve the outcome. The data division steps and all the phases of proposed ensemble model are explained in the upcoming sections.

4.3.2.1 Data Partitioning to Train the Proposed Ensemble Model

To train the models efficiently, data division process plays an important role. In this Chapter, during ensembling, data has been circulated to each model in such a way that it improves the overall learning process and hence increases models' predictability. Figure 4.3 shows the process of data division. Data is divided in such a way that the complete data is covered efficiently. It is beneficial to train the model with informative set of data to improve its performance. **For testing the ensemble model, 100 instances are extracted from the complete dataset. The training dataset (*Tr*) contains rest 3508 instances which are distributed into three models**. The data partitioning process is described in Algorithm 1.

4.3.2.2 Proposed Ensembling Approach

After data partitioning, effective ensemble technique is required to fuse the models. Here, three models including **decision tree, avNNet and RRF** are combined as mentioned in Figure 4.4 to form the proposed ensemble model. Train the model with informative data in such a that they learn it

Algorithm 1 This algorithm describes the steps of data division:

Step 1 : The training dataset *Tr* is partitioned into two datasets *M1tr and M2tr*. *M1tr* contains 50% randomly drawn data from *Tr* and rest 50% data in taken in *M2tr*.

Step 2 : Two testing datasets, *TE1 and TE2* have been generated containing 30% data of *M1tr and M2tr* in each respectively.

Step 3 : A new dataset containing the instances of *M1tr and M2tr* which have not been used in *TE1 and TE2* is generated and is referred to as *Data0*. This dataset will become the part of RRF model training process in Algorithm 2.

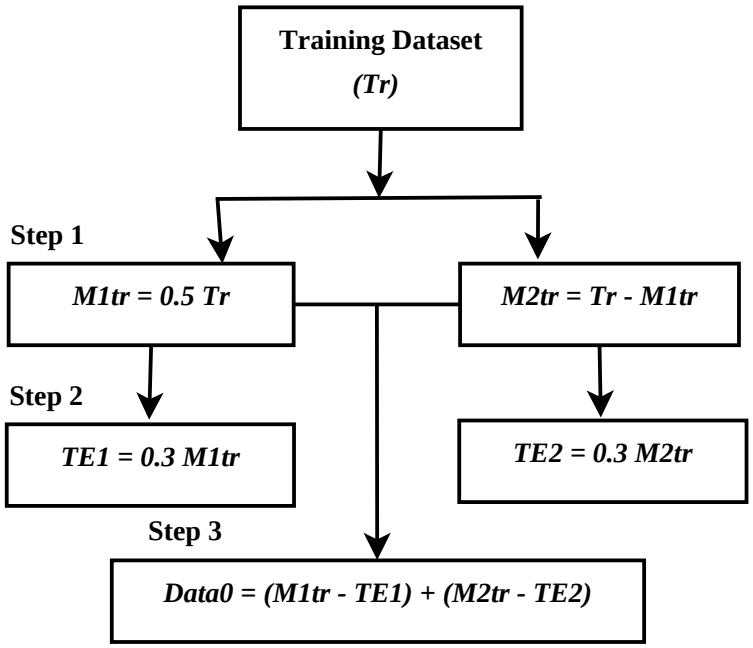

Figure 4.3: Data partitioning for the proposed ensemble model.

properly and will be able to produce relevant predictions. Figure 4.4 represents four phases which are required in the fusion of three models. In Phase I, Decision tree and avNNet models are trained and tested on different subsets of the dataset. Phase II consists of error calculation and generation of weights on the testing dataset. In Phase III, new sets of data are generated which have weights as a new column. Finally, in Phase IV, a combined set of data is used to train the RRF model. These four Phases are described in Algorithm 2 which are followed in the ensembling process.

Algorithm 2 This algorithm describes the phases to design proposed ensemble model. Terms used in this algorithm are:

Actual value of peptide: 0 or 1

Predicted probability: real number

Predicted value of peptide: 0 or 1(by rounding of (predicted probability))

n: total number of rows in *TE2* or *TE1*

m: total number of rows where actual value and predicted value are not same in *TE2*

r: total number of rows where actual value and predicted value are not same in *TE1*

Phase I : Decision tree and avNNet models are trained on *M1tr* and *M2tr* datasets respectively. For testing decision tree, *TE2* dataset is used whereas for avNNet model *TE1* is used.

Phase II : Decision tree will give prediction probabilities on *TE2 (*$te2_1$, $te2_2$,....., $te2_n$*)* and generate error set *Er1 (*$er1_1$, $er1_2$,....., $er1_n$*)*.

avNNet will give prediction probabilities on *TE1 (*$te1_1$, $te1_2$,....., $te1_n$*)* and generate error set *Er2 (*$er2_1$, $er2_2$,....., $er2_n$*)*.

Each error in sets *Er1 and Er2* is generated by subtracting predicted probability from the actual value of a peptide.

$$er1_i = \textit{Predicted prob } (te2_i)\textit{-Actual value } (te2_i), i = 1, 2, ..., n \tag{4.1}$$

$$er2_i = \textit{Predicted proba } (te1_i)\textit{-Actual value } (te1_i), i = 1, 2, ..., n \tag{4.2}$$

Weight sets *Weight1 (*$weight1_1$, $weight1_2$,....., $weight1_n$*) and Weight2 (*$weight2_1$, $weight2_2$,....., $weight2_n$*)* corresponding to *Er1 and Er2* are thus randomly generated by adding 1 in each error instance.

$$weight1_i = er1_i + 1, i = 1, 2, ..., n \tag{4.3}$$

$$weight2_i = er2_i + 1, i = 1, 2, ..., n \tag{4.4}$$

Phase III : Now, two datasets *Data1 and Data2* are generated by extracting those rows of *TE2 and TE1* respectively where actual and predicted value of a given peptide is not same.

if(Actual value ($te2_i$*)≠Predicted value (*$te2_i$*))*
$Data1_k = te2_i$, k = 1, 2,....., m; i = 1, 2,....., n $m \leq n$ *end*

if(Actual value ($te1_i$*)≠Predicted value (*$te1_i$*))*
$Data2_k = te1_i$, k = 1, 2,....., r; i = 1, 2,....., n $r \leq n$ *end*

Now, *Data1 and Data2* contain one additional column of weights (as described in phase II) with respect to each instance.

Phase IV : *Data0, Data1 and Data2* are finally combined to generate a new dataset *M3tr* which is further used to train the RRF model. For *Data0* each instance is given weight 1 [98]. Since each instance in *M3tr* contains an associated weight as calculated in Phase II, therefore, RRF model is now trained with weighted instances which will help the model to learn the importance of each instance for better model training.

To predict the test and benchmark datasets, contribution of these three trained models (decision tree, avNNet and RRF) is required and ensembling of their prediction is done by using weighted voting.

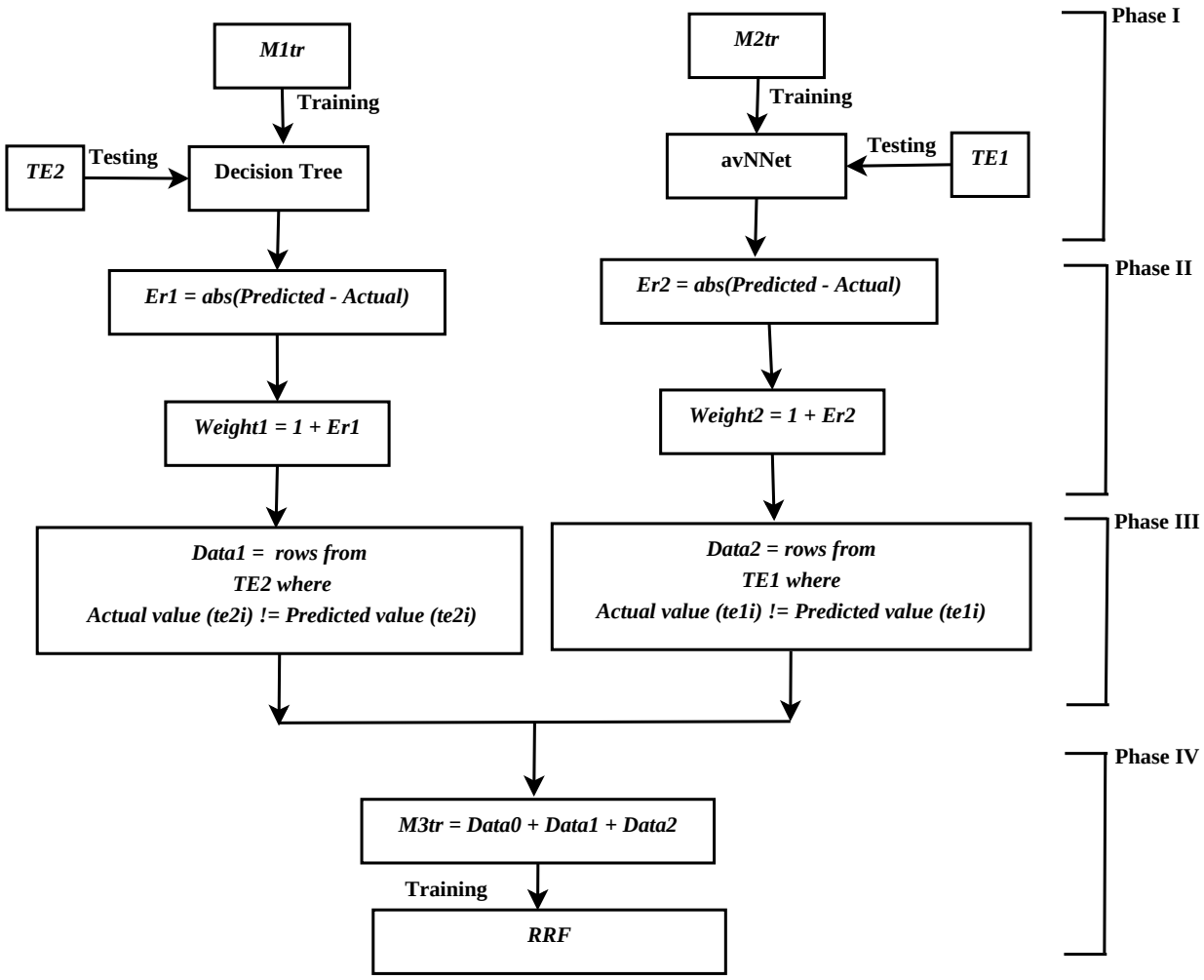

Figure 4.4: The proposed ensemble model for prediction of T-cell M. tuberculosis epitopes.

4.3.3 Analysis of Improvement in the Results

By using the above proposed ensemble model for M. tuberculosis epitope prediction, all the considered evaluation parameters are improved because of the following reasons:

1. Prediction from the single model is less reliable as compared to prediction from group of models. In the proposed ensemble model, errors of the confusion matrix which are false positive (FP) and false negative (FN) have been refined.

2. While training the RRF model in Phase IV, *M3tr* is used which contains weights corresponding to each false predicted instances *i.e.* FP and FN. This model has been trained more efficiently by using false predicted instances and data from *M1tr and M2tr*. Therefore, it helps the RRF model to enhance it's predictability.

3. The complete dataset is circulated to each model in such a way that they can learn it properly.

The impact of data division and ensemble model has been described through evaluation parameters as discussed in Table 4.5.

4. In this Chapter, weighted voting is used to combine the results of trained models. Instead of considering a class (0 or 1), prediction probability of each class has been used.

4.4 Model Evaluation

In model evaluation phase, the proposed ensemble model has been tested by using various evaluation parameters, cross validation and benchmark dataset. The outcome of this phase concludes that how well the model has learnt the data and how efficiently it produces predictions. Equation 4.5 shows the features which are considered to train the models. The performance of the model has been evaluated by various parameters such as Gini, accuracy, AUC, specificity and sensitivity as described in Chapter 1. While training the models, problems including overfitting/underfitting/biasness may occur. To handle these issues and to test the robustness of the proposed ensemble model, repeated k-fold cross validation has been performed. To validate and compare the performance of proposed ensemble model, a benchmark dataset has been used.

After applying feature selection approach on the dataset, four features (F_{ab}, F_t, F_n and F_n) are discarded and rest of the features are considered to train the model as mentioned in Section 4.3.1. The formula for the training process of the three individual models and the proposed ensemble model is:

$$CL \sim f(F_a, F_b, F_c, F_d, F_e, F_f, F_g, F_h, F_i, F_j, F_l, F_m, F_o, F_p, F_q, F_r, F_s, F_u, F_v, F_w, F_x, F_y, F_z, F_{aa}, F_{ac}, SL) \tag{4.5}$$

4.4.1 Repeated K-Fold Cross Validation

Cross validation is used to evaluate and compare the models by dividing dataset into two portions; one portion is used to train the model and other is used to validate the model. The main goal of cross validation is to make sure that each portion of dataset gets equal chance to occur in the training and testing datasets. In k-fold cross validation, each portion of the dataset comes under testing dataset exactly once and rest of the time it is considered in training dataset. To increase the occurrence of each portion in testing dataset, repeated k-fold cross validation is used. Repeated k-fold cross validation is same as that of k-fold cross validation but the number of iterations get increased. Here, dataset has been divided into 10 folds and each fold has been executed 5 times.

4.4.2 Benchmark of the Proposed Model Correctness

Another way to validate the model is to test it with the benchmark dataset. In the benchmark dataset, 30 epitopes of T-cell M. tuberculosis have been collected from literature [99, 100] and MtbVeb server (*http://crdd.osdd.net/raghava/mtbveb/*). 16 non-epitopes are extracted from MHCBN version 4.0 [101] server. All these epitopes and non-epitopes are unique. These epitopes and non-epitopes are not present in training and testing datasets.

Table 4.6 describes the 30 T-cell epitopes of M. tuberculosis and 16 non-epitopes with predicted binding capacity of peptide from NetMHC 2.3 and NetMHC 4.0 existing servers and predictions from the proposed ensemble model. The proposed ensemble model is efficient enough to predict benchmark dataset. The outcome of proposed ensemble model suggests that the peptide is M. tuberculosis epitope or a non-epitope which is not concluded by the results of NetMHC 2.3 and NetMHC 4.0 servers.

The benchmark dataset has been used to compare the performance of proposed ensemble model with existing CTLpred server. CTLpred online server is able to predict T-cell epitopes having length 9-mers. The proposed ensemble model outperforms CTLpred server as shown in Table 4.7.

4.5 Evaluation of Results

The machine learning models have been trained efficiently on the given dataset as discussed in Section 4.3.2. Since data plays an important role in the model training process, therefore, in this Chapter, data division has been carried out in an efficient way as mentioned in Section 4.3.2.1. Individual models are selected on the basis of their performance. Like, a set of two strong models and one weak model is taken to design ensemble model. The proposed ensemble model overcomes the false predictions and enhances the predictability as compared to the individual models.

The results are shown in Table 4.5 which illustrate that the performance of the proposed ensemble model is better than the individual models. After training the model, there is a possibility that it can be overfitted/underfitted/baised. To handle these issues and to check the robustness of the proposed ensemble model, 5 times 10 fold cross validation and the benchmark dataset are used. In repeated k-fold cross validation, dataset is divided into ten portions, each containing an equal number of epitopes *i.e.* 360 each. For each run, accuracy is shown in Figure 4.5. The average accuracy of

the proposed ensemble model comes out to be 80.61% in 5 times 10 fold cross validation. Therefore, the proposed ensemble model overcomes the issues like overfit/underfit/baiseness.

The ROC curve plot for the proposed and the individual models is shown in Figure 4.6. Since, quality of the model is measured from the bent of the curve towards the upper left corner, it can be concluded by seeing the plot in Figure 4.6 that the performance of proposed ensemble model is better than the other three models.

Table 4.5: Performance evaluation on various parameters of the individual and proposed ensemble models.

SN	Model Name	Gini	ACC	AUC	Spec	Sens
1	AvNNet	0.01	51	0.5	0.52	0.48
2	ELM	0.05	53	0.52	0.46	0.58
3	Glmboost	0.21	60	0.6	0.53	0.67
4	GAM	0.23	63	0.61	0.58	0.65
5	NN	0.25	64	0.62	0.6	0.66
6	Blackboost	0.37	70	0.68	0.69	0.7
7	Gamboost	0.38	72	0.69	0.83	0.68
8	SVM	0.39	72	0.69	0.76	0.7
9	RRF	0.56	78	0.78	0.82	0.74
10	Decision Tree	0.56	78	0.78	0.87	0.71
11	**Proposed ensemble model**	**0.65**	**82**	**0.82**	**0.89**	**0.77**

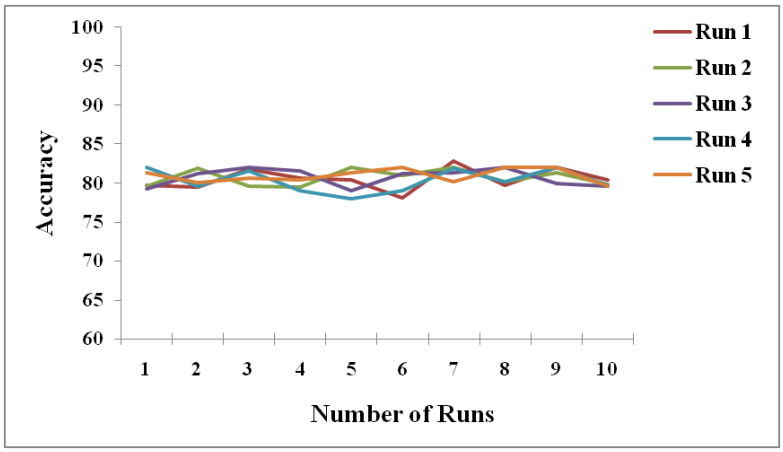

Figure 4.5: Repeated 10-fold cross validation of proposed ensemble model.

4.6 Discussion

The proposed ensemble model is compared with the existing systems.

4.6.1 Comparison of the Proposed Model with NetMHC 2.3 and NetMHC 4.0 Servers

The Table 4.6 contains the benchmark dataset (as mentioned in Section 4.4.2) with predictions from the proposed ensemble model, NetMHC 2.3 [102] and NetMHC 4.0 [102] servers. As shown in table, all the epitopes are correctly predicted by the proposed model and shows it's reliability. NetMHC 2.3 and NetMHC 4.0 existing servers provide binding capacity of the peptide which can be weak or strong binder based upon the thresholds defined on ranks. Even some peptides do exist which don't lie between the given range of ranks. Those peptides are considered as Out of Range. However, the strong binder peptides have the higher chance to be an epitope. But, it can not be surely concluded that weak and Out of Range peptides cannot be proved to be an epitope in future. This limitation of NetMHC 2.3 and NetMHC 4.0 servers has been overcame by the proposed ensemble model which will classify the peptide as M. tuberculosis epitope or a non-epitope.

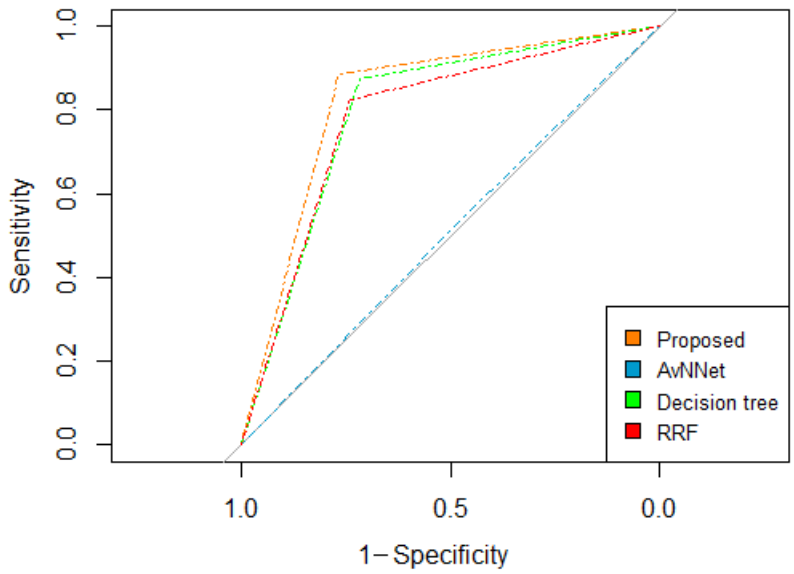

Figure 4.6: ROC curves of proposed ensemble and single models.

4.6.2 Comparison of the Proposed Model with CTLpred Server

The proposed ensemble model is then compared with the CTLpred [35] T-cell prediction server. CTLpred server classifies the T-cell epitope or non-epitope of length 9-mers. The comparison between proposed ensemble model and CTLpred server is given in Table 4.7. This comparison shows that the proposed ensemble model performs better than CTLpred. The proposed model is also capable of predicting epitope of any length.

Thus, after validation and comparison, finally it can be concluded that the proposed ensmeble model is adequate for the prediction of M. tuberculosis epitope.

4.6.3 Other Discussion

In CTLpred [35], they had introduced a direct method for predicting cytotoxic T lymphocyte epitopes. Since, there are indirect methods to predict T-cell epitope which predict MHC class I binders instead of CTL epitopes. They used quantitative matrix and machine learning models including SVM and NN. The accuracy scored by NN is better than quantitative matrix and SVM. The proposed model has contributing in the direct prediction of M. tuberculosis epitopes. There are many indirect methods to predict M. tuberculosis epitopes which predict MHC class I or MHC class II binding capacity of the peptide (as described in Chapter 2) instead of M. tuberculosis epitope.

4.7 Conclusion

In this Chapter, direct method is used to predict T-cell M. tuberculosis epitopes instead of predicting binding capacity of the peptide. By using proposed ensemble model, prediction of T-cell M. tuberculosis epitopes is enhanced as compared to the individual models and existing systems. Three models (decision tree, avNNet and RRF) have been used to design the ensemble model which produces high accuracy, Gini, AUC, specificity and sensitivity. While training the proposed ensemble model, data division is uniquely performed which refines the false predictions. The benefit of using this approach is to enhance the performance and effectiveness of the proposed model. The data is circulated to three individual models which train them adequately in order to provide reliable and accurate results. To analyze the robustness of proposed ensemble model, repeated k-fold cross validation has been used. To validate and compare the proposed ensemble model, NetMHC 4.0,

NetMHC 2.3 and CTLpred servers have been considered. The results conclude that the proposed ensemble model is capable of producing effective predictions. We believe that outcome may be improved by using more properties of epitopes, different preprocessing techniques and machine learning models.

Table 4.6: Performance comparison of the proposed ensemble model with NetMHC 2.3 and NetMHC 4.0.

MHC I	% Rank by NetMHC 4.0 Server	Target	Prediction	MHC II	% Rank NetMHC 2.3 Server	Target	Prediction
ARVESRTYI	50	1	1	AHRRFAAAFAAVLLAVVCL	0.17	1	1
ELTQPDRVV	28	1	1	FRRRNPRPAIVVVAFLVV	24.0	1	1
GLLSWVEEV	0.01	1	1	GLVFLAVLVIFAIIVQ	90.0	1	1
GSEEEFQRL	33	1	1	HVVVGAAVLAFVAVVV	42.0	1	1
KHKNSYLAL	50	1	1	IKIFMLVTAVVLLC	35.0	1	1
RLCDQLVEA	0.8	1	1	MRYLIATAVLVAVVLVG	3.00	1	1
SWVEEVAEL	12	1	1	PMRMLVALLLS	17.0	1	1
VEAGTFIRL	22	1	1	SLVRIVGVVVATTLAL	44.0	1	1
VRAVRVPNSFILATNFSF	13	1	1	TRRMYSNYGF	17.0	1	1
FLENFVRSSNL	5	1	1	TSNVSVAKIAFTGVL	35.0	1	1
FLTSELPQWL	0.3	1	1	TVLLDANVLIALVVA	37.0	1	1
GLSIVMPV	0.8	1	1	VNLVDTLNSGQYTVFAPTNAA	28.0	1	1
KLVANNTRLWV	5	1	1				
LTSELPQWL	5.5	1	1				
QTYKWETFLT	15	1	1				
RLWVYCGNGT	4.5	1	1				
SMAGSSAMIL	3.5	1	1				
YLLDGLRA	1.2	1	1				
PEGRAWAQPGYPWPA	12	0	0				
KEFRKTKRNTLRRPA	80	0	0				
ATRKTSERSQPRGRA	70	0	0				
GMGWAGWLLSPRGSA	16	0	0				
KARQPEGRAWAQPGA	49	0	0				
KEFRKTKRNTLRRPA	80	0	0				
LYGNEGMGWAGWLLA	5	0	0				
MSTNPKEFRKTKRNA	70	0	0				
IPFPIVRYL	15	0	0				
PTDPRRRSRNLGKVA	90	0	0				
RLGVRATRKTSERSA	34	0	0				

to be cont'd on next page

Table 4.6: Performance comparison of the proposed ensemble model with NetMHC 2.3 and NetMHC 4.0. (cont.)

MHC I	**% Rank by NetMHC 4.0 Server**	**Target**	**Prediction**	**MHC II**	**% Rank NetMHC 2.3 Server**	**Target**	**Prediction**
RPSWGPTDPRRRSRA	85	0	0				
AYLKQATAK	85	0	1				
ARSMAAAAA	43	0	1				
ERSMAAAAA	65	0	1				
RRGPRLGVRATRKTA	38	0	0				

Thresholds for MHC I: According to NetMHC 4.0- $\%Rank < 0.5$ *strong binder*, $2 > \%Rank > 0.5$ *weak binder*.
Thresholds for MHC II: According to NetMHC 4.0- $\%Rank < 2.00$ *strong binder*, $10.00 > \%Rank > 2.00$ *weak binder*.

Table 4.7: Performance comparison with existing CTLpred server and the proposed ensemble model.

Sr. No.	Peptide	Actual	CTLpred	Proposed ensemble model
1	AHRRFAAAFAAVLLAVVCL	Epitope	–	Epitope
2	ARVESRTYI	Epitope	Epitope	Epitope
3	ELTQPDRVV	Epitope	Non-epitope	Epitope
4	FLENFVRSSNL	Epitope	–	Epitope
5	FLTSELPQWL	Epitope	–	Epitope
6	FRRRNPRPAIVVVAFLVV	Epitope	–	Epitope
7	GLLSWVEEV	Epitope	Epitope	Epitope
8	GLSIVMPV	Epitope	–	Epitope
9	GLVFLAVLVIFAIIVQ	Epitope	–	Epitope
10	GSEEEFQRL	Epitope	Non-epitope	Epitope
11	HVVVGAAVLAFVAVVV	Epitope	–	Epitope
12	IKIFMLVTAVVLLC	Epitope	–	Epitope
13	KHKNSYLAL	Epitope	Non-epitope	Epitope
14	KLVANNTRLWV	Epitope	–	Epitope
15	LTSELPQWL	Epitope	Non-epitope	Epitope
16	MRYLIATAVLVAVVLVG	Epitope	–	Epitope
17	PMRMLVALLLS	Epitope	–	Epitope
18	QTYKWETFLT	Epitope	–	Epitope
19	RLCDQLVEA	Epitope	Epitope	Epitope
20	RLWVYCGNGT	Epitope	–	Epitope
21	SLVRIVGVVVATTLAL	Epitope	–	Epitope
22	SMAGSSAMIL	Epitope	–	Epitope
23	SWVEEVAEL	Epitope	–	Epitope
24	TRRMYSNYGF	Epitope	–	Epitope
25	TSNVSVAKIAFTGVL	Epitope	–	Epitope
26	TVLLDANVLIALVVA	Epitope	–	Epitope
27	VEAGTFIRL	Epitope	Epitope	Epitope
28	VNLVDTLNSGQYTVFAPTNAA	Epitope	–	Epitope
29	VRAVRVPNSFILATNFSF	Epitope	–	Epitope
30	YLLDGLRA	Epitope	–	Epitope
31	PEGRAWAQPGYPWPA	Non-Epitope	–	Non-Epitope
32	KEFRKTKRNTLRRPA	Non-Epitope	–	Non-Epitope
33	ATRKTSERSQPRGRA	Non-Epitope	–	Non-Epitope
34	GMGWAGWLLSPRGSA	Non-Epitope	–	Non-Epitope
35	KARQPEGRAWAQPGA	Non-Epitope	–	Non-Epitope
36	KEFRKTKRNTLRRPA	Non-Epitope	–	Non-Epitope
37	LYGNEGMGWAGWLLA	Non-Epitope	–	Non-Epitope
38	MSTNPKEFRKTKRNA	Non-Epitope	–	Non-Epitope
39	IPFPIVRYL	Non-Epitope	Epitope	Non-Epitope
40	PTDPRRRSRNLGKVA	Non-Epitope	–	Non-Epitope
41	RLGVRATRKTSERSA	Non-Epitope	–	Non-Epitope
42	RPSWGPTDPRRRSRA	Non-Epitope	–	Non-Epitope
43	AYLKQATAK	Non-Epitope	Epitope	Epitope
44	ARSMAAAAA	Non-Epitope	Non-Epitope	Epitope
45	ERSMAAAAA	Non-Epitope	Non-Epitope	Epitope
46	RRGPRLGVRATRKTA	Non-Epitope	–	Non-Epitope

-: CTLpred server only predicts 9 length epitopes

Chapter 5

Improvement in Prediction of Antigenic Epitopes using Stacked Generalization: An Ensemble Approach

The major intent of peptide vaccine designs, immunodiagnosis and antibody productions is to accurately identify linear B-cell epitopes. The determination of epitopes through experimental analysis is highly expensive. Therefore, it is desirable to develop a reliable model with significant improvement in prediction models. In this Chapter, a hybrid model has been designed by using stacked generalization ensemble technique for prediction of linear B-cell epitopes. The goal of using stacked generalization ensemble approach is to refine predictions of base classifiers and to get rid of the worse predictions. In this Chapter, six machine learning models are fused to predict variable length epitopes (6 to 49 mers). The proposed ensemble model achieves 76.6% accuracy and average accuracy of repeated 10-fold cross validation is 73.14%. The trained ensemble model has been tested on the benchmark dataset and compared with existing sequential B-cell epitope prediction techniques including APCpred, ABCpred, BCpred and AAP_{BCPred}. [†]

5.1 Introduction

Interaction between antigen and antibody plays a vital role in the humoral immune response. Antigenic determinants (epitopes) are the specific region of antigens where antibodies bind. B-cell epitopes can be categorized into two parts: sequential and discontinuous epitopes. Sequential epitopes are the ones which have amino acids lying linearly in the polypeptide chain. The discontinuous epitopes are generated by using amino acids which are located in different segments of the polypeptide chain. 10% of epitopes are sequential and the rest are discontinuous. Disclosure of sequential epitopes play an important role in experimental designs, immunodiagnostic tests and vaccine production [82] where most of the B-cell epitopes are discontinuous.

[†]D Khanna and PS Rana, Improvement in Prediction of Antigenic Epitopes using Stacked Generalization: An Ensemble Approach, IET Systems Biology, IET 14 (1) (2019) 1-7.

Traditionally, single property of amino acid was used to describe the information of sequence including hydrophilicity [52], antigenicity [53] and surface accessibility [54]. Later on, more than one physicochemical properties had been employed in the methods like PREDITOP [55], PEOPLE [56], BEPITOPE [57] and BcePred [58]. ABCPred [5], SVM based model [38], BCPred [37], BEST [61], SVM model [3], RF based model [1], LBE predictor [62] and APCpred [36] were used to predict B-cell epitopes.

Biologists recognise B-cell epitopes to generate peptide based vaccines, epitope based antibodies and diagnostic tools. Without computer interference, biologists identify B-cell epitopes by doing experiments in the wet labs. While doing experiments, they have to test all the peptides individually to get B-cell epitope. This makes their task tedious in terms of efforts, cost and time. To make biologist's task easy, an accurate statistical model is required which can predict whether a peptide is an epitope or a non-epitope. Therefore, machine learning techniques are used to generate predictions which reduce the human efforts, time, cost and wet lab experiments. Machine learning technique is beneficial because it facilitates the computer to understands the hidden patterns within the dataset and produces predictions on the unknown data without human interference. Therefore, with the help of machine learning techniques only those samples which are filtered by these techniques are used in the wet labs for further analysis like in experiments, peptide based vaccines, epitope based antibodies and diagnostic tools.

In this Chapter, the large number of peptides are given to the machine learning models and they predict whether that peptide is an epitope or a non-epitope. The filtered peptides which are epitopes according to the models are used for further analysis rather than using all the peptides. This makes biologist's job easy by reducing time, cost and efforts for identifying B-cell epitopes.

Inspired from the performance of machine learning models and need to find a reliable model which can predict antigenic epitopes and reduces the expense on the experimental testing of epitopes, a hybrid method has been proposed by using stacked generalization ensemble technique. To train the models, physicochemical properties of amino acids are used which in turn classify the sequential B-cell epitopes as described in Section 5.3. From literature survey, some shortcomings of B-cell epitope prediction methods have been found which includes feature selection phase [3, 5, 37, 38], fixed length of amino acid sequences [3, 5, 38, 61], small dataset and basic models (RF, SVM, Neural network). Feature selection phase is essential because it reduces complexity of dataset and enhances

the performance of model. Model trained with fixed length of epitopes is used to predict fixed length of epitopes. Nowadays, flexible model is required which can predict any length of epitope. The effectiveness of model is dependent on the size of the training dataset. The datasets used in existing methods [1, 3, 5, 36–38, 61] contain approximately 700, 2479, 701, 4925, 2479, 727, 727, 1573 antigenic epitopes respectively.

In order to overcome the above stated flaws, the contributions of the proposed ensemble model are stated below:

- The proposed ensemble model is a combination of six models (Blackboost, Regularized RF, SVM, RF, GBM and avNNet) which has been explained in Section 5.3.2. It is different from existing sequential B-cell prediction techniques because such techniques are based upon single model (mostly used models RF, SVM and Neural Network), which may produce false predictions.

- In this Chapter, variable length epitopes (6 to 49 mers) are used to train the models. 45,320 epitopes are taken out of which 21,999 are positive and rest are negative.

- The features of amino acids have been filtered by using boruta [103] as mentioned in Section 5.2.3. Boruta feature selection algorithm is based upon wrapper technique which uses RF model to eliminate the least important features and gives important features to train the models.

- There are many approaches like bagging, boosting and stacked generalization to create an ensemble model. In this Chapter, stacked generalization ensemble technique has been used. One of the benefits, for selecting stacked generalization technique is to refine the output of the base classifier. The models are then linked with each other in such a way that wrong prediction by one model may be corrected by the other model which produces stable and effective results.

- There exist many sequential B-cell epitope prediction techniques. The comparison between some targeted techniques and the proposed ensemble model is performed. It describes that the proposed ensemble model enhances the accuracy of prediction model which is shown in Table 5.4.

- The proposed model will be beneficial for the biologists because of its predictability. Only filtered epitopes will be available to them which decreases the expenditure cost to do the experiments in wet lab.

The Chapter is structured as: the brief of the dataset, feature extraction, feature importance, the models of machine learning and the benchmark dataset are mentioned in Section 5.2. The proposed methodology and ensemble model are narrated in Section 5.3. Section 5.4 consists of model evaluation process. Section 5.5 contains result analysis, comparison and discussion. In the end, the conclusion and future work are mentioned in Section 5.6.

5.2 Materials and Methods

The section describes the dataset, feature extraction, feature importance, machine learning models and the benchmark dataset.

5.2.1 Dataset and its Features

Normally, 10% epitopes are sequential and the rest are discontinuous. In this Chapter, continuous epitopes have been considered. The dataset of sequential B-cell epitopes which contains positive and negative epitopes, is accessed from LBtope server [104, 105]. The extracted dataset is imbalanced thus to handle this issue, fixed length of epitopes are added to the dataset. Fixed length epitopes are extracted from the same source. After removing duplicate sequences and imbalanced class handling, 45320 sequences are obtained which are of variable length ranging from 6 to 49 mers. There are 21,999 positive and rest are negative sequences. An example of the dataset is presented in Table 5.1.

Table 5.1: Sample dataset containing the sequence length, physicochemical properties of amino acid and class of epitopes.

***SL**	$\mathbf{F}_a$	$\mathbf{F}_b$	$\mathbf{F}_c$	$\mathbf{F}_d$	——	$\mathbf{F}_z$	$\mathbf{F}_{aa}$	$\mathbf{F}_{ab}$	$\mathbf{F}_{ac}$	$^{\#}$**CL**
12	57.67	4.08	1.46	-1.00	——	34.33	9.33	26.00	5	0
15	131.0	0.80	1.38	1.09	——	21.00	14.33	7.67	3	0
8	49.75	3.62	1.50	4.09	——	51.00	51.00	1.00	8	0
20	84.00	1.91	1.25	-1.00	——	21.00	6.00	16.00	4	1
6	82.67	2.35	1.35	1.09	——	17.67	17.67	1.00	6	1
49	98.35	3.70	1.62	2.95	——	45.90	25.49	21.41	540	1

*SL represents sequence length; $^{\#}$CL represents class label *i.e.* 1 means antigenic epitope and 0 means non antigenic epitope

5.2.2 Feature Extraction

In feature extraction phase, a set of features is defined which represents meaningful information about the area of interest and that set is important for the further analysis. To increase the accuracy and

effectiveness of supervised learning, feature extraction phase is essential. In this Chapter, twenty-nine different physicochemical properties of amino acids including Aliphatic index (F_a), Potential protein interaction index (F_b), Hydrophobic moment (F_c), Instability index (F_d), Probability of detection of peptides (F_e), Number of possible neighbours F_f, Tiny (F_g), Small (F_h), Aliphatic (F_i), Aromatic (F_j), Nonpolar (F_k), Polar (F_l), Charged (F_m), Basic (F_n), Acidic (F_o), Percentage of tiny (F_p), Percentage of small (F_q), Percentage of aliphatic (F_r), Percentage of aromatic (F_s), Percentage of nonpolar (F_t), Percentage of polar (F_u), Percentage of charged (F_v), Percentage of basic (F_w), Percentage of acidic (F_x), Charge of protein sequence (F_y), Hydrophobicity (F_z), Kidera factor (F_{aa}), Molecular Weight (F_{ab}), Isoelectric point (F_{ac}) are used and described in Chapter 3. All these properties have been extracted by using R which is an open source software licensed under GNU GPL and calculated with default parameters of all the functions.

5.2.3 Boruta for Feature Importance

In the feature importance phase, those features are removed which are highly correlated with other feature, biases and noise from the data. It filters the required features which improves the performance of model. Huang [1] uses RF model's inbuilt property to select important features in which mean decrease in accuracy is used to get important features. Three different sets of features are then created based upon their importance values larger than 0.05, 0.1, and 0.15 respectively. Now, a model is trained multiple times depending on the sets of features.

Motivated from this property of RF model and to reduce the overhead of training the model with different set of features, boruta [103] algorithm has been used which gives a list of important, unimportant and tentative features.

In this Chapter, feature selection task has been done by Boruta [106–108] because it uses RF results and z score to find out the importance of a feature. The boruta feature selection algorithm is based upon wrapper technique which uses RF to eliminate the less important features. The RF [87] has been selected because it uses an ensemble approach and has low-cost calculations. It gathers votes from all decision trees which are based upon weak classifiers. The z score is calculated by dividing mean loss of accuracy by its standard deviation. In boruta, the dataset is shuffled by creating random copies of all the features which are known as the shadow features. It trains the RF model on the huge dataset and applies a feature significance measure to know the importance of every

feature. It repeatedly checks in each run that a real feature has a high significance than the best of its shadow features and constantly removes insignificant features. At final stage, the stopping criteria of algorithm is when all the elements get accepted or rejected or it achieves a predefined breaking point of RF runs.

Boruta can be executed in Python, R and does tedious work in a simple way [109]. To filter out the features, boruta gives a list of important, unimportant and tentative features. There is not any tentative feature in the given dataset. The important features are filtered out which are used to train the models. According to the boruta algorithm, features F_e and F_{ab} are least important. So these are discarded and rest are considered to train the model. Figure 5.1 represents the importance of features in which green

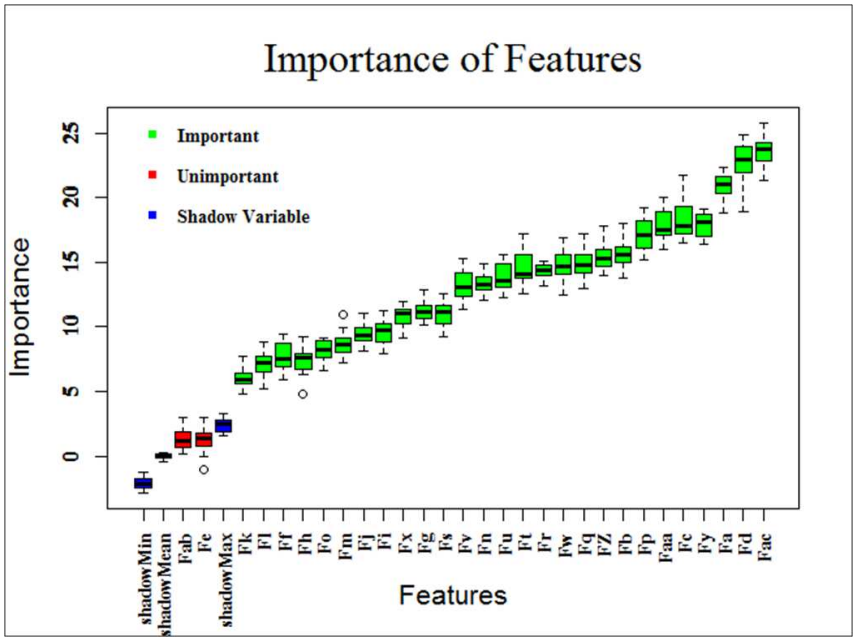

Figure 5.1: A plot representing the importance of each feature calculated by using boruta algorithm.

represents the important features, red represents the unimportant features and blue represents shadow min, shadow mean and shadow max. Variables having z score less than shadow variables are marked as unimportant and hence discarded.

5.2.4 Machine Learning Methods

Table 5.2 shows the models which are used in this Chapter. It describes required packages and default tuning parameters which are used in execution of the models. To get better results, models can be tuned but in this Chapter default values of parameters are considered.

Table 5.2: Machine learning models considered for ensembling; their respective R packages, methods and tuning parameters.

Machine learning Model	Function	Package	Tuning Parameter
BlackBoost [95]	blackboost	mboost	None
Avnnet [92]	avNNet	caret	size=5, linout=TRUE, trace=FALSE
RRF [86]	RRF	RRF	None
SVM [97]	ksvm	kernlab	kernel=”rbfdot”, prob.model=TRUE
RF [110]	randomForest	randomForest	ntree=500,mtry=3
GBM [111]	gbm	gbm	var.monotone,distribution=”gaussian” , n.trees=1000

5.2.5 Benchmark of the Proposed Ensemble Model Correctness

For the benchmark of proposed ensemble model correctness, benchmark dataset is collected from Shen [36] and he has provided the comparison of ABCpred, BCpred, AAP_{BCPred} with APCpred. The benchmark dataset is composed of 187 epitopes and 200 non-epitopes of length 16-mers [5]. The benchmark dataset is used to test the proposed ensemble model and compared with APCpred, ABCpred, BCpred, AAP_{BCPred} techniques. There isn't any overlapping between training peptide data and benchmark peptide data. The proposed ensemble model and existing models have been evaluated on different parameters including accuracy, MCC, AUC, sensitivity and specificity as mentioned in the Table 5.4. Results reveal that the proposed ensemble model is performing well in comparison to the existing techniques which is discussed in Section 5.5.1.

5.3 Methodology

The proposed methodology is presented in Figure 5.2. Initially, the peptide sequences are extracted from LBtope server [104]. The dataset contains negative and positive epitopes having variable length ranging from 6 to 49 mers. The dataset is imbalanced thus to handle this issue, fixed length of epitopes are added to the dataset. Fixed length epitopes are extracted from the same source. In next step, the feature extraction is performed, as mentioned in Section 5.2.2. Duplicate and missing entries are eliminated from the dataset in the third step. In the fourth step, boruta algorithm [103] has been used to extract the important features. After these steps, the dataset is generated which is used to train the models. Table 5.2 represents the models which have been used in this Chapter. By using stacked generalization ensemble technique, six models have been combined as detailed in Section 5.3.2. The control flow of the proposed scheme has been represented in Figure 5.3 and discussed in Section

5.3.1. Finally, performances of the models have been evaluated on different parameters including specificity, AUC, accuracy, Gini and sensitivity. To rank the models on the basis of their evaluation parameters TOPSIS has been used. Section 5.5.1 describes the benchmark dataset which is used to validate the proposed ensemble model. Repeated k-fold cross validation has been used to measure the robustness of its predictability.

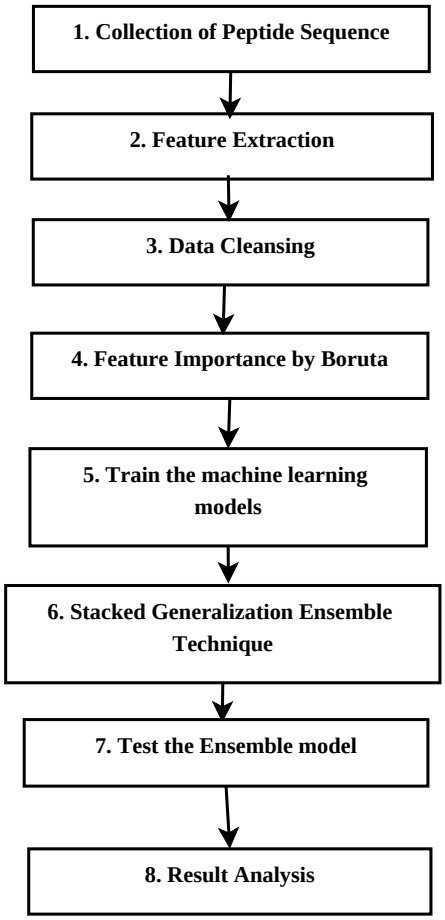

Figure 5.2: Methodology: step by step procedure of the proposed work.

5.3.1 Flow of the Proposed Scheme

Figure 5.3 shows the proposed ensemble model for prediction of antigenic epitopes. To train the models, a dataset which consists of B-cell epitopes with their twenty-nine physicochemical properties (Section 5.2) has been used. An ensemble model has been obtained by fusing six models as described in Section 5.3.2. The proposed ensemble model gives final prediction regarding the fact that whether an epitope is antigenic or non-antigenic.

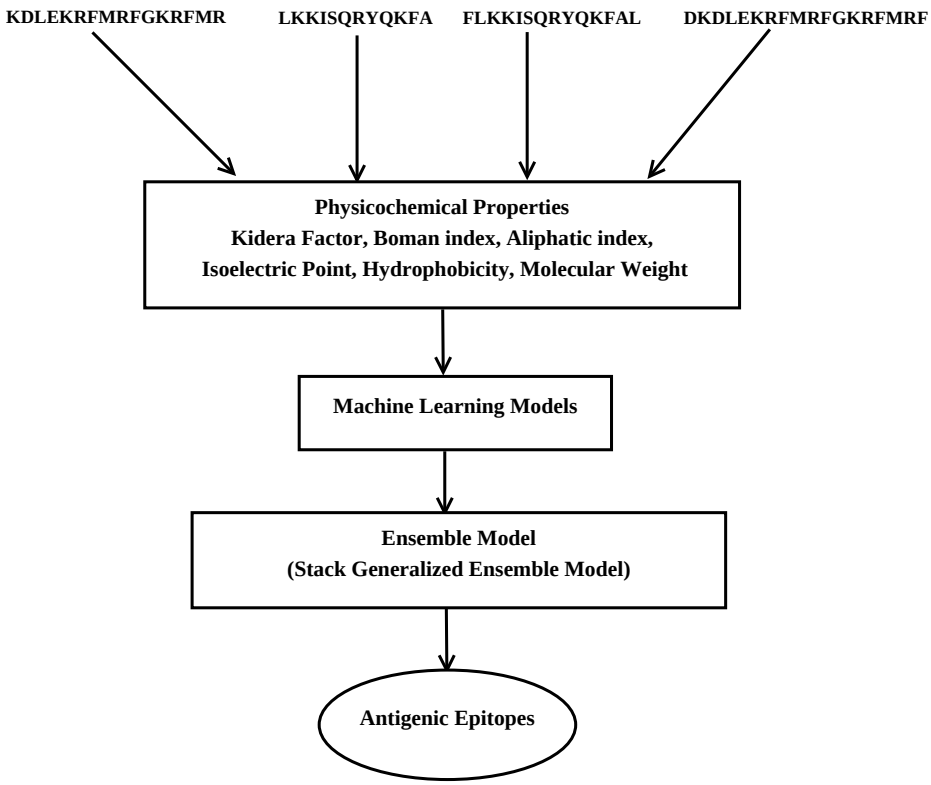

Figure 5.3: Workflow of the proposed ensemble model.

5.3.2 Proposed Stacked Generalized Ensemble Model

Ensembling has been performed to get rid of the worst prediction of the model. In this Chapter, major focus is on the refinement of predictions made by the base classifiers which has been dealt with stacked generalization ensemble technique. The combination six models including Blackboost, RRF, SVM, RF, avNNet and GBM are used to improve the accuracy as described in the Figure 5.4. 70% of dataset is used to train all these models and the rest of the dataset is used as testing dataset. The proposed ensemble model has been partitioned in 3 phases which are detailed below:

Phase 1 : Base classifiers in Tier 1 include RF, SVM and RRF which have been trained on the training dataset. To check whether the models have learned the training data properly or not, the trained models are also tested with training dataset.

Phase 2 : The predictions on training dataset from Phase I are used to create CTD1 dataset which is a combination of training dataset and predictions from RF, SVM and RRF. CTD1 (combined training dataset 1)dataset has been used to train the Tier 2 classifiers which includes blackboost and avNNet. These models are then tested by using CTD1 dataset. If a base classifier of Tier 1 incorrectly learned some particular instances, the second tier (Tier 2) classifiers can detect this

undesired behavior.

Phase 3 : The predictions on CTD1 dataset from Phase II are used to create CTD2 dataset which is a combination of CTD1 dataset and predictions from blackboost and avNNet models. Along with the learned behaviors of base classifiers, it can correct improper training. CTD2 dataset has been used to train GBM model which is Tier 3 meta classifier.

Final predictions on testing dataset are obtained by taking the average of minimum and maximum prediction probabilities of each instance. Here, minimum and maximum predicted probability of each instance has been obtained from six above described trained models. In this Chapter, instead of considering a class (0 or 1), prediction probability of each class has been used and hence it increases the impact of proposed ensemble model.

5.4 Model Evaluation

Model evaluation is an essential phase in developing the models. It suggests the model which efficiently represents the data and produces accurate predictions. While training the models, overfitting/underfitting/biasness problems may occur. Such issues are resolved by cross validation and benchmark dataset. Different parameters like Gini, accuracy, AUC, specificity and sensitivity are used to evaluate the performance of the model as mentioned in Chapter 1. TOPSIS a multiple criteria decision making method is used to rank the individual and proposed ensemble models on the basis of evaluation parameters. To analyze the consistency of the proposed ensemble model, repeated k-fold cross validation is performed.

After applying boruta algorithm on the dataset, 2 features (F_e and F_{ab}) are discarded and rest are considered as important which has been discussed in Section 5.2.3. In this Chapter, 5.1 is formulated by using important features and target class to train the models.

$$CL \sim f(F_a, F_b, F_c, F_d, F_f, F_g, F_h, F_i, F_j, F_k, F_l, F_m, F_n, F_o, F_p, F_q, F_r, F_s, F_t, F_u, F_v, F_w, F_x, F_y, F_z, F_{aa}, F_{ac}) \tag{5.1}$$

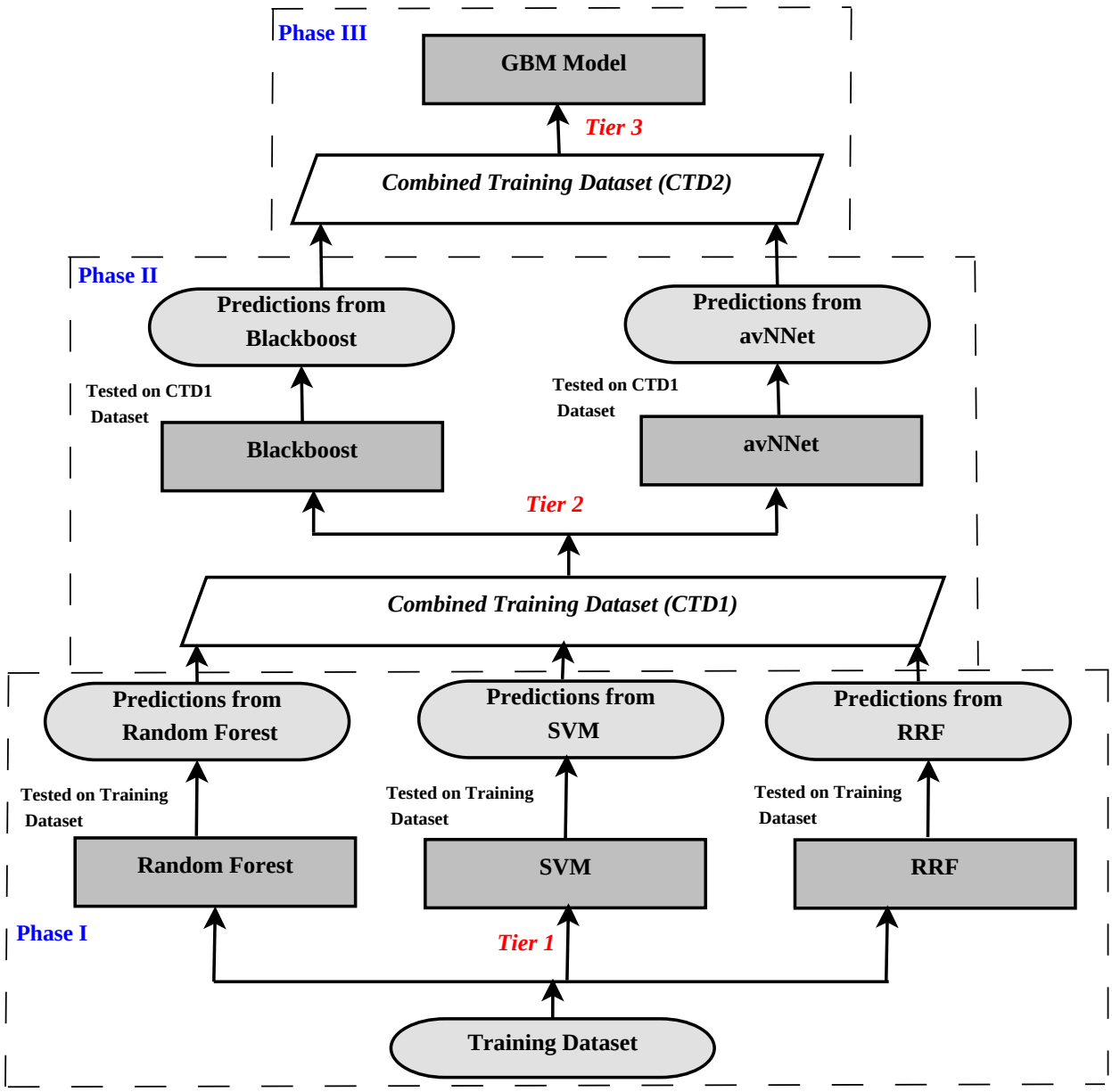

Figure 5.4: Procedural steps to build the proposed ensemble model.

5.4.1 Technique for Order Preference by Similarity to an Ideal Solution

To get an optimized result which is based upon the combination of these evaluation parameters, TOPSIS a multiple criteria decision making method has been used. It generates score by using these evaluation parameters and rank each model according to this score.

TOPSIS [28, 29] is one of the multiple criteria decision making methods. This technique is useful for decision makers to structure the problems to be solved, conduct analyses, comparisons and ranking of the alternatives. In other words, it is used to find out the combined solution which involves multiple criteria. In this Chapter, a R package named TOPSIS is used to get optimized result by using evaluation parameters. Rather giving importance to one evaluation parameter, all the evaluation

parameters are considered to generate the TOPSIS score which is used to rank all the individual and proposed ensemble models.

5.4.2 Repeated K-Fold Cross Validation

Number of iterations are beneficial for reliability comparison of model performance. Repeated k-fold cross validation has been used to increase the number of iterations or rerun the k-fold cross validation multiple times. On the other hand, in k-fold cross validation, it runs only k times. The data has been shuffled in each fold to do the comparisons. In this Chapter, 10-fold cross validation has been repeated for 5 times.

5.5 Result Analysis, Comparison and Discussion

In this Chapter, sequential B-cell epitopes have been considered because they are important for antibody production, experimental designs, immunodiagnostic tests and vaccine productions. There are some shortfalls in existing sequential B-cell epitopes prediction techniques which are discussed in Section 5.1. The proposed ensemble model has been used to overcome those shortfalls.

While training the models, problems like overfitting and underfitting can occur. An overfitted model learns too much and an underfitted model learns too less. In both the cases, results get fluctuated during every run. Solutions for such problems are cross validation and testing with unknown data. In the cross validation process, model runs n times and the accuracy is noted. If there is high fluctuation in the accuracy then it means the model is overfitted/underfitted/biased. Repeated k-fold cross validation has been performed and the accuracy is consistent. It shows that the proposed ensemble model is not affected from any issues as described above. For validation of the proposed ensemble model, benchmark dataset has been used. The output represents the two factors: former is, the proposed ensemble model is not overfitted/underfitted/biased and another one is, outcome of the proposed ensemble model is better than the existing techniques.

In this Chapter, boruta algorithm is used to select the important features. The impact of feature selection phase is demonstrated in Table 5.3. Evaluation parameters like accuracy, Gini, sensitively and AUC are boosted up by using feature selection phase. Therefore, for training the models, only important features are considered and rest are discarded.

The stacked generalized ensemble approach is used to train the machine learning models as

explained in Section 5.3.2. To create the proposed ensemble model, six models are used as mentioned in Table 5.2. 70% of the dataset is used to train these models and 30% is used as testing dataset. The performance of above individual models and the proposed ensemble model is shown in Table 5.3.

In this Chapter, multiple criteria decision making method TOPSIS is used to get the ranking of models on the basis of their evaluation parameters. The benefit of this technique is that the decision of selecting best model is based upon all the five evaluation parameters rather than any one or two parameters. The evaluation parameters get increased by using the proposed ensemble model. According to TOPSIS technique, the proposed ensemble model is at first rank which suggests that the proposed ensemble model is better than individual models.

The proposed ensemble model is compared with the existing techniques as mentioned in Section 5.5.1. Table 5.4 shows the performance of the proposed ensemble model and existing techniques on the benchmark dataset. TOPSIS and other evaluation parameters have suggested that the proposed ensemble model is outperforming the existing techniques.

To analyze the robustness of the proposed ensemble model, 10-fold cross validation is performed 5 times which scores mean accuracy of 73.13%. In cross validation process, dataset is divided into two sets: 70% for training and 30% for testing. Figure 5.5 shows the accuracy of the proposed ensemble model for 10 runs executed 5 times each which in turn represents the consistency in accuracy.

From the results and comparison, it is concluded that predictability of the proposed ensemble model has been improved significantly as compared to the individual models.

Table 5.3: Performance evaluation of machine learning models and the proposed ensemble model.

Model Name	**Spec**	**Sens**	**ACC%**	**Gini**	**AUC**	**TOPSIS Score**	**Rank**
RF	0.66	0.83	75	0.51	0.75	0.74	4
SVM	0.57	0.85	71.25	0.47	0.73	0.18	6
RRF	0.67	0.83	75.01	0.51	0.75	0.81	2
Blackboost	0.57	0.86	72.1	0.48	0.74	0.19	5
avNNet	0.54	0.87	71	0.47	0.73	0.06	7
GBM	0.67	0.82	75.05	0.51	0.75	0.80	3
Proposed Model	**0.70**	**0.82**	**76.6**	**0.52**	**0.76**	**0.94**	**1**

5.5.1 Performance Comparison on Benchmark Dataset

The proposed technique has been compared with existing sequential B-cell epitopes prediction techniques on the benchmark dataset. The results conclude that the proposed ensemble model

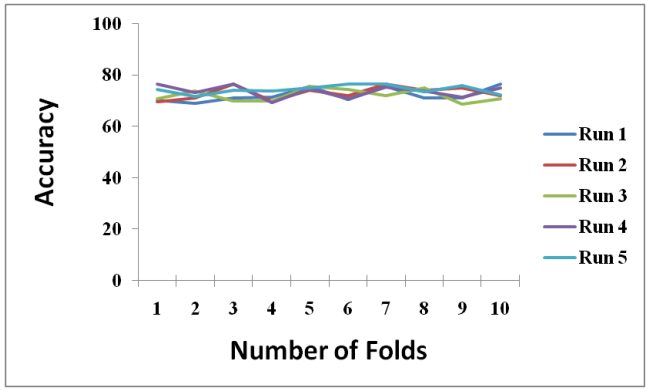

Figure 5.5: Repeated k-fold cross validation of the proposed ensemble model for 10 runs executed 5 times.

outperforms current techniques which is shown in Table 5.4. Accuracy scored by APCpred, ABCpred, BCpred, AAP_{BCPred} and the proposed ensemble model is 67.96%, 66.41%, 65.89%, 64.60% and 69.2% respectively. The proposed ensemble model has boosted the accuracy as well as the other parameters and scores at first rank according to the TOPSIS technique which is shown in Table 5.4. The increased results and TOPSIS ranking suggests that the proposed ensemble model is more accurate and effective than that of the existing techniques.

Table 5.4: Performance comparison of existing and the proposed ensemble model.

Models	**Acc(%)**	**Sen(%)**	**Spe(%)**	**MCC**	**AUC**	**TOPSIS Score**	**Rank**
Proposed Model	**69.2**	**62.82**	**79.6**	**0.375**	**0.692**	**0.84**	**1**
APCpred [36]	67.96	56.15	79	0.362	0.748	0.78	2
ABCpred [5]	66.41	71.66	61.5	0.333	0.736	0.24	4
Bcpred [37]	65.89	66.31	65.5	0.318	0.699	0.27	3
AAP_{BCPred} [38]	64.6	64.17	65	0.292	0.689	0.21	5

5.6 Conclusion

This Chapter contributes in peptide vaccine designs, immunodiagnosis, antibody productions and experimental determination by predicting sequential B-cell epitopes. The proposed ensemble model works well on large dataset and produces improved results for variable length epitopes (6 to 49 mers). Length of the epitopes is important for better performance of the model as well as for prediction of antigenic epitopes. In this Chapter, six models blackboost, avnnet, RF, SVM, GBM and RRF have been used to create an ensemble model by using stack generalized ensemble technique which improves the predictability of the proposed ensemble model. Different parameters like Gini, AUC,

specificity, senstivity and accuracy have been used to evaluate six models individually. The evaluation process is repeated for the proposed ensemble model. TOPSIS a multiple criteria decision making method is used to rank the models on the basis of their evaluation parameters. The benefit of this technique is that the decision of selecting best model is based upon all the five evaluation parameters rather than any one or two parameters. The comparison and ranking by TOPSIS shows that an ensemble model performs better than that of the individual models. For validation, comparison between APCpred, ABCpred, BCpred, AAP_{BCPred} and the proposed ensemble model is performed, which demonstrates that the proposed ensemble model is more efficient. To analyze the robustness of the proposed ensemble model, repeated k-fold cross validation has been performed. It is a crucial task to identify sequential B-cell epitopes. Although, different techniques already exist for the same but the proposed technique is better as shown by comparative analysis.

The proposed ensemble approach can be expanded to perform beneficial role in the different areas of the biology including drug designing, prediction of chronic diseases, prediction of T-cell epitopes, protein structure prediction, allergy and infection predictions and many more. The results may be further enhanced by using emerging machine learning models, optimizing the tuning parameters of the models, extracting more peptides of variable length and adding more physicochemical properties.

5.7 Supplement Data

The dataset used in this Chapter is available at *https://bit.ly/2PeOlvf*. There are three files:

- **"Positive_Negative_epitopes.csv"** contains all the positive and negative epitopes.
- **"Complete_Dataset.csv"** contains complete dataset with all features.
- **"Blind_Dataset.csv"** contains the benchmark dataset with all features.

Chapter 6

Conclusions and Future Work

This Chapter concludes the thesis and also suggests some suggestions towards which the present work can be further extended. Section 6.1 describes the overall conclusions of the research work which has been presented in this thesis. Section 6.2 describes some ideas regarding the future research directions and feasible extension of the present work.

6.1 Conclusions

In this thesis, an effort has been taken to improve the prediction of antigenic epitopes by using machine learning techniques. The major contributions of the thesis are mentioned below:

1. Prediction of the antigenic epitopes can be achieved by using B-cells and T-cells. In this thesis, B-cells, antibody IgG and IgA and T-cells are used separately for the prediction of antigenic epitopes.

2. In machine learning models, the most important phase is data extraction. In this thesis, epitopes and non-epitopes have been collected from the reliable and authenticated resources.

3. The physicochemical properties of peptides have been extracted by using R an open source software and is licensed under GNU GPL.

4. The preprocessing of the data which includes data cleansing, class balancing, feature extraction, feature selection is performed.

5. The feature selection process has been performed by using different techniques which includes RRF, caret and boruta.

6. A multilevel ensemble model is combination of seven different models and is used to predict antibodies IgG and IgA epitopes.

7. To predict T-cell mycobacterium tuberculosis epitopes, an ensemble model is proposed which classifies a peptide is an epitope or a non-epitope rather predicting its binding capacity. While training and testing the models, data partitioning is performed in such a way that all the models are able to access the whole data. Here, the ensemble model is a combination of three different models.

8. Prediction of B-cell epitopes is done by the proposed ensemble model which is combination of six different models.

9. The three different ensemble models are efficient to predict variable and fixed length of epitopes.

10. The consistency of proposed ensemble models predictions has been validated by performing repeated k-fold cross validation.

11. For validation, comparison between existing systems and the proposed ensemble models has been performed, which demonstrates that the proposed ensemble models are more efficient. The existing systems includes IgPred, NetMHC 2.3, NetMHC 4.0, CTLpred, APCpred, ABCpred, Bcpred and AAPBCPred servers.

12. The proposed ensemble models are capable of providing direct prediction by classifying the peptide as an epitope or a non-epitope.

13. The existing tools, prediction of antigenic epitopes is dependent upon single model's outcome but in this thesis, three different ensemble models are proposed. In each ensemble model, different machine learning models are used. Here, boosting and stacked generalization techniques are used to develop the different ensemble models for prediction of antigenic epitopes.

6.2 Future Work

Research follows systematic and iterative approach to analyse and solve the problems. In this thesis, the focus is on the prediction of antigenic epitopes by using ensemble machine learning technique. There are many more areas where the proposed work can be used and expanded as per the requirements. Suggestions for the future work are mentioned below:

1. The proposed ensemble approaches can be expanded to perform beneficial role in the different areas of the biology including drug designing, prediction of chronic diseases, prediction of T-cell epitopes, protein structure prediction, allergy and infection predictions and many more.

2. The proposed data portioning approach can be used in other problems to enhance the predictability of the model.

3. The results may be further enhanced by using emerging machine learning models, optimizing the tuning parameters of the models, extracting more peptides of variable length and adding more physicochemical properties.

References

[1] J.-H. Huang, et al., Using random forest to classify linear B-cell epitopes based on amino acid properties and molecular features, Biochimie, Elsevier 103 (1) (2014) 1–6.

[2] S. Y. Lin, et al., Prediction of B-cell epitopes using evolutionary information and propensity scales, BMC Bioinformatics, Springer 14 (S2) (2013) S10.

[3] B. Yao, et al., SVMTriP: a method to predict antigenic epitopes using support vector machine to integrate tri-peptide similarity and propensity, PloS One, Public Library of Science 7 (9) (2012) e45152.

[4] L. J. Wee, et al., SVM-based prediction of linear B-cell epitopes using bayes feature extraction, BMC Genomics, Springer 11 (4) (2010) 1–9.

[5] S. Saha, et al., Prediction of continuous B−cell epitopes in an antigen using recurrent neural network, Proteins: Structure, Function, and Bioinformatics, Wiley Online Library 65 (1) (2006) 40–48.

[6] A. Abbas, et al., Cellular and molecular immunology, Elsevier Health Sciences (2017).
URL `https://bit.ly/3dzCuW1`

[7] J. Miller, Self-nonself discrimination and tolerance in T and B lymphocytes, Immunologic Besearch, Springer 12 (2) (1993) 115.

[8] B. Alberts, et al., Molecular biology of the cell, sixth edition, Taylor and Francis Group (2014).
URL `https://bit.ly/3j92sAI`

[9] C. A. Janeway, et al., Immunobiology: the immune system in health and disease, New York: Garland Science 6 (2017).
URL `https://bit.ly/2FN7G7G`

[10] A. Maton, Human biology and health, Englewood Cliffs, N.J. : Prentice Hall (1993).
URL `https://bit.ly/3k8NkVl`

[11] P. J. Delves, et al., Roitt's essential immunology, Chichester, West Sussex: Wiley-Blackwell (2011).
URL https://bit.ly/346YMv5

[12] T. Doan, et al., Lippincott's Illustrated Reviews, Immunology, Lippincott Williams and Wilkins (LWW) (2012).
URL https://bit.ly/3kmcU9q

[13] A. K. Abbas, et al., Basic immunology: functions and disorders of the immune system, Elsevier Health Sciences (2014).
URL https://bit.ly/3ocadtx

[14] E. Mix, et al., Immunoglobulins basic considerations, Journal of Neurology, Springer 253 (5) (2006) v9–v17.

[15] J. V. Ravetch, et al., IgG fc receptors, Annual Review of Immunology, Annual Reviews USA 19 (1) (2001) 275–290.

[16] T. G. Dietterich, et al., Ensemble learning, The handbook of brain theory and neural networks, The MIT Press, Cambridge, MA 2 (1) (2002) 110–125.

[17] R. E. Schapire, The boosting approach to machine learning: An overview, Nonlinear Estimation and Classification, Springer (2003) 149–171.

[18] T. G. Dietterich, Ensemble methods in machine learning, International Workshop on Multiple Classifier Systems, Springer (2000) 1–15.

[19] K. C. Lee, et al., Performance of ensemble classifier for location prediction task: emphasis on markov blanket perspective, International Journal of u-and e-Service, Science and Technology 3 (3) (2010) 1–10.

[20] R. Polikar, Ensemble based systems in decision making, IEEE Circuits and Systems Magazine, IEEE 6 (3) (2006) 21–45.

[21] T. G. Dietterich, Machine-learning research, AI Magazine 18 (4) (1997) 97–97.

[22] D. H. Wolpert, et al., No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 67–82.

[23] C. Zhang, et al., Ensemble machine learning: methods and applications, Springer Science and Business Media (2012).
URL https://bit.ly/3dMcEOv

[24] L. Breiman, Bagging predictors, Machine Learning, Springer 24 (2) (1996) 123–140.

[25] R. E. Schapire, et al., Boosting the margin: A new explanation for the effectiveness of voting methods, The Annals of Statistics, Institute of Mathematical Statistics 26 (5) (1998) 1651–1686.

[26] M. Graczyk, et al., Comparison of bagging, boosting and stacking ensembles applied to real estate appraisal, Asian Conference on Intelligent Information and Database Systems, Springer (2010) 340–350.

[27] T. J. Hastie, Generalized additive models, Statistical Models in S, Routledge (2017) 249–307.

[28] E. Roszkowska, Multi-criteria decision making models by applying the TOPSIS method to crisp and interval data, Multiple Criteria Decision Making/University of Economics in Katowice 6 (1) (2011) 200–230.

[29] A. Mardani, et al., Multiple criteria decision-making techniques and their applications–a review of the literature from 2000 to 2014, Economic Research-Ekonomska Istraživanja, Taylor and Francis 28 (1) (2015) 516–571.

[30] S. Gupta, et al., Identification of B-cell epitopes in an antigen for inducing specific class of antibodies, Biology direct, Springer 8 (1) (2013) 27.

[31] M. Nielsen, et al., NN-align. an artificial neural network-based alignment algorithm for MHC class II peptide binding prediction, BMC Bioinformatics, Springer 10 (1) (2009) 296.

[32] K. K. Jensen, et al., Improved methods for predicting peptide binding affinity to MHC class II molecules, Immunology, Wiley Online Library 154 (3) (2018) 394–406.

[33] S. Buus, et al., Sensitive quantitative predictions of peptide-MHC binding by a query by committee artificial neural network approach, Tissue antigens, Wiley Online Library 62 (5) (2003) 378–384.

[34] M. Andreatta, et al., Gapped sequence alignment using artificial neural networks: application to the MHC class I system, Bioinformatics, Oxford University Press 32 (4) (2015) 511–517.

[35] M. Bhasin, et al., Prediction of CTL epitopes using QM, SVM and ANN techniques, Vaccine, Elsevier 22 (23-24) (2004) 3195–3204.

[36] W. Shen, et al., Predicting linear B-cell epitopes using amino acid anchoring pair composition, BioData Mining, BioMed Central 8 (1) (2015) 1–14.

[37] Y. EL-Manzalawy, et al., Predicting linear B-cell epitopes using string kernels, Journal of Molecular Recognition, Wiley Online Library 21 (4) (2008) 243–255.

[38] J. Chen, et al., Prediction of linear B-cell epitopes using amino acid pair antigenicity scale, Amino Acids, Springer 33 (3) (2007) 423–428.

[39] A. M. Living stone, et al., The structure of T-cell epitopes, Annual Review of Immunology, Annual Reviews USA 5 (1) (1987) 477–501.

[40] C. G. Fathman, et al., T-lymphocyte clones, Annual Review of Immunology, Annual Reviews USA 1 (1) (1983) 633–655.

[41] R. H. Schwartz, T-lymphocyte recognition of antigen in association with gene products of the major histocompatibility complex, Annual Review of Immunology, Annual Reviews USA 3 (1) (1985) 237–261.

[42] M. Reth, et al., The B-cell antigen receptor complex, Immunology today, Elsevier Current Trends 12 (6) (1991) 196–201.

[43] H. W. Schroeder Jr, et al., Structure and function of immunoglobulins, Journal of Allergy and Clinical Immunology, Elsevier 125 (2) (2010) S41–S52.

[44] J. M. Woof, et al., Mucosal immunoglobulins, Immunological Reviews, Wiley Online Library 206 (1) (2005) 64–82.

[45] T. W. Chang, et al., Anti-IgE antibodies for the treatment of IgE-mediated allergic diseases, Advances in Immunology, Elsevier 93 (1) (2007) 63–119.

[46] A. Patronov, et al., T-cell epitope vaccine design by immunoinformatics, Open Biology, The Royal Society 3 (1) (2013) 120–139.

[47] G. B. Singh, Introduction to bioinformatics, Fundamentals of Bioinformatics and Computational Biology, Springer (2015) 3–10.

[48] V. Brusic, et al., Bioinformatics tools for identifying T-cell epitopes, Drug Discovery Today: Biosilico, Elsevier 2 (1) (2004) 18–23.

[49] P. Sun, et al., Bioinformatics resources and tools for conformational B-cell epitope prediction, Computational and Mathematical Methods in Medicine, Hindawi 13 (1) (2013) 943636–943636.

[50] J. A. Greenbaum, et al., Towards a consensus on datasets and evaluation metrics for developing B-cell epitope prediction tools, Journal of Molecular Recognition: An Interdisciplinary Journal, Wiley Online Library 20 (2) (2007) 75–82.

[51] B. Yao, Zheng, et al., Conformational B-cell epitope prediction on antigen protein structures: a review of current algorithms and comparison with common binding site prediction methods, PloS One, Public Library of Science 8 (4) (2013) e62249–e62249.

[52] J. Parker, et al., New hydrophilicity scale derived from high-performance liquid chromatography peptide retention data: correlation of predicted surface residues with antigenicity and X-ray-derived accessible sites, Biochemistry, ACS Publications 25 (19) (1986) 5425–5432.

[53] A. Kolaskar, et al., A semi-empirical method for prediction of antigenic determinants on protein antigens, FEBS Letters, Wiley Online Library 276 (1-2) (1990) 172–174.

[54] J. Pellequer, et al., Predicting location of continuous epitopes in proteins from their primary structures., Methods in Enzymology 203 (1) (1990) 176–201.

[55] J.-L. Pellequer, et al., Correlation between the location of antigenic sites and the prediction of turns in proteins, Immunology Letters, Elsevier 36 (1) (1993) 83–99.

[56] A. J. Alix, Predictive estimation of protein linear epitopes by using the program PEOPLE, Vaccine, Elsevier 18 (3) (1999) 311–314.

[57] M. Odorico, et al., BEPITOPE: predicting the location of continuous epitopes and patterns in proteins, Journal of Molecular Recognition, Wiley Online Library 16 (1) (2003) 20–22.

[58] S. Saha, et al., BcePred: prediction of continuous B−cell epitopes in antigenic sequences using physico−chemical properties, Artificial Immune Systems, Springer 3239 (1) (2004) 197–204.

[59] J. E. P. Larsen, et al., Improved method for predicting linear B-cell epitopes, Immunome Research, BioMed Central 2 (1) (2006) 1–7.

[60] M. J. Sweredoski, et al., Cobepro: a novel system for predicting continuous B-cell epitopes, Protein Engineering, Design and Selection, Oxford University Press 22 (3) (2008) 113–120.

[61] J. Gao, et al., BEST: improved prediction of B-cell epitopes from antigen sequences, PloS One, Public Library of Science 7 (6) (2012) e40104.

[62] L. Yao, et al., An Improved Method for Predicting Linear B-cell Epitope Using Deep Maxout Networks, Biomedical and Environmental Sciences, Elsevier 28 (6) (2015) 460–463.

[63] S. Saha, et al., AlgPred: prediction of allergenic proteins and mapping of IgE epitopes, Nucleic Acids Research, Oxford University Press 34 (2006) W202–W209.

[64] M. B. Stadler, et al., Allergenicity prediction by protein sequence, The FASEB Journal, FASEB 17 (9) (2003) 1141–1143.

[65] Å. K. Björklund, et al., Supervised identification of allergen-representative peptides for in silico detection of potentially allergenic proteins, Bioinformatics, Oxford University Press 21 (1) (2005) 39–50.

[66] H. Mohabatkar, et al., Prediction of allergenic proteins by means of the concept of chou's pseudo amino acid composition and a machine learning approach, Medicinal Chemistry, Bentham Science Publishers 9 (1) (2013) 133–137.

[67] J. L. Flynn, Immunology of tuberculosis and implications in vaccine development, Tuberculosis, Elsevier 84 (1) (2004) 93–101.

[68] J. Ferraz, et al., Immune factors and immunoregulation in tuberculosis, Brazilian Journal of Medical and Biological Research, SciELO Brasil 39 (11) (2006) 1387–1397.

[69] Y. Zhao, et al., Application of support vector machines for T-cell epitopes prediction, Bioinformatics, Oxford University Press 19 (15) (2003) 1978–1984.

[70] V. Brusic, et al., Computational methods for prediction of T-cell epitopes a framework for modelling, testing, and applications, Methods, Elsevier 34 (4) (2004) 436–443.

[71] M. Nielsen, et al., Reliable prediction of T-cell epitopes using neural networks with novel sequence representations, Protein Science, Wiley Online Library 12 (5) (2003) 1007–1017.

[72] P. Dönnes, et al., Prediction of MHC class I binding peptides, using SVMHC, BMC Bioinformatics, Springer 3 (1) (2002) 1–8.

[73] W. Fleri, et al., The immune epitope database and analysis resource in epitope discovery and synthetic vaccine design, Frontiers in Immunology, Frontiers 8 (1) (2017) 278.

[74] S. Dhanda, et al., Designing of interferon-gamma inducing MHC class-II binders, Biology Direct, BioMed Central 8 (1) (2013) 30.

[75] C. Vizcaíno, et al., Computational prediction and experimental assessment of secreted/surface proteins from mycobacterium tuberculosis H37Rv, PLoS Comput Biol, Public Library of Science 6 (6) (2010) e1000824.

[76] M. Andreatta, et al., NNAlign: a web-based prediction method allowing non-expert end-user discovery of sequence motifs in quantitative peptide data, PLoS One, Public Library of Science 6 (11) (2011) e26781.

[77] M. Nielsen, et al., NetMHCpan-3.0; improved prediction of binding to MHC class I molecules integrating information from multiple receptor and peptide length datasets, Genome Medicine, BioMed Central 8 (1) (2016) 1–9.

[78] L. Yel, Selective IgA deficiency, Journal of Clinical Immunology, Springer 30 (1) (2010) 10–16.

[79] A. O. Vladutiu, Immunoglobulin D: properties, measurement, and clinical relevance, Clinical and Diagnostic Laboratory Immunology, Am Soc Microbiol 7 (2) (2000) 131–140.

[80] J. L. Fahey, Antibodies and Immunoglobulins: I. Structure and Function, JAMA, American Medical Association 194 (1) (1965) 71–74.

[81] J. F. Ludvigsson, et al., IgA deficiency and risk of cancer: a population-based matched cohort study, Journal of Clinical Immunology, Springer 35 (2) (2015) 182–188.

[82] A. Schlessinger, et al., Epitome: database of structure-inferred antigenic epitopes, Nucleic Acids Research, Oxford University Press 34 (s1) (2006) D777–D780.

[83] D. Osorio, et al., Peptides: A package for data mining of antimicrobial peptides, The R Journal 7 (1) (2015) 4–14.

[84] H. Boman, Antibacterial peptides: basic facts and emerging concepts, Journal of Internal Medicine, Wiley Online Library 254 (3) (2003) 197–215.

[85] H. Hofmann, et al., Evaluation of Diversity in Nucleotide Libraries 1.
URL https://bit.ly/3kbjD67

[86] L. Breiman, et al., Package rrf, CRAN R Project.
URL https://bit.ly/3lRiEbw

[87] A. Liaw, et al., Classification and regression by randomforest, R News 2 (3) (2002) 18–22.

[88] S. S. Keerthi, et al., Convergence of a generalized SMO algorithm for SVM classifier design, Machine Learning, Springer 46 (1-3) (2002) 351–360.

[89] T. Therneau, et al., Package rpart, CRAN R Project.
URL https://bit.ly/33ZIHaF

[90] B. Ripley, et al., Package nnet, R Package Version 7 (2016) 3–12.
URL https://bit.ly/3k1W7ID

[91] A. Gosso, et al., Package elmnn, ELM Package Version 1.
URL https://bit.ly/3k1KykR

[92] M. Kuhn, et al., Package caret, The R Journal.
URL https://bit.ly/3lPzgk5

[93] W. H. Organization, et al., Global tuberculosis report 2016, World Health Organization 1.

[94] P. Shah, et al., In silico design of Mycobacterium tuberculosis epitope ensemble vaccines, Molecular Immunology, Elsevier 97 (1) (2018) 56–62.

[95] T. Hothorn, et al., Package mboost, CRAN R Project.
URL https://bit.ly/33ZfZqa

[96] T. Hastie, et al., Package gam, GAM Package CRAN.
URL https://bit.ly/37a6xT8

[97] A. Karatzoglou, et al., Package kernlab, CRAN R Project.
URL https://bit.ly/2MUgRos

[98] E. Bauer, et al., An empirical comparison of voting classification algorithms: Bagging, boosting, and variants, Machine Learning, Springer 36 (1-2) (1999) 105–139.

[99] A. Geluk, et al., Identification of major epitopes of Mycobacterium tuberculosis AG85B that are recognized by HLA-A* 0201-restricted CD8+ T cells in HLA-transgenic mice and humans, The Journal of Immunology, Am Assoc Immnol 165 (11) (2000) 6463–6471.

[100] J. McMurry, et al., Analyzing Mycobacterium tuberculosis proteomes for candidate vaccine epitopes, Tuberculosis, Elsevier 85 (1) (2005) 95–105.

[101] S. Lata, et al., MHCBN 4.0: a database of MHC/TAP binding peptides and T-cell epitopes, BMC Research Notes, BioMed Central 2 (1) (2009) 61.

[102] M. Nielsen, et al., Improved prediction of MHC class I and class II epitopes using a novel gibbs sampling approach, Bioinformatics, Oxford University Press 20 (9) (2004) 1388–1397.

[103] M. B. Kursa, et al., Feature selection with the boruta package, J Stat Softw 36 (11) (2010) 1–13.

[104] H. Singh, et al., LBtope: Linear B-cell Epitope Prediction Server (2013).
URL http://crdd.osdd.net/raghava/lbtope/data.php

[105] H. Singh, et al., Improved method for linear B-cell epitope prediction using antigen's primary sequence, PloS One, Public Library of Science 8 (5) (2013) e62216.

[106] P. Guo, et al., Mining gene expression data of multiple sclerosis, PloS One, Public Library of Science 9 (6) (2014) e100052.

[107] M. Kursa, et al., Musical instruments in random forest, International Symposium on Methodologies for Intelligent Systems, Springer (2009) 281–290.

[108] L. S. Whitmore, et al., BioCompoundML: a general biofuel property screening tool for biological molecules using Random Forest Classifiers, Energy and Fuels, ACS Publications 30 (10) (2016) 8410–8418.

[109] M. B. Kursa, et al., Package boruta, CRAN R Project 29 (2020) 2015.
URL https://bit.ly/3iXCHDr

[110] L. Breiman, et al., Package randomforest, University of California, Berkeley: Berkeley, CA, USA.
URL https://bit.ly/2lTmsgo

[111] G. Ridgeway, et al., Package gbm, Viitattu, Citeseer 10 (2013) (2013) 40.
URL https://bit.ly/2H0hJqG