Classification And Prediction Of Epitopes Using Machine Learning Algorithms

Pradeep Kumar H S[1], Amrutha C Bhat[2], Anagha A S[3], Isha N[4], Khushi M S[5]

[1,2,3,4,5] *The National Institute of Engineering, Mysuru-570008, Karnataka, India*

***

Abstract* - Epitope classification stands as a cornerstone in vaccine development. The accurate classification of epitopes and non-epitopes significantly influences vaccine effectiveness. This project employs feature engineering techniques to raise the quality of input data. Various Machine learning algorithms, such as Support Vector Machine, k-Nearest Neighbours, Logistic Regression, Random Forest and XGBoost alongside deep learning algorithms like Convolutional Neural Networkundergo rigorous evaluation and comparison to identify the most effective methods for precise epitope prediction. The exploration further extends to the integration of transfer learning methodologies, leveraging pre-existing knowledge to enhance epitope classification performance. The primary objective is to assess the precision of epitope classification, a critical aspect in immunology and vaccine development. Clearly separating epitopes from non- epitopes is crucial in designing vaccines. This ensures the vaccine prompts the right immune reactions while minimising unwanted responses. Utilising feature engineering techniques and systematic algorithm evaluation, the project strives to optimise the accuracy of epitope classification.

***Keywords* - Machine Learning; B-cell Epitopes; Immunology; Feature Engineering; Vaccine Development**

1. INTRODUCTION

Immunoinformatics is a swiftly advancing field, pivotal to vaccine development and immune response. At its heart are B-cell epitopes (BCEs), which are precise sections on antigens, the proteins showcased by pathogens, prompting the creation of antibodies, the immune system's defenders [9]. Understanding BCEs is paramount for designing effective vaccines, antibodies, and immunotherapies [9]. Traditional experimental methods for identifying BCEs are laborious and time-consuming [10]. This research addresses this gap by proposing a novel approach to BCE classification using machine learning (ML) algorithms.

BCEs can be classified into two main types: linear, where binding occurs on a continuous surface chain, and conformational, involving folded protein chains with discontinuous amino acids [10]. Understanding linear BCEs is crucial for pinpointing the exact regions that trigger the immune response [10]. This knowledge is instrumental in designing targeted vaccines that stimulate a robust immune response against specific pathogens [10]. Furthermore, BCE classification can offer insights into autoimmune diseases and allergies, aiding in the development of therapeutic interventions [10].

In silico prediction of linear B-cell epitopes (BCEs) has evolved from rudimentary sequence-based methods to more advanced Machine Learning (ML) techniques, yet substantial challenges remain [11]. Earlier models primarily focused on compositional properties and physicochemical characteristics of proteins, including antigenicity, torsion, and surface accessibility [12]. Although some, like PREDITOP [13], BcePred [14], BEPITOPE [15], and PEOPLE [16], achieved seemingly high performance, later studies revealed overestimations in their predictive accuracy [17, 11]. The availability of expanding proteomic data has led to the application of various ML techniques for BCE prediction, aiming to address limitations of prior methods. BepiPred 2.0, for instance, achieved an Area Under the Curve (AUC) of 0.671 through a Hidden Markov Model (HMM) incorporating secondary structure and hydrophilicity propensity scales [18]. ABCpred, established in 2006, utilises recurrent neural networks (RNNs) and achieves an accuracy (ACC) of 0.66 and a Matthews Correlation Coefficient (MCC) of 0.319 with a sliding window size of 16 [19]. However, it relies on a combination of biochemical and physicochemical features that may not fully capture the complexities of BCEs.

Support Vector Machines (SVMs) have also been implemented, like SVMTriP, which integrates tri-peptide composition and propensity scales to forecast linear antigenic B-cell epitopes, achieving a precision (Prec) of 55.20% and an AUC value of 0.702 [20]. LBtope utilises a wider range of primary sequence-based features and achieves an accuracy range of 58.39% to 66.7% and AUC values ranging from 0.60 to 0.73 [21]. Deep learning approaches have recently shown significant promise. NetBCE, a deep learning framework, outperforms conventional methods by a substantial margin, achieving an AUC of 0.8400. It attributes this success to its use of feature analysis, encoding, and a ten-layer architecture with CNN, BLSTM, and attention mechanisms [22]. SEMA, another recent development, focuses on antigen B-cell conformational epitope prediction using deep transfer learning. It achieves an ROC AUC of 0.76, demonstrating effectiveness in antibody-antigen interaction prediction and epitope residue distinction [23]. While these advancements are noteworthy, accurately predicting linear B-cell epitopes using computational methods remains challenging. Our study aims to address this gap by incorporating methodologies that specifically target the complexities associated with linear B-cell epitope prediction. This study investigates the effectiveness of various machine learning algorithms for B-cell epitope classification. We conduct a comparative analysis of Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Convolutional Neural Networks (CNN), Logistic Regression, Random Forest and XGBoost. Our approach emphasises feature engineering to improve model interpretability. By extracting and selecting relevant features from the epitope data, we aim to uncover the underlying biological mechanisms that influence epitope classification. Additionally, we explore transfer learning by evaluating pre-trained models for their suitability in B-cell epitope prediction. This comprehensive analysis, utilising metrics like accuracy will identify the most effective algorithms for B-cell epitope classification based on empirical evidence.

3. METHODOLOGY

The machine learning project focused on classifying and predicting B cell epitopes using various algorithms. Fig 1 represents the design flow of the proposed model. The dataset, collected meticulously from the IEDB dataset through an extensive literature survey and analysis, primarily consisted of amino acid sequences. These sequences underwent preprocessing using feature scaling and engineering techniques to enhance the informativeness of the dataset. Feature engineering involved systematically creating and transforming features, including extracting information from amino acid sequences and incorporating physicochemical properties of amino acids.

For the classification of epitopes, the project considered several machine learning and deep learning algorithms. These algorithms underwent a training phase on the dataset, focusing on optimizing hyperparameters to enhance predictive performance. Rigorous evaluation was conducted using metrics such as accuracy, precision, recall, and F1 score. The effectiveness of each algorithm in epitope classification was rigorously assessed using testing datasets. Results were analyzed to draw conclusions regarding their efficacy in immunoinformatics and vaccine development.

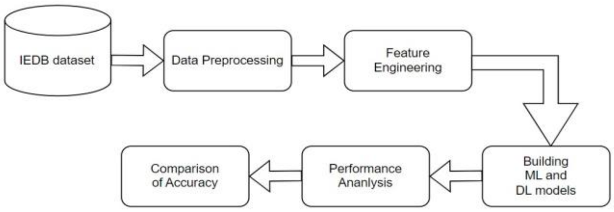

Fig. 1 Design Flow

3.1 Collection of Benchmark Dataset and Preprocessing

The dataset utilised in this study was sourced from the LBtope database [1]. Specifically, the LBtope fixed non redundant dataset was employed, which comprises positive epitope patterns totaling 7825 instances and negative epitope patterns totaling 7854 instances. Notably, this dataset exhibits a near-balanced distribution between positive and negative epitope patterns, fostering a robust foundation for machine learning model training and evaluation. Each epitope pattern within the dataset was standardised to a fixed length of 20 amino acids, ensuring uniformity and compatibility across the dataset. The uniform length facilitates the application of machine learning algorithms, streamlining feature extraction and model training processes. Prior to model training, the dataset underwent preprocessing steps to prepare it for analysis. The preprocessing pipeline encompassed several key stages, including data extraction, formatting, and integration. Subsequently, the extracted patterns were organised into structured data frames. Each pattern was represented as a row within the data frame, with an accompanying label denoting its class (positive or negative). The formatted data frames were then exported to CSV files, facilitating seamless integration into the machine learning pipeline. To ensure a balanced representation of positive and negative epitope patterns within the dataset, a stratified random sampling approach was employed. The positive and negative epitope pattern datasets were merged into a unified dataset, with instances shuffled to mitigate bias. The resulting dataset served as the foundational input for subsequent machine learningand deep learning models training and evaluation.

For the IEDB dataset utilised in this study, a comprehensive collection of epitope sequences was obtained from the Immune Epitope Database (IEDB) [2]. This dataset comprises epitope sequences derived from various antigens, with each sequence exhibiting a variable length ranging between 8 to 20 amino acids. In addition to the epitope sequences, information regarding the qualitative measure and response frequency associated with each epitope was acquired from the IEDB dataset. The qualitative measure denotes the experimental outcome of epitope analysis, categorised as either "positive" or "negative", providing valuable labels for supervised learning tasks. Furthermore, the response frequency indicates the frequency of immune response elicited by each epitope, offering insights into epitope immunogenicity. Following dataset acquisition, a series of preprocessing steps were performed to prepare the data for subsequent analysis. The preprocessing pipeline encompassed several key stages, including data extraction, formatting and cleansing. The raw dataset was initially imported into a structured format using the Python pandas library [3]. Subsequently, specific columns containing relevant information, including epitope sequences, qualitative measures, and response frequencies, were selected for further analysis. To ensure data integrity and consistency, various data cleaning operations were conducted. This involved removing duplicate entries, handling missing values, and standardising column names.

Additionally, sequences with lengths falling outside the specified range (i.e., 8 to 20 amino acids) were filtered out to maintain dataset coherence. To facilitate uniformity and compatibility across epitope sequences, preprocessing operations such as sequence trimming and normalisation were applied. Specifically, non-alphabetic characters were removed, and all alphabetic characters were converted to uppercase to standardise sequence representation. The preprocessed dataset serves as the foundational input for subsequent machine learning and deep learning models training and evaluation, providing a comprehensive resource for epitope prediction and analysis.

3.2 Feature Engineering

Feature engineering plays a crucial role in epitope prediction, enabling the extraction of informative features from epitope sequences to facilitate accurate classification. Here we describe the feature engineering process employed for the LBtope Fixed Non-Redundant dataset, encompassing the derivation of dipeptide-based features and antigenicity scales. Dipeptide-Based Features, representing consecutive pairs of amino acids within epitope sequences, serve as fundamental units for feature extraction. To capture the compositional characteristics of epitopes, a dictionary of dipeptide frequencies was constructed for each epitope sequence. Subsequently, the dipeptide frequencies were normalised to account for sequence length variations, yielding a representative feature vector for each epitope. The AAP (Amino Acid Pair) antigenicity scale, proposed by Chen et al. [4], was leveraged to quantify the antigenic propensity of dipeptides within epitope sequences. The AAP scale quantifies the ratio of dipeptide frequencies in the positive epitope set to their counterparts in the negative set. The resulting antigenicity values, normalised between +1 and -1, provide insights into the discriminatory power of dipeptides in distinguishing epitopes from non-epitopes. In addition to the AAP scale, the AAT (Amino Acid Triplet) antigenicity scale was adopted to capture higher-order amino acid interactions within epitope sequences. Inspired by SVMTriP [5], the AAT scale evaluates the antigenic potential of amino acid triplets based on their frequency distribution across epitope and non-epitope sequences. Analogous to the AAP scale, the AAT antigenicity values are normalised to facilitate comparative analysis and interpretation. The feature vectors for epitope sequences were constructed by concatenating dipeptide frequencies with AAP and AAT antigenicity scores. Each feature vector encapsulates comprehensive information regarding the compositional and antigenic attributes of epitopes, enabling effective discrimination between epitope and non-epitope sequences.

In addition to the AAP and AAT antigenicity scales previously discussed [4,5], we extended the feature engineering process for the IEDB dataset to encompass additional features aimed at capturing diverse aspects of epitope sequences. The feature engineering pipeline leveraged a combination of compositional, positional, and frequency-based features to enhance the discriminative power of the predictive model. Compositional features, such as dipeptide frequencies, were extracted to capture the local amino acid composition within epitope sequences. Dipeptide-based feature vectors were constructed to quantify the occurrence of consecutive pairs of amino acids, providing insights into the sequence- level compositional patterns. Positional features, including VOD (Vector of Occurrence Density) [6], APOV (Average Position of Occurrence) [7], and RAPOV (Reverse Average Position of Occurrence), were computed to encode positional information within epitope sequences. These features characterise the distribution and arrangement of amino acids along the sequence length, facilitating the identification of spatial motifs and patterns relevant to epitope recognition. Frequency-based features, such as response frequency, were incorporated to capture the prevalence and immunogenicity of epitope sequences. Response frequency values, obtained from experimental data, were integrated into the feature vector to provide quantitative measures of epitope immunogenicity, augmenting the predictive capabilities of the model. The feature vectors for epitope sequences were constructed by concatenating dipeptide frequencies, AAP and AAT antigenicity scores, positional features (VOD, APOV, RAPOV) [8], and response frequency values. Each feature vector encapsulates comprehensive information regarding the compositional, positional, and immunogenic attributes of epitopes, facilitating accurate epitope classification and prediction.

3.3. Prediction Models

The Support Vector Machine (SVM) algorithm, a powerful and widely used supervised learning method, was employed for epitope prediction on the LBtope Fixed dataset. In this study, a linear kernel was chosen for SVM modelling, which implies that the decision boundary is linear in the feature space. The training process involved partitioning the dataset into training and testing subsets using a stratified splitting strategy. The training subset was used to train the SVM model, where the algorithm iteratively adjusted the hyperplane parameters to minimise classification errors and maximise the margin between classes.

K-Nearest Neighbors (KNN) algorithm was employed to perform classification tasks on two distinct datasets: LBtope fixed dataset and the IEDB dataset. For both datasets, a KNN model with a hyperparameter of k=3 was instantiated and trained using the respective training data.

Logistic regression with polynomial features was utilised to perform classification tasks on the LBtope fixed dataset. Prior to fitting the logistic regression model, polynomial features of degree 2 were generated using the PolynomialFeatures transformer from the scikit-learn library. This process facilitated the creation of higher- order feature interactions while maintaining computational tractability. Specifically, the training features were transformed into polynomial features, and the same transformation was applied to the test features toensure consistency. Subsequently, a logistic regression model was instantiated and trained using the transformed training features and their corresponding labels. The logistic regression model aimed to learn the relationship between the polynomial features and the target labels, thereby enabling the classification of LBtope fixed dataset instances into relevant classes.

To optimise the performance of the Random Forest Classifier on the LBtope fixed dataset and the IEDB dataset, a hyperparameter tuning approach utilising GridSearchCV was employed. For the LBtope fixed dataset, a grid search was conducted over the hyperparameter space of the number of estimators (n_estimators) in the Random Forest Classifier. Specifically, values ranging from 50 to 300 were considered for the number of estimators. The grid search was configured with 3-fold cross-validation and evaluated based on accuracy. Following the grid search, the best performing value for n_estimators was identified as 250, indicating the optimal number of decision trees to be included in the ensemble model. Consequently, a Random Forest Classifier was instantiated with the optimal hyperparameters, utilising 250 decision trees. Similarly, for the IEDB dataset, the same grid search procedure was employed, resulting in the determination of 300 as the optimal value for n_estimators. Thus, a Random Forest Classifier was instantiated for the IEDB dataset with 300 decision trees. This parameter tuning process aimed to enhance the predictive performance of the Random Forest Classifier on both datasets by selecting the most suitable hyperparameters through systematic evaluation.

XGBoost, an efficient and scalable gradient boosting framework, was employed for classification tasks on both the LBtope fixed dataset and the IEDB dataset. For the LBtope fixed dataset, an XGBoost classifier was instantiated with the objective set to binary logistic regression and hyperparameters configured with 100 estimators and a learning rate of 0.1. The model wasthen trained using the training features and their corresponding labels. Similarly, for the IEDB dataset, an XGBoost classifier with identical hyperparameters was instantiated and trained. The choice of XGBoost algorithm stems from its capability to handle complex datasets, while the selection of hyperparameters was guided by empirical observations and domain knowledge.

Convolutional Neural Networks (CNNs) were employed for classification tasks on the LBtope fixed dataset and the IEDB dataset, demonstrating their efficacy in capturing spatial patterns from sequential data. For the LBtope fixed dataset, preprocessing steps involved feature scaling using MinMaxScaler to normalise input features. Subsequently, the input data was reshaped to conform to the requirements of CNNs. A CNN model was constructed comprising convolutional layers, batch normalisation, max-pooling layers, dense layers, and dropout regularisation. Specifically, the model consisted of convolutional layers with ReLU activation followed by batch normalisation and max-pooling to extract hierarchical features. Dropout regularisation was incorporated to mitigate overfitting, while dense layers with ReLU activation were utilised for non-linear transformations. The final layer employed a sigmoid activation function for binary classification. The model wascompiled using the Adam optimizer with binary cross- entropy loss and trained over 200 epochs. Similarly, for the IEDB dataset, preprocessing steps and model architecture were analogous, involving feature scaling, reshaping, convolutional layers with max-pooling, and dense layers. However, the number of epochs was set to 100 for training. This approach aimed to leverage CNNs' ability to automatically learn relevant features from raw input data, facilitating robust classification performance on both LBtope fixed and IEDB datasets.

This study investigated the potential of transfer learning to enhance the performance of a deep learning model for B-cell epitope classification. Pre-trained GloVe word embeddings were employed to capture semantic relationships between amino acids. These embeddings were loaded and utilised within a frozen Embedding layer of the Convolutional Neural Network (CNN) architecture. This layer serves to map amino acid sequences into a lower-dimensional vector space, while preserving inherent relationships between the amino acids. By freezing the embedding layer weights during training, the study essentially transferred the knowledge learned from a broader amino acid space to the specific task of B-cell epitope classification.

For the Convolutional Neural Network (CNN) model, we employed a rigorous evaluation strategy within our comprehensive model assessment. The dataset underwent a 5-fold cross-validation procedure, ensuring robustness invalidation. Additionally, to further enhance the evaluation of the CNN model, a dedicated validation set was allocated.Our analysis of the CNN model's performance revealed notable insights into its generalisation capabilities. With a focus on B-cell epitope classification, our CNN model demonstrated a cross-validation accuracy of 55.69%. This outcome underscores the effectiveness of our approach and signifies a significant contribution to the field of computational immunology research.

Generative Adversarial Networks (GANs) was used to enhance the training dataset for B cell epitope classification. GANs were utilised to generate synthetic epitope sequences, which were then seamlessly integrated with the original dataset to enrich the diversity of training samples for classification models. The approach involved training a generator network to produce synthetic epitope sequences based on patterns learned from the original dataset, aiming to generate realistic sequences closely resembling those observed in experimental data. Simultaneously, a discriminator network was trained to differentiate between real epitope sequences from the original dataset and synthetic sequences generated by the generator, ensuring the authenticity of the augmented data. By leveraging GANs for data augmentation, we aimed to improve the robustness and generalisation capability of our classification models for B cell epitopes.

4. RESULTS AND DISCUSSION

4.1 Evaluation Metrics

In evaluating our epitope classification models, we utilised key metrics such as accuracy, Area Under the Receiver Operating Characteristic (ROC) Curve (AUC), and confusion matrices. Accuracy, a fundamental metric, reflects the overall correctness of our models by considering both true positive and true negative predictions in relation to the total number of samples. The ROC curve and its corresponding AUC score provide valuable insights into the models' ability to distinguish between positive and negative epitopes across different classification thresholds. A higher AUC score indicates stronger discriminatory power, capturing the balance between true positive and false positive rates. The confusion matrix further detailed our models' performance, delineating predictions into true positives, true negatives, false positives, and false negatives. This breakdown offered a nuanced view of specific prediction errors and allowed for a comprehensive assessment of our models in epitope classification.

4.2 Performance Evaluation and Comparison

In our epitope classification study with LBtope Fixed Non-Redundant Dataset, we evaluated the performance of various machine learning and deep learning models. The Support Vector Machine (SVM) exhibited an accuracy of 65.52% and an AUC of 0.71, highlighting its ability to correctly classify epitopes and its decent discriminatory power. K-Nearest Neighbors (KNN) demonstrated a slightly higher accuracy at 66.13%, coupled with an AUC of 0.70, showcasing its competitive performance in capturing epitope distinctions. Logistic Regression, with an accuracy of 65.14% and an AUC of 0.71, demonstrated consistency in its predictive abilities. XGBoost, another gradient boosting algorithm, achieved an accuracy of 65.56% and an AUC of 0.71, indicating its robust performance in epitope classification. Notably, Random Forest emerged as a top performer with an accuracy of 68.35% and an AUC of 0.75, emphasising its superior predictive capabilities. Lastly, the ConvolutionalNeural Network (CNN) displayed an accuracy of 64.14% and an AUC of 0.69, showcasing its competence in capturing complex patterns within epitope data. Fig 2 shows the comparison of acuuracy.

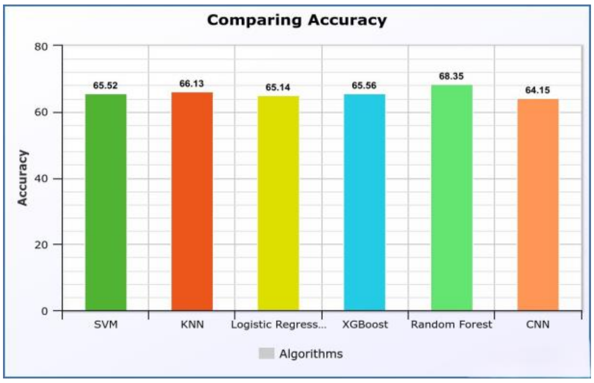

Fig. 2 Comparing Accuracy for LBtope Fixed Non-Redundant dataset

Table -1: Comparing Accuracy and AUC score of prediction models for LBtope Fixed Non-Redundant dataset.

Model	**Accuracy**	**AUC**
SVM	65.52	0.71
KNN	66.13	0.70
Logistic Regression	65.14	0.71
XGBoost	65.56	0.71
Random Forest	68.35	0.75
CNN	64.15	0.69

By applying transfer learning in conjunction with a Convolutional Neural Network (CNN) and utilising pre-trained GloVe word embeddings to capture semantic connections among amino acids, we attained a 51% accuracy.Employing a Generative Adversarial Network (GAN) in conjunction with a Convolutional Neural Network (CNN), we achieved a notable accuracy of 58.34%. The confusion matrix provides a comprehensive view of the performance of a classification model by showing the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions.

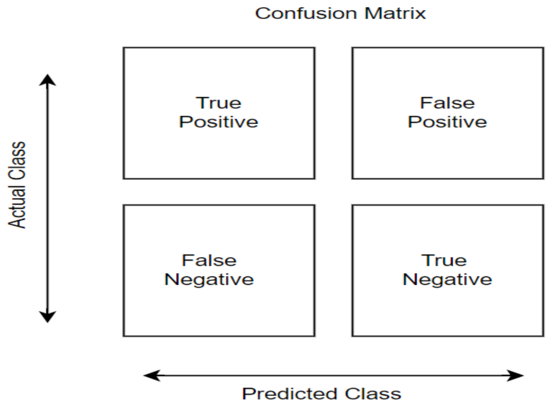

Fig. 3 Confusion matrix

Table -2: .Confusion matrix Results for different models for LBtope Fixed Non-Redundant dataset

Model	True Positive	False Positive	False Negative	True Negative
SVM	1049	481	600	1006
KNN	1092	438	624	982
Logistic Regression	1025	505	588	1018
XGBoost	1042	488	592	1014
Random Forest	1084	446	550	1056
CNN	557	237	327	447

In our exploration of epitope classification using the IEDB dataset, distinct machine learning and deep learning models were employed, each showcasing varying degrees of effectiveness. The K-Nearest Neighbors (KNN) model exhibited a notable accuracy of 70.89% and an AUC of 0.80, indicating its competence in capturing epitope patterns within the dataset.

Remarkably, XGBoost and Random Forest models outperformed with exceptional accuracy scores of 99.88% and 99.84%, respectively, coupled with perfect AUC scores of 1.00. This exceptional performance underscores the robustness of ensemble methods in epitope classification tasks. Additionally, the Convolutional Neural Network (CNN) demonstrated a commendable accuracy of 73.74% and an AUC of 0.84, highlighting its efficacy in capturing complex patterns within epitope sequences. Fig 4 shows the comparison of accuracy.

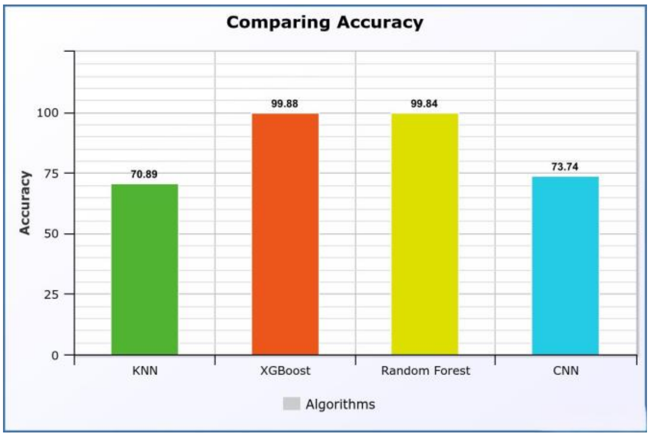

Fig. 4 Comparing Accuracy for IEDB dataset

Table - 3: Comparing Accuracy and AUC score of prediction models for IEDB dataset

Model	Accuracy	AUC
KNN	70.89	0.80
XGBoost	99.88	1.00
Random Forest	99.84	1.00
CNN	73.74	0.84

Additionally, we employed a 5-fold cross-validation approach on the Convolutional Neural Network (CNN), revealing an average accuracy of 55.69%. The confusion matrix (as in Fig.1) provides a comprehensive view of the performance of a classification model by showing the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions.

5 . CONCLUSIONS

In conclusion, the existing landscape of B-cell epitope (BCE) prediction models has seen notable advancements. However, certain drawbacks persist, such as feature selection challenges and model complexity. These limitations motivate the proposed project, which aims to address existing gaps and intricacies associated with linear B-cell epitope classification and prediction. By undertaking a comprehensive comparative analysis, the project seeks to refine the accuracy and robustness of BCE prediction models, contributing to the advancement of immune informatics.

In the proposed system, the emphasis is on a systematic approach that explores and evaluates various machine learning algorithms, employs feature engineering for improved model interpretability, and investigates transfer learning to benchmark pre-trained models. Generative Adversarial Networks are used to address imbalanced datasets, generating synthetic examples of the minority class to enhance the learning of the model and improve classification accuracy. The project's ultimate goal is to optimize the accuracy of epitope classification, a critical factor in immunology and vaccine development. By shedding light on the strengths and limitations of different algorithms, the proposed system aims to provide valuable insights for the development of more effective and precise immunotherapies and vaccines.

REFERENCES

[1] Saha, S., Raghava, G.P.S., 2004. BcePred: prediction of continuous B-cell epitopes in antigenic sequences using physico-chemical properties. Lect. Notes Comput. Sci. (Incl. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinforma.) 3239. https://doi. org/10.1007/978-3- 540-30220-9_16.

[2] Vita R, Overton JA, Greenbaum JA, Ponomarenko J, Clark JD, Cantrell JR, Wheeler DK, Gabbard JL, Hix D, Sette A, Peters B. The immune epitope database (IEDB)

3.0. Nucleic Acids Res. 2015 Jan;43(Database issue):D405-12. doi: 10.1093/nar/gku938. Epub 2014 Oct 9. PMID: 25300482; PMCID: PMC4384014.

[3] McKinney, Wes. "Data Structures for Statistical Computing in Python." Proceedings of the 9th Python inScience Conference. 2010.

[4] Chen, J., Liu, H., Yang, J. and Chou, K.C., 2007. Prediction of linear B-cell epitopes using amino acid pair antigenicity scale. Amino acids, 33, pp.423-428.

[5] Yao, B., Zhang, L., Liang, S. and Zhang, C., 2012. SVMTriP: a method to predict antigenic epitopes using support vector machine to integrate tri-peptide similarity and propensity.

[6] Khan, Y.D., Batool, A., Rasool, N., Khan, S.A., Chou, K.-C., 2018a. Prediction of nitrosocysteine sites using position and composition variant features. Lett. Org. Chem. https://doi.org/10.2174/157017861566618080212 2953.

[7] Khan, Y.D., Ahmed, F., Khan, S.A., 2014. Situation recognition using image moments and recurrent neural networks. Neural Comput. Appl. 24, 1519–1529.

[8] Muhammad Attique, Tamim Alkhalifah, Fahad Alturise, Yaser Daanial Khan, DeepBCE: Evaluation of deep learning models for identification of immunogenic B-cell epitopes, Computational Biology and Chemistry, Volume 104, 2023, 107874, ISSN 1476-9271, https://doi.org/10.1016/j.compbiolchem.2023.107874

[9] Janeway, C. A. (2001). Immunobiology: The immune system in health and disease (5th ed.). Garland Science.

[10] El-Mancy, A. M., & Kazdar, H. O. (2023). In silico B-cell epitope prediction: Methods, databases, and future directions. Briefings in Bioinformatics, 24(2), bbab440. [doi: 10.1093/bib/bbab440]

[11] Galanis, K.A., Nastou, K.C., Papandreou, N.C., Petichakis, G.N., Pigis, D.G., Iconomidou, V.A., 2021. Linear B-cell epitope prediction for in silico vaccine design: a performance review of methods available via command-line interface. Int. J. Mol. Sci. https://doi.org/10.3390/ijms22063210.

[12] Blythe, M.J., Flower, D.R., 2009. Benchmarking B cell epitope prediction: underperformance of existing methods. Protein Sci. 14. https://doi.org/10.1110/ Ps.041059505.

[13] Pellequer, J.L., Westhof, E., 1993. PREDITOP: A program for antigenicity prediction. J. Mol. Graph. 11. https://doi.org/10.1016/0263-7855(93)80074-2.

[14] Saha, S., Raghava, G.P.S., 2004. BcePred: prediction of continuous B-cell epitopes in antigenic sequences using physico-chemical properties. Lect. Notes Comput. Sci. (Incl. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinforma.) 3239. https://doi. org/10.1007/978-3-540-30220-9_16.

[15] Odorico, M., Pellequer, J.L., 2003. BEPITOPE: predicting the location of continuous epitopes and patterns in proteins. J. Mol. Recognit. 16, 20–22. https://doi.org/ 10.1002/jmr.602.

[16] Alix, A.J.P., 1999. Predictive estimation of protein linear epitopes by using the program PEOPLE. Vaccine 311–314. https://doi.org/10.1016/S0264-410X(99)00329-1.

[17] Blythe, M.J., Flower, D.R., 2009. Benchmarking B cell epitope prediction: underperformance of existing methods. Protein Sci. 14. https://doi.org/10.1110/ Ps.041059505.

[18] Jespersen, M.C., Peters, B., Nielsen, M., Marcatili, P., 2017. BepiPred-2.0: improving sequence-based B-cell epitope prediction using conformational epitopes. Nucleic Acids Res. 45, W24–W29. https://doi.org/10.1093/nar/gkx346.

[19] Saha, S., Raghava, G.P.S., 2006. Prediction of continuous B-cell epitopes in an antigen using recurrent neural network. Proteins Struct. Funct. Genet 65, 40–48. https://doi. org/10.1002/prot.21078.

[20] 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. https://doi.org/10.1371/journal.pone.0045152. [21] Singh, H., Ansari, H.R., Raghava, G.P.S., 2013. Improved method for linear B-Cell epitope prediction using Antigen's primary sequence. PLoS One 8. https://doi.org/ 10.1371/journal.pone.0062216.

[22] Haodong Xu, Zhongming Zhao, 2022. NetBCE: An Interpretable Deep Neural Network for Accurate Prediction of Linear B-cell Epitopes. https://doi.org/10.1016/j.gpb.2022.11.009.

[23] Tatiana I Shashkova1, Dmitriy Umerenkov, Mikhail Salnikov, Pavel V Strashnov, Alina V Konstantinova, Ivan Lebed, Dmitriy N Shcherbinin, Marina N Asatryan, Olga L Kardymon, Nikita V Ivanisenko1. SEMA: Antigen B-cell conformational epitope prediction using deep transfer learning. https://doi.org/10.3389/fimmu.2022.960985.