Interdisciplinary Sciences: Computational Life Sciences
https://doi.org/10.1007/s12539-018-0309-0

ORIGINAL RESEARCH ARTICLE

Ensemble Technique for Prediction of T-cell *Mycobacterium tuberculosis* Epitopes

Divya Khanna[1] · **Prashant Singh Rana**[1]

Received: 28 June 2018 / Revised: 14 September 2018 / Accepted: 24 October 2018

Abstract
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 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. Identification of T-cell or B-cell epitopes in the targeted antigen is the main goal in designing epitope-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 the present study, 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 (7–40-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.

Keywords T-cell epitopes · *Mycobacterium tuberculosis* · Machine-learning models · Ensemble model · Feature selection

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 [1], approximately 10.4 million TB cases worldwide were found in which 5.9 million were 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 diseases 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 two to three billion TB-infected patients spread TB disease during their lifespan [1]. This issue has boosted the research field for identifying better vaccines and antibodies for curing tuberculosis. According to [2], 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 using advanced computational procedures [2].

The pathogenic Mycobacterium which causes TB survives within the cells. Thus, T cells are required to fight against bacteria rather than antibodies [3, 4]. 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, and immune-diagnostic and antibody production. For prediction of T-cell epitopes with MHC type

✉ Divya Khanna
divya.khanna@thapar.edu

[1] Computer Science and Engineering Department, Thapar Institute of Engineering & Technology, Patiala, Punjab 147004, India

Published online: 07 November 2018

I-restricted T-cell clone, support vector machine (SVM) was used with cross-validation and scored sensitivity 0.763 [5]. Neural network with other methods such as binding motifs, molecular modelling, quantitative matrices and hidden Markov models was used for recognition and prediction of T-cell epitopes and MHC-binding peptides [6]. CTLpred [7] 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 artificial neural network. This scores an accuracy of 70.0%, 72.2% and 75.2%. An improved neural network model [8] 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 [9] (SVMHC) prediction of peptides binding to MHC class I molecules was used which scored Matthews correlation coefficient (MCC) 0.85 with fourfold cross-validation.

On the other hand, for prediction of B-cell epitopes using machine-learning models, methods such as PREDITOP [10], PEOPLE [11], BEPITOPE [12] and BcePred [13], ABCPred [14], Chen et al. [15], BCPred [16], SVMTrip [17], Huang [18], LIAN Yao [19], and APCpred [20] had been proposed. Algpred [21], SVM [22], IgPred [23] and multilevel ensemble model [24] had contributed in predicting specific class of antibodies in an antigen.

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 recognizing 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 the present study, targeted epitopes contain both MHC class I and II T-cell epitopes.

Although many studies exist for predicting binding capacity [25] of *M. tuberculosis* such as interferon-gamma-inducing MHC II binders [26] in which they used SVM for prediction, linear B-cell epitopes were used for prediction of *M. tuberculosis* epitopes [27], NetMHC 2.2 [28] server used artificial neural networks to predict peptide-binding capacity of MHC class II, and NetMHC 2.3 [29] server was an improved version of NetMHC 2.2 server which predicted the binding capacity with HLA class II. This server was constructed 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 [30] server predicted the binding capacity of HLA class I using neural networks, and NetMHC 4.0 [31] overcame the limitation of NNAlign method [32] that it could detect only fixed length motifs. In NetMHC 4.0, artificial neural network 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 [33] server predicted the binding capacity of HLA class I using neural network 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.

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 paper is as follows: the dataset and feature extraction are detailed in Sect. 2. The proposed methodology, feature selection approach, machine-learning models and proposed ensemble model are explained in Sect. 3. Model evaluation on various parameters, repeated k-fold cross-validation, blind dataset and comparison with existing systems are detailed in Sect. 4. Evaluation of Results is narrated in Sect. 5. Discussion is done in Sect. 6. Conclusion and future work are described in Sect. 7.

2 Materials and Methods

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 1.

2.2 Alleles Used in the Present Study

In the present study, epitopes with their corresponding alleles are extracted from IEDB. MHC class I (major histocompatibility complex) alleles in the present study 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,

Table 1 Sample dataset of T-cell *M. tuberculosis*

Sequence	SL[a]	F_a	F_b	F_c	F_d	–	F_z	F_{aa}	F_{ab}	F_{ac}	CL[b]
DMWEHAFYL	9	54.44	0.66	0.36	− 1.91	–	33.33	33.33	11.11	22.22	1
VLMGGVPGVE	10	126.00	− 1.54	0.26	− 1.00	–	10.00	10.00	0.00	10.00	1
STEGNVTGMFA	11	35.45	0.81	0.11	− 1.00	–	45.46	9.09	0.00	9.09	1
SEFAYGSFVRTVSL	14	76.43	0.92	0.40	0.00	–	42.86	14.29	7.14	7.14	1
TDAATLAQEAGNFER	15	52.67	2.57	0.52	− 2.00	–	53.33	26.67	6.67	20.00	1
MTEQQWNFAGIEAAAS	16	49.38	1.03	0.16	− 2.00	–	43.75	12.50	0.00	12.50	1
DAATAQTLQAFLHWAITDGN	20	83.50	0.87	0.39	− 1.91	–	45.00	15.00	5.00	10.00	1
LLAFTNPTV	9	130.00	− 0.77	0.17	0.00	–	33.33	0.00	0.00	0.00	0
GLSTHEGALL	10	127.00	− 0.10	0.08	− 0.91	–	40.00	20.00	10.00	10.00	0
LAQEAGNFERISGDL	15	91.33	1.97	0.51	− 2.00	–	46.67	26.67	6.67	20.00	0
KTDAATLAQEAGNFE	15	52.67	1.94	0.40	− 2.00	–	53.33	26.67	6.67	20.00	0
GEAWTGGGSDKALAAATP	18	49.44	0.53	0.22	− 1.00	–	33.33	16.67	5.56	11.11	0
PAIAAGLNAPRRNRVGRQ	18	81.67	3.08	0.49	4.00	–	38.89	22.22	22.22	0.00	0
MQLVDRVRGAVTGMSRRLVV	20	116.50	2.07	0.51	3.00	–	40.00	25.00	20.00	5.00	0

[a]SL represents sequence length

[b]CL represents class label i.e. 1 means *M. tuberculosis* epitope and 0 means non-*M. tuberculosis* epitope

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 study 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.

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. Table 2 shows the physicochemical properties of an amino acid with brief explanation, related R packages, functions and short names of properties which have been used in the present study. To extract the features, R is used with default parameters of functions. R is an open-source software licensed under GNU GPL.

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 allows them to improve their performances without being programmed. In the present study, ten models are explored as mentioned in Table 3 with required packages and their tuning parameters. To get better results, models may be tuned but in present study default parameters have been used.

The brief detail of models includes RRF, Decision tree and Avnnet which are used in ensemble model given below:

1. *Decision tree model* 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 is used to calculate the entropy:

$$\text{Entropy} = \sum_i -p_i \times \log_2 \times p_i, \tag{1}$$

Table 2 Physicochemical properties of amino acid

Sr. no.	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 [34]	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 [35]	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 neighbors	It describes neighbors of degree one for a group of peptide sequences	Peptider [36]	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
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 using pKa scales	Peptides	charge	F_x
25	Hydrophobicity	It computes the hydrophobicity index of an amino acids sequence	Peptides	hydrophobicity	F_y

Table 2 (continued)

Sr. no.	Property	Description	R Package	Function	Notations used in present study
26	Kidera factor	The Kidera factors are 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 does not carry electrical charge	Peptides	pI	F_{ab}
29	Sequence length	Number of amino acids in a sequence	Peptides	lengthpep	SL

Table 3 Machine-learning models

Model	Method	Required package	Tuning parameter
Regularized Random Forest (RRF) [37]	RRF	RRF	None
Decision Tree [38]	rpart	rpart	parms=list(split="information"), control=rpart.control(usesurrogate=0, maxsurrogate=0)
Avnnet [39]	avNNet	caret	size=10
Blackboost [40]	blackboost	mboost	None
Extreme Learning Machine (ELM) [41]	elmtrain	elmNN	nhid=10, actfun="sig"
Generalized Additive Model (GAM) [42]	gam	gam	None
Gamboost [40]	gamboost	mboost	None
Glmboost [40]	glmboost	glmboost	None
Neural Network [43]	nnet	nnet	size=10
Support Vector Machine (SVM) [44]	ksvm	kernlab	kernel="rbfdot", type="C-svc"

where 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. 2 is used to calculate the information gain of each attribute:

$$\begin{aligned}\text{information gain} &= \text{entropy (entire set)} \\ &\quad - [\text{average (entropy (each split))}].\end{aligned} \tag{2}$$

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 layers, 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 using bootstrapping. In classification problems, the class probabilities are averaged to get the final prediction.
3. *Regularized random forest model* 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 are 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 is 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.

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 Fig. 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.

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 Fig. 2. In methodology of the present study, 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 to

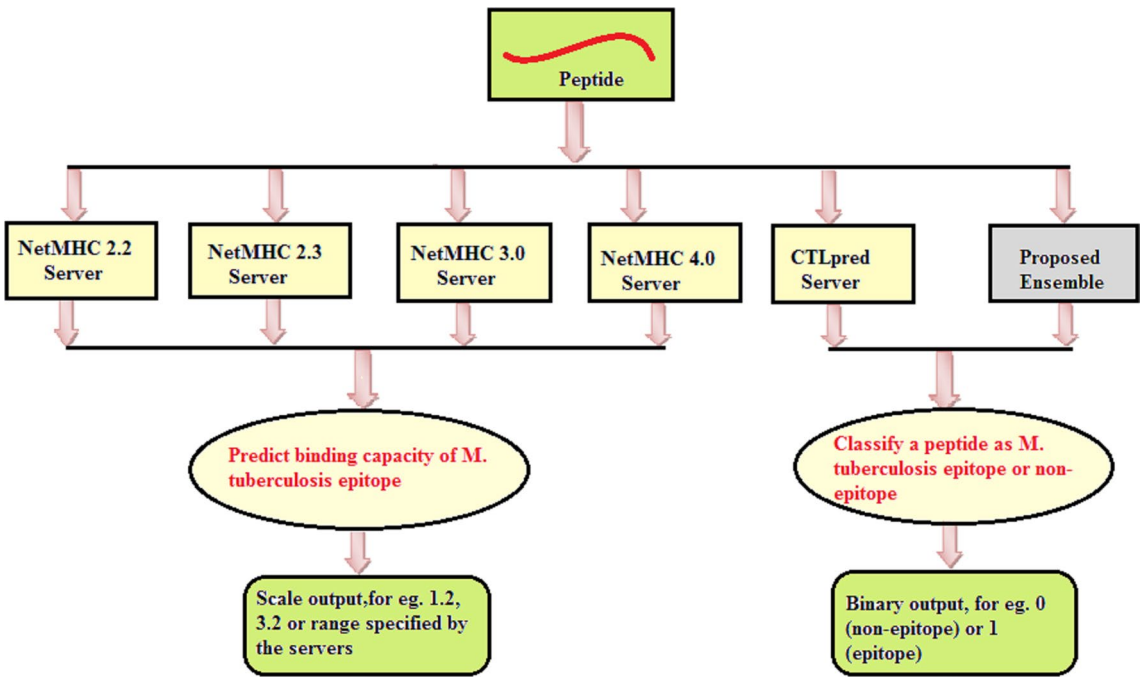

Fig. 1 Prediction outcomes of existing servers and the proposed ensemble model

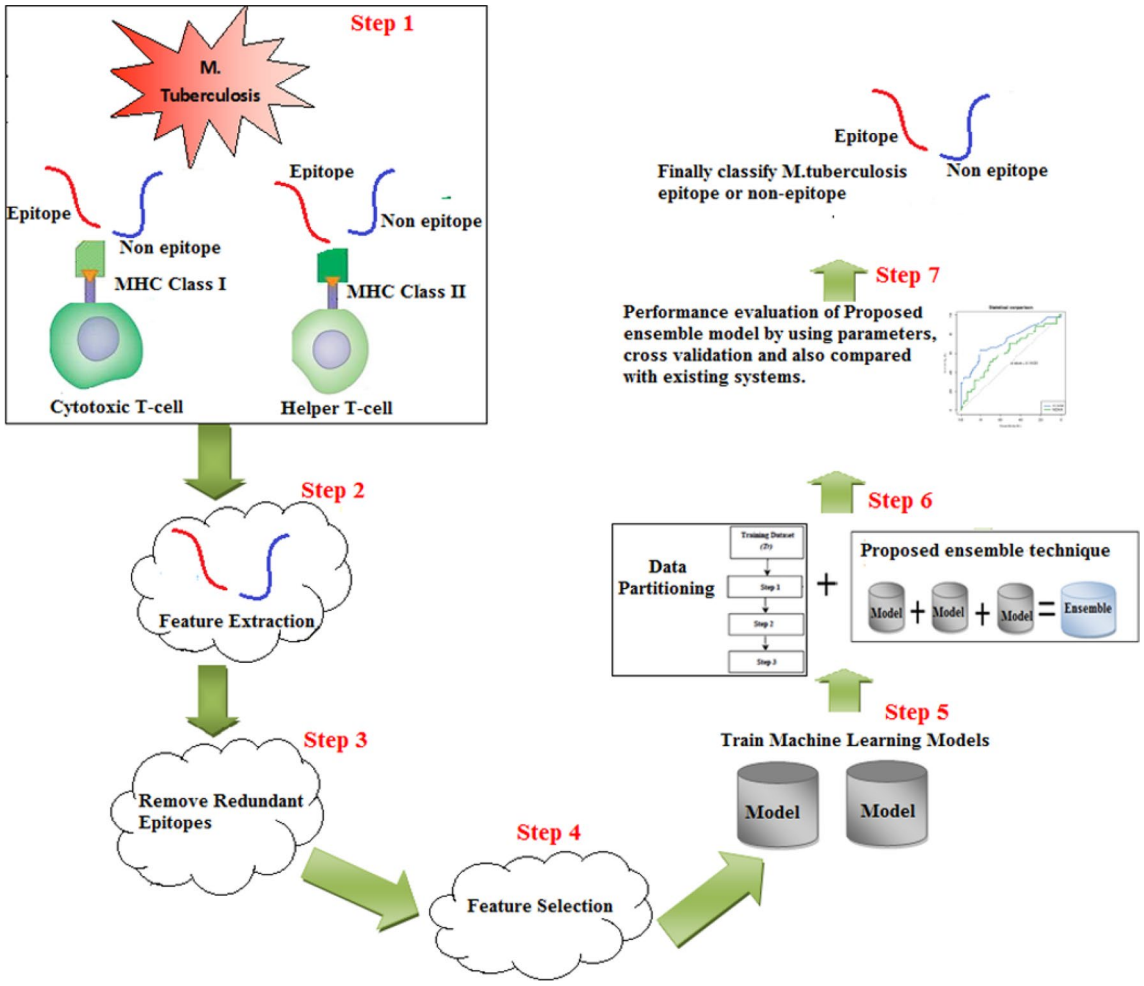

Fig. 2 Graphical view of the proposed work

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 Sect. 2.3.

Step 3 Now, discard redundant epitopes from the dataset.

Step 4 Then importance of each feature is calculated using R package caret. This process enhances the accuracy of prediction as explained in Sect. 3.1.

Step 5 After Step 4, complete dataset is ready which is used to train the models. The models used in the present study are described in Sect. 2.4 and their required R packages, functions and tuning parameters are mentioned in Table 3.

Step 6 In the this step, three models are fused to get proposed model as discussed in Sect. 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, area under the curve (AUC), sensitivity, Gini and accuracy. To analyze the robustness of the model, repeated k-fold cross-validation is used. The blind dataset has been used to validate and compare the proposed ensemble model with existing systems. Finally, results have been analyzed 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.

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 the present study, caret package (available in R) 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. 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 is generated as shown in Table 5. This table shows that the subsets of 10, 15, 20, 25 and 30 features give an accuracy of 81.98%, 80.99%, 80.37%, 82.0% and 78.66%, respectively. According to the evaluation, parameters include accuracy, specificity and sensitivity, and one subset of 25 features (F_a - F_l) is selected. Features F_{ab}, F_t, F_k and F_n are discarded and rest are considered as important ones which are made bold in Table 5.

3.2 Proposed Ensemble Model

Ensemble modelling is a powerful approach to enhance the performance of a given machine-learning model. In the present study, the major goal is to enhance a model's performance by rechecking its false 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.

Table 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			

Table 5 Subset of features and their impact on the performance of proposed ensemble model

Number of features	Features	Accuracy%	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

3.2.1 Data Partitioning to Train the Proposed Ensemble Model

To train the models efficiently, data division process plays an important role. In the present study, during ensembling, data have been circulated to each model in such a way that it improves the overall learning process and hence increases models' predictability. Figure 3 shows the process of data division. Data are divided in such a way that the complete data are 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.

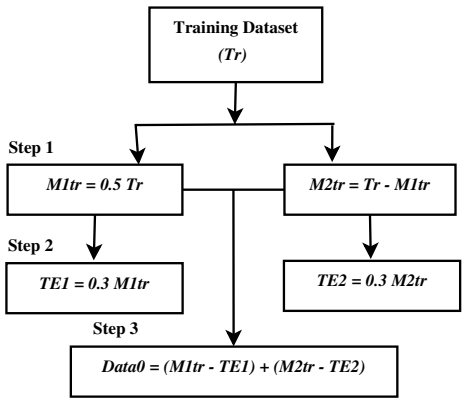

Fig. 3 Data partitioning for the proposed ensemble model

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 3.

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 Fig. 4 to form the proposed ensemble model. Train the model with informative data in such a way that they learn it properly and will be able to produce relevant predictions. Figure 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.

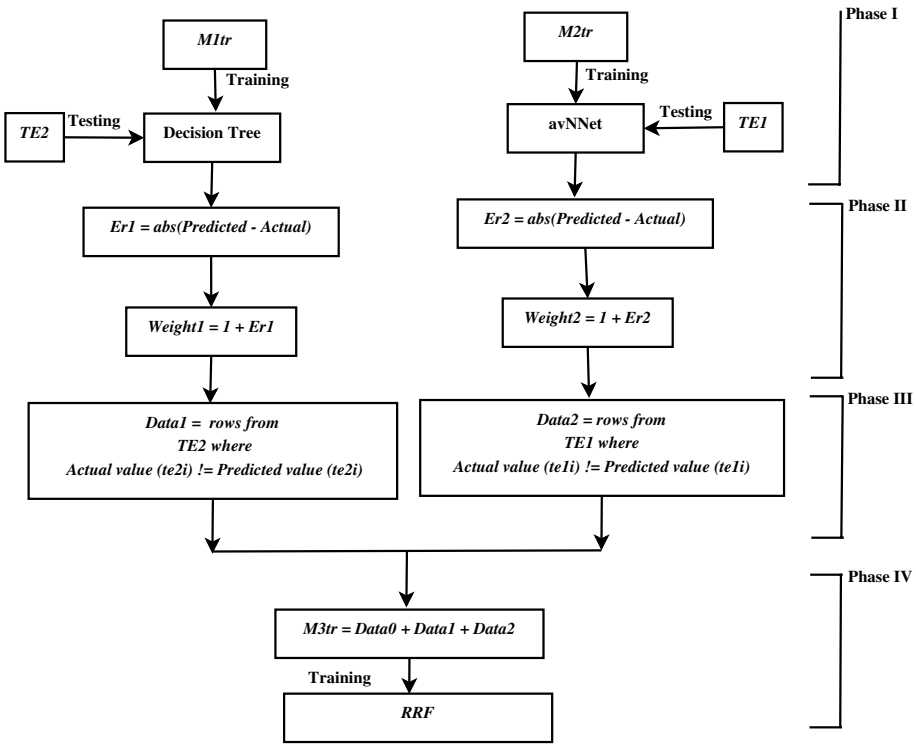

Fig. 4 The proposed ensemble model for prediction of T-cell *M. tuberculosis* epitopes

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.

$$er1_i = \textit{Predicted prob } (te2_i)\textit{-Actual value } (te2_i), \quad i = 1, 2, ..., n \tag{3}$$

$$er2_i = \textit{Predicted proba } (te1_i)\textit{-Actual value } (te1_i), \quad i = 1, 2, ..., n \tag{4}$$

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.

$$weight1_i = er1_i + 1, \quad i = 1, 2, ..., n \tag{5}$$

$$weight2_i = er2_i + 1, \quad i = 1, 2, ..., n \tag{6}$$

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

if(Actual value ($te1_i$)$\neq$*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 [45]. 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 blind datasets, contribution of these three trained models (decision tree, avNNet and RRF) is required and ensembling of their prediction is done using weighted voting.

3.3 Analysis of Improvement in the Results

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

Table 6 Performance evaluation on various parameters of the individual and proposed ensemble models

Sr. no.	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	Neural network	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

1. Prediction from the single model is less reliable 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 using false-predicted instances and data from *M1tr and M2tr*. Therefore, it helps the RRF model to enhance its 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 6.
4. In the present study, 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 Model Evaluation

In model evaluation phase, the proposed ensemble model has been tested using various evaluation parameters, cross-validation and blind dataset. The outcome of this phase concludes that how well the model has learnt the data and how efficiently it produces predictions. Equation 7 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 Sect. 4.1. 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 blind dataset has been used.

After applying feature selection approach on the dataset, four features (F_{ab}, F_t, F_k and F_n) are discarded and rest of the features is considered to train the model as mentioned in Sect. 3.1. The formula for the training process of the three individual models and the proposed ensemble model is:

$$\begin{aligned} CL \sim f(&F_a, F_b, F_c, F_d, F_e, F_f, F_g, F_h, F_i, F_j, F_l, F_m, F_o, \\ &F_p, F_q, F_r, F_s, F_u, F_v, F_w, F_x, F_y, F_z, F_{aa}, F_{ac}, SL). \end{aligned} \tag{7}$$

4.1 Performance Evaluation

To analyze the performance of proposed ensemble model and each individual model, evaluation parameters such as Gini, accuracy, AUC, specificity and sensitivity are used. These parameters are briefly discussed in upcoming section:

4.1.1 Gini Coefficient

Gini coefficient is measured to calculate inequality in the distribution. The Gini 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.

$$\text{Gini} = 2 \times AUC - 1. \tag{8}$$

4.1.2 AUC

To check the quality of the model, AUC (area under curve) is calculated. Basically, area under the receiver operating characteristics (ROC) curve is known as AUC. A model is better than others if model has highest AUC value. Its value lies between 0 and 1. The model has AUC value near to 1 means its quality is good.

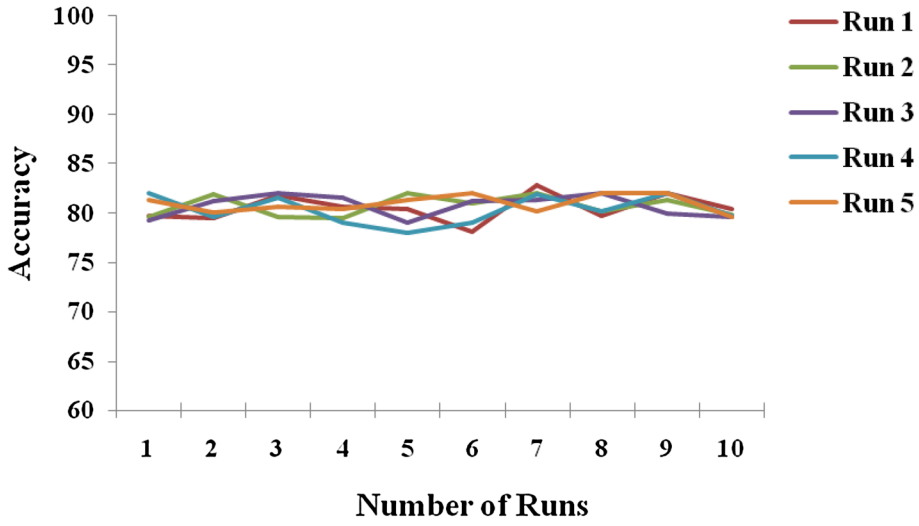

Fig. 5 Repeated tenfold cross-validation of proposed ensemble model

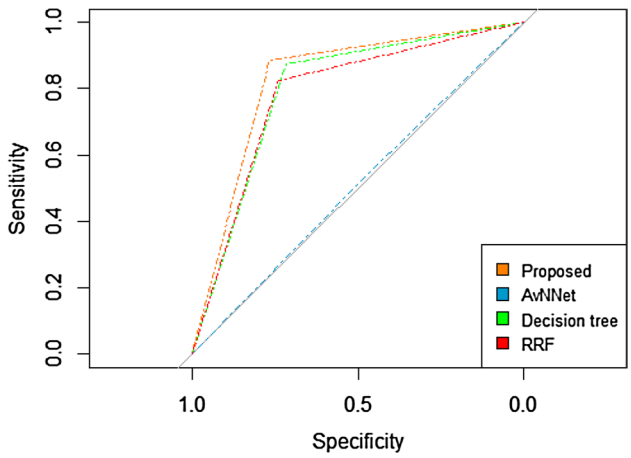

Fig. 6 ROC curves of proposed ensemble and single models

4.1.3 Accuracy

Accuracy tells the correctness of the model predictions and calculated as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Data}} \times 100. \tag{9}$$

4.1.4 Sensitivity

Sensitivity (Sens) is also known as recall or true positive rate. It is the proportion of actual positives which are correctly identified as positives by the model and is computed as follows:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{10}$$

4.1.5 Specificity

Specificity (Spec) is also known as true negative rate. It relates to the models' ability to identify negative results and is computed as follows:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN}} + \text{FP}. \tag{11}$$

TN Non-*M. tuberculosis* epitopes are classified as non-*M. tuberculosis* epitopes *i.e.* true negative.

TP M. tuberculosis epitopes are classified as *M. tuberculosis* epitopes *i.e.* true positive.

FP Non-*M. tuberculosis* epitopes are classified as *M. tuberculosis* epitopes *i.e.* false positive.

FN M. tuberculosis epitopes are classified as non-*M. tuberculosis* epitopes *i.e.* false negative.

AUC AUC (area under curve) is area under the receiver operating characteristics (ROC) curve which is calculated to measure the quality of the model. High AUC value depicts the good quality of the model.

4.2 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 gets increased. In the present

Table 7 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				
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

study, dataset has been divided into tenfolds and each fold has been executed five times.

4.3 Blind Dataset

Another way to validate the model is to test it with the blind dataset. In the blind dataset, 30 epitopes of T-cell *M. tuberculosis* have been collected from literature [46, 47] and MtbVeb server (http://crdd.osdd.net/raghava/mtbveb/). 16 non-epitopes are extracted from MHCBN version 4.0 [48] server. All these epitopes and non-epitopes are unique. These epitopes and non-epitopes are not present in training and testing datasets.

Table 7 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 blind dataset. The outcome of proposed ensemble model suggests that the peptide is *M. tuberculosis* epitope or a non-epitope

Table 8 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-mers epitopes

which is not concluded by the results of NetMHC 2.3 and NetMHC 4.0 servers.

The blind 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 8.

5 Evaluation of Results

The machine-learning models have been trained efficiently on the given dataset as discussed in Sect. 3.2. Since data play an important role in the model training process, therefore, in the present study, data division has been carried out in an efficient way as mentioned in Sect. 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 compared to the individual models.

The results are shown in Table 6 which illustrates 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, five-time tenfold cross-validation and the blind 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 Fig. 5. The average accuracy of the proposed ensemble model comes out to be 80.61% in five-time tenfold cross-validation. Therefore, the proposed ensemble model overcomes the issues such as overfit/underfit/baisness.

The ROC curve plot for the proposed and the individual models is shown in Fig. 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 Fig. 6 that the performance of proposed ensemble model is better than the other three models.

6 Discussion

The proposed ensemble model is compared with the existing systems.

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

Table 7 contains the blind dataset (as mentioned in Sect. 4.3) with predictions from the proposed ensemble model, NetMHC 2.3 [49] and NetMHC 4.0 [49] servers. As shown in table, all the epitopes are correctly predicted by the proposed model and shows its 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 do not 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 overcome by the proposed ensemble model which will classify the peptide as *M. tuberculosis* epitope or a non-epitope.

6.2 Comparison of the Proposed Model with CTLpred Server

The proposed ensemble model is then compared with the CTLpred [7] 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 8. 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 ensemble model is adequate for the prediction of *M. tuberculosis* epitope.

6.3 Other Discussion

In CTLpred [7], 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 artificial neural network. The accuracy scored by artificial neural network is better than quantitative matrix and SVM. The proposed model has contributed 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 Sect. 1) instead of *M. tuberculosis* epitope.

7 Conclusion

In the present study, direct method is used to predict T-cell *M. tuberculosis* epitopes instead of predicting binding capacity of the peptide. Using proposed ensemble model, prediction of T-cell *M. tuberculosis* epitopes is enhanced 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 are circulated to three individual models which train them adequately 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 using more properties of epitopes, different preprocessing techniques and machine-learning models.

References

1. Organization World Health (2016) Global tuberculosis report 2016. WHO. https://bit.ly/2qlTZ4j
2. Shah P, Mistry J, Reche PA, Gatherer D, Flower DR (2018) In silico design of mycobacterium tuberculosis epitope ensemble vaccines. Mol Immunol 97:56–62
3. Ferraz J, Melo F, Albuquerque MdFPM, Montenegro S, Abath F (2006) Immune factors and immunoregulation in tuberculosis. Braz J Med Biol Res 39(11):1387–1397
4. Flynn JL (2004) Immunology of tuberculosis and implications in vaccine development. Tuberculosis 84(1):93–101
5. Zhao Y, Pinilla C, Valmori D, Martin R, Simon R (2003) Application of support vector machines for T-cell epitopes prediction. Bioinformatics 19(15):1978–1984
6. Brusic V, Bajic VB, Petrovsky N (2004) Computational methods for prediction of T-cell epitopes a framework for modelling, testing, and applications. Methods 34(4):436–443
7. Bhasin M, Raghava G (2004) Prediction of CTL epitopes using QM. SVM and ANN techniques. Vaccine 22(23–24):3195–3204
8. Nielsen M, Lundegaard C, Worning P, Lauemøller SL, Lamberth K, Buus S, Brunak S, Lund O (2003) Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. Protein Sci 12(5):1007–1017
9. Dönnes P, Elofsson A (2002) Prediction of MHC class I binding peptides, using SVMHC. BMC Bioinform 3(1):25
10. Pellequer JL, Westhof E, Van Regenmortel MH (1993) Correlation between the location of antigenic sites and the prediction of turns in proteins. Immunol Lett 36(1):83–99
11. Alix AJ (1999) Predictive estimation of protein linear epitopes by using the program PEOPLE. Vaccine 18(3):311–314
12. Odorico M, Pellequer JL (2003) BEPITOPE: predicting the location of continuous epitopes and patterns in proteins. J Mol Recogn 16(1):20–22
13. Saha S, Raghava G (2004) BcePred: prediction of continuous B-cell epitopes in antigenic sequences using physico-chemical properties. In: Nicosia G, Cutello V, Bentley PJ, Timmis J (eds) Artificial immune systems. International conference on artificial immune systems, vol 3239. Springer, Berlin, Heidelberg, pp 197–204
14. Saha S, Raghava G (2006) Prediction of continuous B-cell epitopes in an antigen using recurrent. Neural Netw 65(1):40–48
15. Chen J, Liu H, Yang J, Chou KC (2007) Prediction of linear B-cell epitopes using amino acid pair antigenicity scale. Amino Acids 33(3):423–428
16. EL-Manzalawy Y, Dobbs D, Honavar V (2008) Predicting linear B-cell epitopes using string kernels. J Mol Recogn 21(4):243–255
17. Yao B, Zhang L, Liang S, Zhang C (2012) SVMTriP: a method to predict antigenic epitopes using support vector machine to integrate tri-peptide similarity and propensity. PloS One 7(9):e45152
18. Huang JH, Wen M, Tang LJ, Xie HL, Fu L, Liang YZ, Lu HM (2014) Using random forest to classify linear B-cell epitopes based on amino acid properties and molecular features. Biochimie 103:1–6
19. Yao L, HUANG ZC, Meng G, PAN XM (2015) An improved method for predicting linear B-cell epitope using deep maxout networks. Biomed Environ Sci 28(6):460–463
20. Shen W, Cao Y, Cha L, Zhang X, Ying X, Zhang W, Ge K, Li W, Zhong L (2015) Predicting linear B-cell epitopes using amino acid anchoring pair composition. BioData mining 8(1):1
21. Saha S, Raghava G (2006) AlgPred: prediction of allergenic proteins and mapping of IgE epitopes. Nucleic Acids Res 34(suppl 2):W202–W209
22. Mohabatkar H, Mohammad Beigi M, Abdolahi K, Mohsenzadeh S (2013) Prediction of allergenic proteins by means of the concept of chou's pseudo amino acid composition and a machine learning approach. Med Chem 9(1):133–137
23. Gupta S, Ansari HR, Gautam A, Raghava GP (2013) Identification of B-cell epitopes in an antigen for inducing specific class of antibodies. Biol Direct 8(1):1
24. Khanna D, Rana PS (2017) Multilevel ensemble model for prediction of IgA and IgG antibodies. Immunol Lett 184:51–60
25. Fleri W, Paul S, Dhanda SK, Mahajan S, Xu X, Peters B, Sette A (2017) The immune epitope database and analysis resource in epitope discovery and synthetic vaccine design. Front Immunol 8:278
26. Dhanda SK, Vir P, Raghava GP (2013) Designing of interferon-gamma inducing MHC class-II binders. Biol Direct 8(1):30
27. Vizcaíno C, Restrepo-Montoya D, Rodríguez D, Niño LF, Ocampo M, Vanegas M, Reguero MT, Martínez NL, Patarroyo ME, Patarroyo MA (2010) Computational prediction and experimental assessment of secreted/surface proteins from mycobacterium tuberculosis H37Rv. PLoS Comput Biol 6(6):e1000824
28. Nielsen M, Lund O (2009) NN-align. an artificial neural network-based alignment algorithm for MHC class II peptide binding prediction. BMC Bioinform 10(1):296
29. Jensen KK, Andreatta M, Marcatili P, Buus S, Greenbaum JA, Yan Z, Sette A, Peters B, Nielsen M (2018) Improved methods for predicting peptide binding affinity to MHC class II molecules. Immunology 154(3):394–406
30. Buus S, Lauemøller S, Worning P, Kesmir C, Frimurer T, Corbet S, Fomsgaard A, Hilden J, Holm A, Brunak S (2003) Sensitive quantitative predictions of peptide-mhc binding by a query by committeeartificial neural network approach. Tissue antigens 62(5):378–384

31. Andreatta M, Nielsen M (2015) Gapped sequence alignment using artificial neural networks: application to the MHC class I system. Bioinformatics 32(4):511–517
32. Andreatta M, Schafer-Nielsen C, Lund O, Buus S, Nielsen M (2011) Nnalign: a web-based prediction method allowing non-expert end-user discovery of sequence motifs in quantitative peptide data. PLoS One 6(11):e26781
33. Nielsen M, Andreatta M (2016) NetMHCpan-3.0; improved prediction of binding to MHC class I molecules integrating information from multiple receptor and peptide length datasets. Genome Med 8(1):33
34. Osorio D, Rondon-Villarreal P, Torres R (2015) Peptides: a package for data mining of antimicrobial peptides. R J 7(1):4–14
35. Boman H (2003) Antibacterial peptides: basic facts and emerging concepts. J Intern Med 254(3):197–215
36. Hofmann H, Hare E, GGobi Foundation (2016) Evaluation of diversity in nucleotide libraries, version 0.2.2. https://github.com/heike/peptider
37. RColorBrewer S, Deng H, Deng MH (2018) Package 'RRF', version 1.9. https://sites.google.com/site/houtaodeng/rrf
38. Therneau, T., Atkinson, B., Ripley, B., Ripley, M.B.: Package rpart. https://cran.ma.ic.ac.uk/web/packages/rpart/rpart.pdf. Accessed 20 Apr 2016 (2018)
39. Williams CK, Engelhardt A, Cooper T, Mayer Z, Ziem A, Scrucca L, Tang Y, Candan C, Kuhn MM (2018) Package 'caret', version 6.0-80. https://cran.r-project.org/web/packages/caret/caret.pdf
40. Hothorn T, Buehlmann P, Kneib T, Schmid M, Hofner B, Sobotka F, Scheipl F, Hofner MB (2018) Package 'mboost', version 2.9-1. https://github.com/boost-R/mboost
41. Gosso A, Gosso MA (2012) Package 'elmnn', version 1.0. https://cran.rproject.org/web/packages/elmNN/index.html
42. Hastie T, Hastie MT (2018) Package 'gam', version 1.16. https://cran.r-project.org/web/packages/gam/gam.pdf
43. Ripley B, Venables W, Ripley MB (2016) Package 'nnet', version 7.3-12. https://cran.r-project.org/web/packages/nnet/nnet.pdf
44. Karatzoglou A, Smola A, Hornik K, Karatzoglou MA (2018) Package 'kernlab', version 0.9-27. ftp://tdf.c3sl.ufpr.br/CRAN/web/packages/kernlab/kernlab.pdf
45. Bauer E, Kohavi R (1999) An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Mach Learn 36(1–2):105–139
46. Geluk A, Van Meijgaarden KE, Franken KL, Drijfhout JW, DSouza S, Necker A, Huygen K, Ottenhoff TH (2000) 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. J Immunol 165(11):6463–6471
47. McMurry J, Sbai H, Gennaro M, Carter E, Martin W, De Groot A (2005) Analyzing Mycobacterium tuberculosis proteomes for candidate vaccine epitopes. Tuberculosis 85(1):95–105
48. Lata S, Bhasin M, Raghava GP (2009) MHCBN 4.0: a database of MHC/TAP binding peptides and T-cell epitopes. BMC Res Notes 2(1):61
49. Nielsen M, Lundegaard C, Worning P, Hvid CS, Lamberth K, Buus S, Brunak S, Lund O (2004) Improved prediction of MHC class I and class II epitopes using a novel gibbs sampling approach. Bioinformatics 20(9):1388–1397