A novel fuzzy approach towards *in silico* B-cell epitope identification inducing antigen-specific immune response for Vaccine Design

Aviral Chharia[1], Apurva Narayan[2]
[1]*Mechanical Engineering Department, Thapar Institute of Engineering and Technology, India*
[2]*Department of Computer Science, University of British Columbia, Canada*
achharia_be18@thapar.edu, apurva.narayan@ubc.ca

Abstract

The identification of B-cell epitopes that elicit an antigen-specific immune response is essential for a variety of immunodetection and immunotherapeutic applications, including the development of safe and highly efficacious vaccines. Identifying diagnostically or therapeutically useful epitopes is a difficult, time-consuming, and resource-intensive procedure. *In silico* prediction of B-cell epitope has gained immense attention in recent years due to its low cost, fast results, and less labor-intensive method compared to NMR spectroscopy and 3D X-ray structural analysis of antibody-antigen complexes. However, one of the major problems that most established models confront is gathering huge volumes of data. Moreover, most models do not achieve high levels of accuracy. The current work is the first to propose the 'Fuzzy' approach to *in silico* B-cell epitope prediction. The effectiveness of the proposed approach is demonstrated on severely imbalanced and limited datasets through several experiments. The results show that using the proposed method enhances both accuracy and precision when compared to existing approaches. Further, the model is tested on the SARS-CoV-1 antigen-antibody PDB complex. The proposed approach outperforms state-of-the-art machine learning models trained on the same dataset. Results obtained indicate that applying the proposed method improves the prediction compared to the other approaches.

Index Terms

Fuzzy Classification, Machine Learning, Computational Techniques, B-cell epitope, SARS-CoV

I. Introduction

B-cell epitopes are specific regions on the surface of pathogenic antigens that can be identified by a B-cell receptor or antibody, and serve as a link between infection and the immune system [1]. Only these 'particular' regions on the antigen's surface have been shown to elicit a human immune response in experiments, rather

Sep 20, 2021: This author's *preprint* reports new research that had not yet been certified by peer review and should not be used to guide clinical practice. It is released to ensure the timely dissemination of scholarly work and may differ from the final version, which will undergo peer review and editorial processing.
Dec 15, 2021: The final peer-reviewed and accepted version is now available on IEEE Xplore and is copyrighted by IEEE: https://doi.org/10.1109/BIBE52308.2021.9635292. It was presented at the 21st IEEE International Conference on BioInformatics and BioEngineering, October 25-27, 2021, Kragujevac, Serbia. Please refer to and cite the IEEE version in your research.
©2021 IEEE. Permission from IEEE must be obtained for all uses, in any current or future media.

than the entire antigen. Antibodies released by B-cells move through the bloodstream, neutralizing particular foreign substances. Direct antibody binding generally disables viruses, whereas they attach to the surface bacterial protein in the case of bacteria. One of the major problems in effective vaccine/ antibody design is identifying antigenic areas (in an antigen) that can stimulate B-cell response [1, 2]. Moreover, since epitopes do not represent an intrinsic feature of a protein but rather are relational units defined by binding paratope interaction, it makes the *in silico* prediction of epitopes a challenging task.

NMR spectroscopy and 3D X-ray structural analysis of antibody-antigen complexes, which were initially proposed in 1999, are two conventional techniques for finding B-cell epitope sites. These approaches, although accurate, are resource-consuming and take a long time. The primary difficulty with most approaches is mapping the antigen's 3D structure, which is a relatively complex process. Mimotope-based prediction, for example, is another approach. Computational prediction, on the other hand, is a very beneficial, time-saving, and low-cost option that can assist the laboratory effort. Even though *in silico* B-cell epitope prediction has gotten much attention in recent years, current computational methods for epitope prediction are underused and underestimated due to their low accuracy. Furthermore, to obtain a low false-positive rate, most models require computationally large training datasets, which are difficult to acquire. The task is made more difficult by the limited availability of B-cell epitope data and the high class imbalance present in them.

The current work is the *first* to present a 'Fuzzy' approach to *in silico* B-cell epitope prediction, framing the challenge as an incremental learning problem. We illustrate the model's capacity to train on data that is highly skewed and restricted. The proposed method outperforms state-of-the-art machine learning methods on the same dataset. The significant contributions of this paper include,

1) A 'fuzzy' approach towards the identification of B-cell epitopes is presented. The obtained results are compared to previously developed state-of-the-art machine learning models.
2) In contrast to conventional machine learning models that require vast quantities of epitope data, the model's other uniqueness lies in its ability to achieve better performance in terms of classification accuracy and precision, despite being trained on minimal ($\approx$ 200 to 300) and highly skewed (≈ 0.75) data, as demonstrated experimentally.
3) The model is further validated on the SARS-CoV-1 sequence, on which it surpasses other models (by $\approx 5.77\%$ accuracy) and achieves a higher accuracy for B-cell epitope prediction.

The remaining paper is presented as follows. Section II goes into the specifics of related works. In Section III, the proposed technique is explained. The experiments performed and the results obtained are detailed in Section IV. Section V contains the discussion and limitations. The conclusion and future study directions are presented in Section VI.

II. Related Works

Prediction of B-cell epitopes with high accuracy before laboratory tests can greatly reduce experimental costs while also accelerating the identification process [3]. Several researchers have developed computational models for detecting B-cell epitopes [4-6]. Antigen structure and propensity scales, as well as geometric characteristics and particular physicochemical qualities, are used to identify epitopes in structure-based epitope prediction techniques. A limitation of this technique is that extracting characteristics from 3D antigens rather than the primary sequence is much more challenging.

Mimotope-based epitope prediction is another method that combines mimotope sequences from phage display experiments with 3D antigen structure. By mapping the mimotopes back to the surface of the parent antigen, this method locates the best alignment sequences and predicts possible epitopic regions. In practice, however, this technique is ineffective.

Sequence-based prediction algorithms are another approach. To predict epitope residues, this method uses a feature vector/ matrix produced by scoring each amino acid in an input antigen chain. Based on the primary sequence of antigens, a few techniques, including CoBePro [4], CBTope [5], etc., have been proposed.

Jespersen et al. [6] proposed BepiPred2, a tool to predict linear B-cell epitopes, but attained a low AUC of 0.57.

The recent rise of Deep Learning has triggered a new era in the field. Various models have been developed based on machine learning techniques [7], attention-based LSTM networks [8], deep ensemble learning [9], etc., for B-cell epitope prediction.

III. Proposed Methodology

A. Amino Acid Feature Generation and Encoding

Various antigen structure and amino-acid propensity scales, etc., are used as the elements of the feature vector a_h input to the prediction model. Additional features were created based on the primary sequence features. This includes,

1) Protein Sequence Length - An essential additional feature, formed by mapping the input protein sequence length.
2) Peptide Sequence Length - Created by mapping the length of the input peptide sequence.
3) Parent Protein ID Length - Found by calculating the length of the parent protein-ID.
4) Peptide Length - Calculated by subtracting the start position of the peptide from its end position.

Features, including the parent protein ID, protein sequence, peptide sequence, the start, and the end positions of patches, were dropped from the input feature vector. Antibody valence is the target value to be predicted by the computational model.

B. Fuzzy Classification Model

*1) **Point Hyperbox** $(\mathcal{H})$ **creation**:* The fuzzy min-max neural network is based on hyperbox fuzzy sets [10]. A hyperbox '$\mathcal{H}$' is an n-dimensional geometrical shape [11]. The features of a hyperbox include the hyperbox expansion coefficient, $\theta \in (0, 1)$, which represents the 'hyperbox size'. The network infers the class to which an input feature vector belongs given an input feature vector. Unlike traditional classification models, it can update its learned feature space in real-time with excellent accuracy. After feature normalization, the feature vector a_h for each training sample is passed to the classifier's input nodes, i.e., $(a_1, ..., a_h)$. Here, $a_h =$ (Chou Fasman, Emini, Kolaskar Tongaonkar, Parker, Isoelectric Point, Aromaticity, Hydrophobicity, Stability, Protein Sequence Length, Peptide Sequence Length, Parent Protein ID Length, Peptide Length). In the 12-dimensional feature space, the classifier generates hyperboxes with min coordinate $V_j = (V_{j1}, V_{j2}, ..., V_{jn})$ and max coordinate $W_j = (W_{j1}, W_{j2}, ..., W_{jn})$.

The formed attribute vector is sent to the Classifying Neurons ($\mathcal{CLN}$), which uses min-max hyperboxes to classify the learned samples. In $\mathcal{CLN}$s, b_j represents hyperbox fuzzy set $B_j = A_h, V_j, W_j, f(A_h, V_j, W_j)\ \forall\ (A_h \in I_n)$. In the classifying section nodes, to compute the class memberships, the activation function is used to assign a membership value of 1 when the test sample falls within the hyperbox [12]. When the test sample is outside of $\mathcal{H}$, the model estimates the membership value based on its distance from the $\mathcal{H}$ extreme coordinates. The suggested model and the activation functions of different neurons in the network are depicted in Figure 1.

The input nodes and the hyperbox nodes are linked in the intermediate layer of the classifier. These links reflect the $12-$dimensional hyperbox fuzzy set's min-max coordinates V and W. The neurons in the intermediate layer are dynamically generated during training. The connection between the hyperbox node b_j and class node C_j is represented by the matrix U. The total number of classes learned by the model is represented by the number of output layer nodes in the $\mathcal{CLN}$ section.

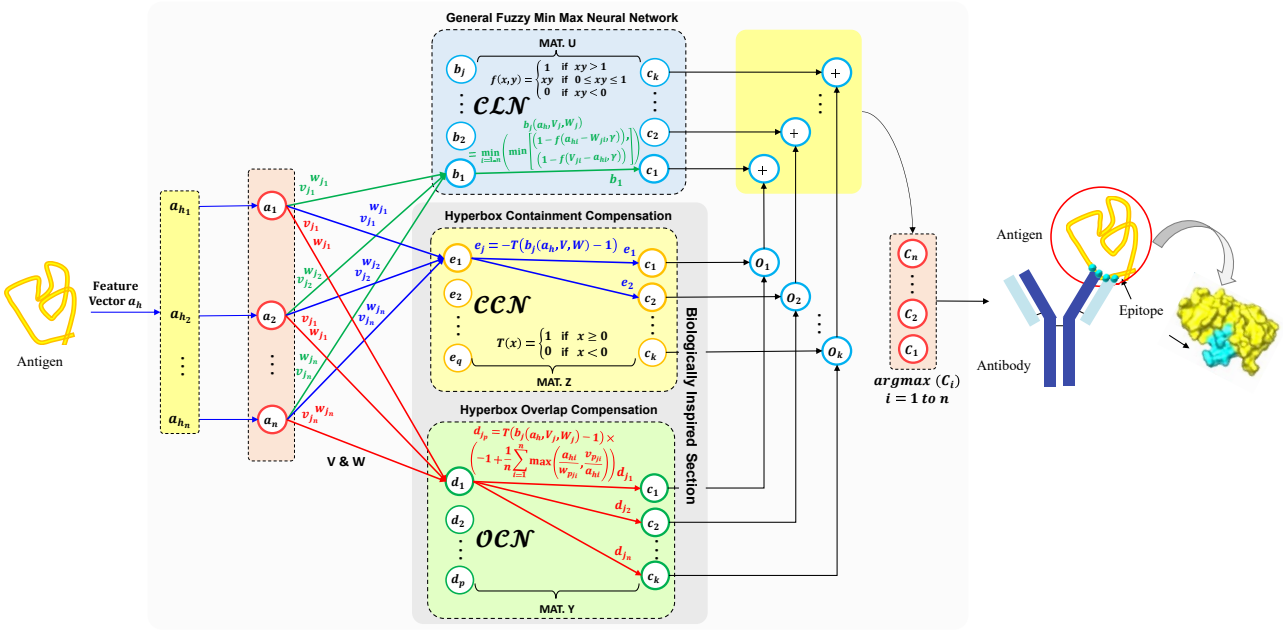


Fig. 1: The proposed architecture of the fuzzy classification model for B-cell epitope identification.

2) ***Model Training****:* A hyperbox node is produced in the $\mathcal{CLN}$ section whenever the model encounters a training sample that does not correspond to the classes it has learnt so far. Subsequently, the model tries to learn on the incoming attribute-class sample $\{a_h, C_i\}$ utilizing past hyperboxes (with the same C_i) using the conditions discussed below, provided the hyperbox size does not exceed a specified maximum limit [10].

$$(1)\ \theta_{\max} \geq \frac{1}{n}\sum_{i=1}^{n}(\max(w_{ji}, a_{hi}) - \min(v_{ji}, a_{hi})) \tag{1}$$

$$(2)\ b_j \text{ is not associated with any Compensation Node} \tag{2}$$

$$(3)\ \text{if } C_i = C_0 \text{ or } C_j = C_0 \text{ then } \mu_j > 0, \text{ where } \mu_j \text{ is membership of hyperbox } b_j, \tag{3}$$

The coordinates of b_j are adjusted, as,

$V_{ji}^{new} = \min(V_{ji}^{old}, a_{hi})$,

$W_{ji}^{new} = \max(W_{ji}^{old}, a_{hi})$, where $i = 1, 2, ..., n$

and if $C_j = C_0$ and $C_i \neq C_0$ then $C_j = C_i$

If it is not possible to expand any of the existing hyperboxes in that class, a new hyperbox is added to the model. Because the high-dimensional feature space contains all of the learnt qualities, hyperbox overlap is a possibility.

3) ***Compensation Nodes for hyperbox error minimization****:* The Reflex section helps in minimizing hyperbox error using compensation neurons [13]. Only when there is an instance of hyperbox overlap and confinement do these neurons become activated. The human brain is cognitively inspired by the reflex system, which automatically gains control of the human body in dangerous situations [13]. The hyperbox node is dynamically formed in the reflex section's intermediate layer if a condition of overlap/ partial or full containment of hyperbox ($\mathcal{H}$) is observed. The $\mathcal{OCN}$ portion is only active if the test data is in the overlap space. It produces two compensation outputs, one for each of the overlapping classes. The $\mathcal{CCN}$ section, on the other hand, overcomes the hyperbox containment issue by activating during an overlap.

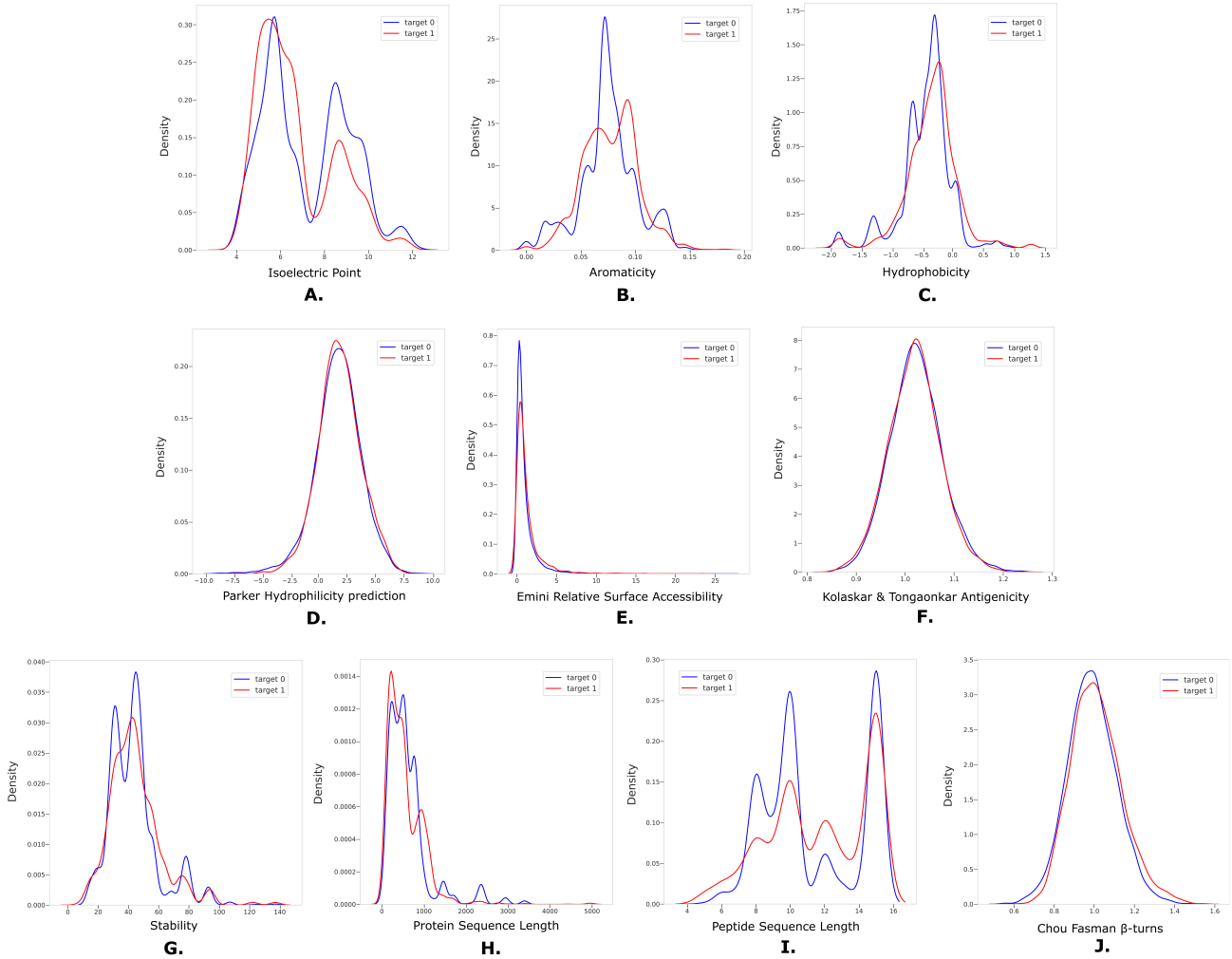


Fig. 2: Feature-wise KDE-plots for epitope targets (in Red) and non-epitope (in Blue) antigen regions. Here, (A) Isoelectric Point, (B) Aromaticity, (C) Hydrophobicity, (D) Parker Hydrophilicity prediction, (E) Emini Relative Surface Accessibility, (F) Kolaskar and Tongaonkar Antigenicity, (G) Stability, (H) Protein Sequence Length, (I) Peptide Sequence Length, (J) Chou Fasman β-turns. A weak negative correlation can be seen.

Hyperbox isolation condition and Containment conditions [10] are used during model training. $\mu_i \leftarrow C_i + O_i$ is the final membership value computation equation, where C_i and O_i are the membership for the i^{th} class.

Hyperbox Overlap Test: The following conditions are utilized for analyzing possible overlap [13]. Initially, we have $\delta^{old} = 1$. If one of the following conditions is true, that implies the existence of an overlap. In that case, $(\delta_{new} - \delta_{old}) > 0$, then, $\Delta = i$ **else** $\Delta = -1$.

C1: $v_{ji} < v_{ki} < w_{ji} < w_{ki} \Rightarrow \delta^{new} = min(w_{ji} - v_{ki}, \delta^{old})$

C2: $v_{ki} < v_{ji} < w_{ki} < w_{ji} \Rightarrow \delta^{new} = min(w_{ki} - v_{ji}, \delta^{old})$

C3: $v_{ji} < v_{ki} \leq w_{ki} < w_{ji} \Rightarrow \delta^{new} = min(min(w_{ki} - v_{ji}, w_{ji} - v_{ki}), \delta^{old})$

C4: $v_{ki} < v_{ji} \leq w_{ji} < w_{ki} \Rightarrow \delta^{new} = min(min(w_{ki} - v_{ji}, w_{ji} - v_{ki}), \delta^{old})$

Hyperbox Contraction Test: The hyperboxes are contracted using the following provided criteria if overlap occurs and is minimal along the Δ dimension [13]:

C1: $v_{j\Delta} < v_{k\Delta} < w_{j\Delta} < w_{k\Delta} v_{k\Delta}^{new} = w_{j\Delta}^{new} = \frac{w_{j\Delta}^{old} + v_{k\Delta}^{old}}{2}$

C2: $v_{k\Delta} < v_{j\Delta} < w_{k\Delta} < w_{j\Delta} v_{k\Delta}^{new} = w_{j\Delta}^{new} = \frac{w_{k\Delta}^{old} + v_{j\Delta}^{old}}{2}$
C3: $v_{k\Delta} < v_{j\Delta} \leq w_{j\Delta} < w_{k\Delta}$ and $w_{k\Delta} - v_{j\Delta} < w_{j\Delta} - v_{k\Delta}$, then $v_{j\Delta}^{new} = w_{k\Delta}^{old}$ else $w_{j\Delta}^{new} = v_{k\Delta}^{old}$
C4: $v_{j\Delta} < v_{k\Delta} \leq w_{k\Delta} < w_{j\Delta}$ and $w_{k\Delta} - v_{j\Delta} < w_{j\Delta} - v_{k\Delta}$, then $w_{j\Delta}^{new} = v_{k\Delta}^{old}$ else $v_{j\Delta}^{new} = w_{k\Delta}^{old}$

IV. Experiments and Results

A. Experimental Settings

*1) **Dataset**:* The epitope candidate amino acid sequence (peptides) and the activity label data from the B-cell epitope data were used from IEDB [14] and UniProt [15]. The presented antibody proteins were IgG. To minimize the high-class imbalance, the epitope data were converted to binary classification. For this study, 'Positive-High,' 'Positive-Intermediate,' and 'Positive Low' samples were all regarded as 'Positive' samples. Nonetheless, high-class imbalance makes it harder to get positive samples compared to negative ones, making many deep learning models inefficient to train. The proposed model was trained using the B-cell data subset, which consisted of only 200 and 300 data samples with information on whether or not an amino acid peptide had antibody-inducing activity, which was indicated by the activity label.

TABLE I: Comparison of B-cell epitope prediction results with other machine learning classifiers for $n = 200$, Imb$= 0.75$. Here, learning parameters $\theta = 0.25$, $\gamma = 2$ for the proposed model

Classification Model	Data Subset Configuration- I					
	Accuracy (↑)	Precision (↑)	Recall (↑)	F1-Score (↑)	MCC (↑)	TT (*sec*) (↓)
Ada Boost Classifier	66.25	26.50	21.50	23.33	0.0376	0.364
Support Vector Machine	66.88	38.42	37.00	35.31	0.1728	0.013
Logistic Regression	68.12	29.33	16.00	19.13	0.0587	0.022
Decision Tree	68.12	41.90	35.00	34.23	0.1710	**0.012**
Ridge Classifier	68.12	15.00	14.00	14.44	0.0226	0.013
Gradient Boosting Classifier	69.38	46.67	29.00	35.44	0.1776	0.083
K-Neighbors Classifier	70.62	45.83	26.50	32.90	0.1842	0.063
Random Forest	72.50	46.67	21.00	27.19	0.1966	0.412
Proposed Method	**72.50**	**90.00**	**47.37**	**62.07**	**0.4914**	0.614

TABLE II: Comparison of B-cell epitope prediction results with other machine learning classifiers for $n = 300$, Imb$= 0.75$. Here learning parameters $\theta = 0.20$, $\gamma = 2$ for the proposed model

Classification Model	Data Subset Configuration- II					
	Accuracy (↑)	Precision (↑)	Recall (↑)	F1-Score (↑)	MCC (↑)	TT (*sec*) (↓)
Support Vector Machine	65.42	36.87	30.48	31.68	0.0993	0.015
K-Neighbors Classifier	69.58	34.50	12.62	17.58	0.0619	0.062
Decision Tree	71.67	48.09	**46.19**	**45.98**	0.2780	**0.013**
Ada Boost Classifier	73.75	53.20	41.19	45.56	0.2962	0.092
Random Forest	74.17	50.17	22.38	30.12	0.2031	0.411
Logistic Regression	74.17	49.83	22.38	29.78	0.2089	0.023
Gradient Boosting Classifier	74.58	55.17	33.57	40.79	0.2777	0.098
Ridge Classifier	**75.42**	57.50	28.57	36.87	0.2792	**0.013**
Proposed Method	75.00	**80.00**	22.22	34.78	**0.3289**	1.814

*2) **Evaluation Metrics**:* The classification performance of various models is evaluated using confusion matrix-based metrics, which include Accuracy, precision, recall, F1-score, and Matthews Correlation Coefficient (MCC).

- **Accuracy** - It calculates the percentage of correctly identified illnesses in comparison to the total test dataset. A model's performance improves as its accuracy improves. It lies between [0,1] and is reported

generally as a percentage (%). When the sample distribution in multiple classes is a balanced one, accuracy serves as a suitable metric for the test dataset. It is calculated as:

$$Accuracy = \frac{TN + TP}{TN + FN + TP + FP} \tag{4}$$

- **Precision** - It estimates the ratio of a particular epitope that has been correctly classified to the total number of that particular epitope identified by the model. A model is better if the Precision is greater. Precision, like Accuracy, falls between [0,1] and is usually expressed as a percentage (%). Precision is calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

- **Recall** - It is the measure of a model correctly identifying True Positives. Thus, for all the epitopes that are actually positive, recall tells how many sequences were correctly identified as positive B-cell epitopes. Precision and recall are both equal to 1 in a perfect classifier. Precision and recall are not particularly useful metrics when used in isolation. Recall is calculated as:

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

- **F1-Score** - It is defined as the harmonic mean of Precision and Recall. This is also known as the F1-Measure. If the F1-score is higher, a model is better. F1-Score [0,1] is generally reported as a percentage (%) and is calculated as:

$$F1 - Score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right) \tag{7}$$

- **Matthews Correlation Coefficient (MCC)** - It accounts for true and false positives and negatives, and is widely considered a fair metric that can be applied even when the classes are of varying sizes. The MCC is essentially a (-1, +1) correlation coefficient number. A perfect forecast has a coefficient of +1, an average random prediction has a coefficient of 0, and an inverse prediction has a coefficient of -1. It is calculated as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{8}$$

where TP, TN, FP, and FN are the number of true positives, true negatives, false positives, and false negatives, respectively.

*3) **Implementation Details**:* The studies were carried out on an NVIDIA K80 GPU workstation with 12GB RAM and TensorFlow as a backend. The normalisation method was 'zscore', computed as $z = (x - u)/s$. When training machine learning classifiers for comparison, the 'yeo-johnson' transformation was used. The data was divided into a typical 80-20% train-test set.

B. Experiment-1: Prediction Performance and Comparison on Limited and High Imbalance Data Configuration

To show the efficacy of the suggested technique, a set of three experiments is carried out. In both experiments, the model was trained on datasets that represent the real-world scenario. This section discusses the experiments and the results of the model's ability to predict B-Cell Epitope prediction on two data subset configurations. We also discuss the hyperparameters chosen during these experiments.

In the first experiment, the model was trained on a limited data subset configuration with $n = 200$. Here, 'n' represents the number of samples taken. The dataset contained a high-class imbalance of 0.75, and the model's classification ability was tested. The experiment was repeated for another subset configuration with the layer containing $n = 300$ samples and 0.75 class imbalance to establish the generalizability of the model's ability. State-of-the-art machine learning models were trained on the same dataset, and their

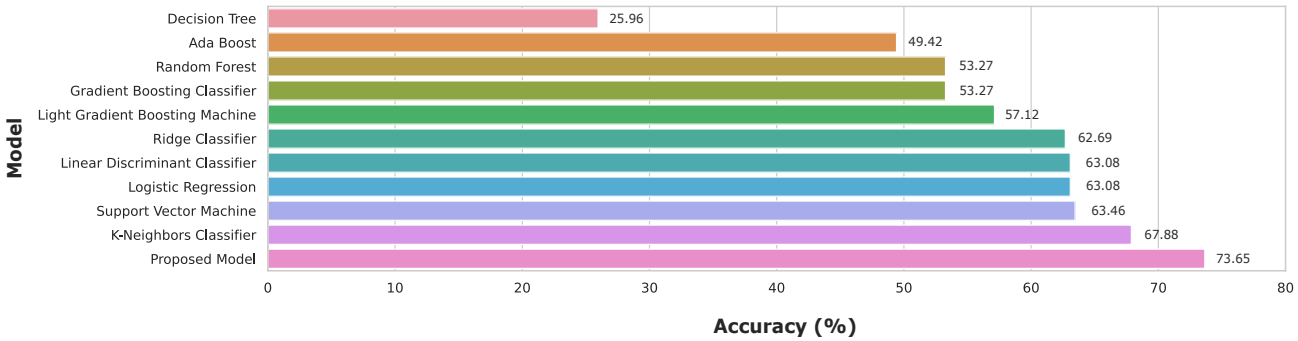


Fig. 3: Comparison bar-plot between various models and the obtained classification results on the SARS-CoV-1 antigen sequence

results were compared with those obtained by the proposed method. Tables I and II show the prediction performance and compare the obtained results with machine learning models for data subset configurations I and II, respectively.

The suggested model outperforms all other models by a significant margin in both experiments, i.e., with $n = 200$ and 300. In subset configuration- I, it can be seen that the proposed model obtained comparable accuracy as Random Forest, which is the second-best performing model in the configuration, but obtains a far larger precision, Recall, and a high F1-score. On data with high-class imbalance, using accuracy alone as an evaluation metric may sometimes be misleading. Therefore, accuracy, along with precision, serves as a better metric for assessing model performance. From Table I, it can be inferred that though various machine learning models obtain accuracy comparable to the proposed approach, when seen in unison with precision and recall, a significant difference is evident.

Similar results are obtained on subset configuration- II, where the proposed approach maintained a high precision along with the epitope prediction accuracy. Furthermore, it is seen that as the number of training samples varies, the second-best performing model loses consistency, i.e., for $n = 200$, Random Forest is the second-best-performing model, its performance decreases for $n = 300$, where the Ridge classifier and Gradient Boosting Classifier are seen performing better. This illustrates that no single machine learning model exhibits robust and consistent performance while training on limited data. On the other hand, the fuzzy model outperforms machine learning models since it's consistent across all subset configurations with restricted data. This indicates the model's remarkable capacity to perform well even with small and imbalanced data.

C. Experiment-2: Prediction performance and comparison on complete SARS-CoV-1 Sequence

In contrast to the previous experiment, in which the model's classification performance was assessed on a small and severely unbalanced dataset, we analyze the model on the entire SARS-CoV-1 antigen sequence in this section. The model was tested to predict the B-cell epitope regions on the SARS-CoV-1 sequence. It should be noted that while the model was trained using IEDB [14] and Uniprot [15] data, the antigen sequence was previously unknown to the model. This translates to real-time circumstances involving the prediction of B-cell epitopes of new antigen sequences.

V. Discussion and Limitations

To assess the influence of various model parameters, an in-depth parametric study was undertaken in which the effect of varying hyperbox expansion coefficient (θ) and fuzziness control parameter (γ) on the number of hyperboxes ($\mathcal{H}$) produced during training, and the model training time (sec) was investigated. In addition to the parametric research, a temporal complexity analysis is performed. The sample testing time

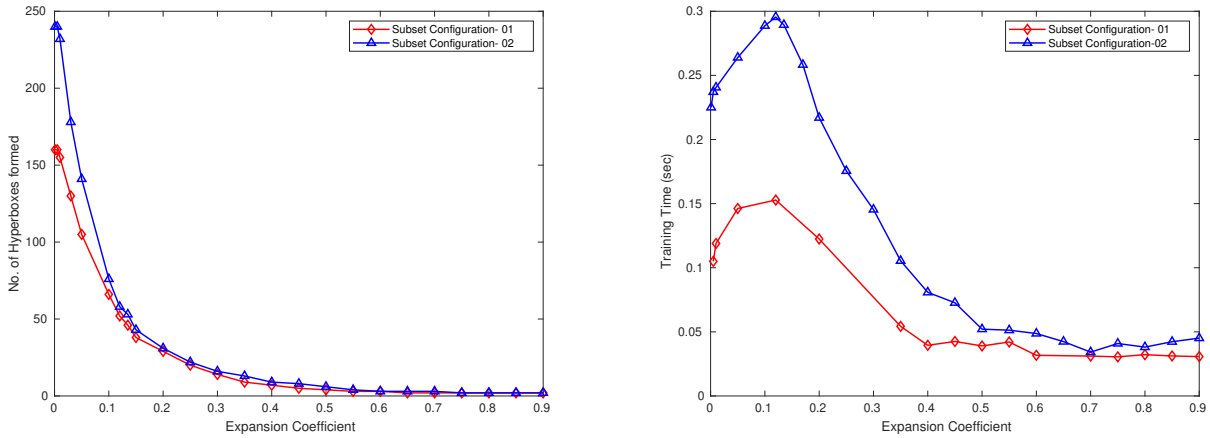


Fig. 4: Parametric study results (a) No. of hyperboxes formed vs. expansion coefficient (θ) (b) Training time (sec) vs. expansion coefficient (θ).

(sec) for the B-cell epitope prediction task is plotted graphically by adjusting the hyperparameters. The experiment is performed several times, varying the hyperparameters to determine the total model training time. The obtained study indicates that, while the model's overall training time is extremely short (i.e., 1 to 5 sec), the sample test time is rather long, ranging from 5 to 10 sec. Such a substantial gap is not found in low-dimensional data categorization tasks. Moreover, as the hyperbox expansion coefficient (θ) increases, the number of hyperboxes created during training increases in an exponential rather than linear manner. Figure 3(a) depicts this graphically. On the other hand, as shown in Figure 3(b), the model training time climbs steeply until 0.2, after which it reduces exponentially on both training data configurations.

VI. Conclusion and Future Work

In this paper, we present a novel fuzzy approach towards identifying B-cell epitopes that trigger an antigen-specific immune response. Experimental results show that the model outperforms state-of-the-art machine learning models trained on the same dataset. The obtained results also establish that an appropriate combination of attributes as epitope features could enhance the prediction precision of B-cell epitopes.

The present study has certain limitations. First, the suggested methodology necessitates a significant amount of time to determine if a sequence is positive/ negative for a B-cell epitope. Comparatively, the total model training time ($1-5$ sec) is less than the sample test time ($5-10$ sec). Future studies might focus on reducing the model's lengthy sample testing time. Second, without the need for operator interaction, an algorithm for tuning the hyperbox expansion coefficient (θ) and fuzziness coefficient (γ) might be created. By improving the model architecture, we expect to increase the model's sensitivity to predicting epitopes in the future.

Acknowledgments

This work was supported in part by the MITACS Globalink Research Award to A.C. to pursue a Research project at the University of British Columbia as a Visiting International Research Student.

References

[1] T. A. Ahmad, A. E. Eweida, and S. A. Sheweita, "B-cell epitope mapping for the design of vaccines and effective diagnostics," Trials Vaccinol., vol. 5, pp. 71–83, 2016.

[2] S. E. C. Caoili, "Benchmarking B-cell epitope prediction for the design of peptide-based vaccines: problems and prospects," J. Biomed. Biotechnol., vol. 2010, p. 910524, 2010.
[3] D. J. Barlow, M. S. Edwards, and J. M. Thornton, "Continuous and discontinuous protein antigenic determinants," Nature, vol. 322, no. 6081, pp. 747–748, 1986.
[4] M. J. Sweredoski and P. Baldi, "COBEpro: a novel system for predicting continuous B-cell epitopes," Protein Eng. Des. Sel., vol. 22, no. 3, pp. 113–120, 2009.
[5] H. R. Ansari and G. P. Raghava, "Identification of conformational B-cell Epitopes in an antigen from its primary sequence," Immunome Res., vol. 6, no. 1, p. 6, 2010.
[6] M. C. Jespersen, B. Peters, M. Nielsen, and P. Marcatili, "BepiPred-2.0: improving sequence-based B-cell epitope prediction using conformational epitopes," Nucleic Acids Res., vol. 45, no. W1, pp. W24–W29, 2017.
[7] K. V. Kavitha, R. Saritha, and C. S. S. Vinod, "Computational prediction of continuous B-cell epitopes using random forest classifier," in 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp. 1–5.
[8] T. Noumi et al., "Epitope prediction of antigen protein using attention-based LSTM network," bioRxiv, 2020.
[9] P. Sun, Y. Yu, R. Wang, M. Cheng, Z. Zhou, and H. Sun, "B-cell Epitope prediction method based on deep ensemble architecture and sequences," in 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2019, pp. 94–97.
[10] P. K. Simpson, "Fuzzy min-max neural networks. I. Classification," IEEE Trans. Neural Netw., vol. 3, no. 5, pp. 776–786, 1992.
[11] B. Alpern and L. Carter, "The hyperbox," in Proceeding Visualization '91, 2002, pp. 133–139.
[12] B. Gabrys and A. Bargiela, "General fuzzy min-max neural network for clustering and classification," IEEE Trans. Neural Netw., vol. 11, no. 3, pp. 769–783, 2000.
[13] A. V. Nandedkar and P. K. Biswas, "A fuzzy min-max neural network classifier with compensatory neuron architecture," IEEE Trans. Neural Netw., vol. 18, no. 1, pp. 42–54, 2007.
[14] R. Vita et al., "The Immune Epitope Database (IEDB): 2018 update," Nucleic Acids Res., vol. 47, no. D1, pp. D339–D343, 2019.
[15] UniProt Consortium, "UniProt: the universal protein knowledgebase in 2021," Nucleic Acids Res., vol. 49, no. D1, pp. D480–D489, 2021.