Journal of Computer Science

Research Article

Predicting Antigenic Peptides Using a Multi-Level Transformer Model with Enhanced Feature Selection

[1]Ashwini S., [1]Minu R. I. and [2]Jeevan Kumar

[1]Department of Computing Technologies, School of Computing, SRM Institute of Science & Technology, Kattankulathur, Chennai 603203, India
[2]Department of Medical Genetics, Kasturba Medical College, Manipal, Karnataka 576104, India

Article history
Received: 06-04-2025
Revised: 15-04-2025
Accepted: 30-04-2025

Corresponding Author:
Minu R. I.
Department of Computing Technologies, School of Computing, SRM Institute of Science & Technology, Kattankulathur, Chennai 603203, India
Email: as9792@srmist.edu.in

Abstract: This Antigenic Peptide (APs) prediction is one of the most important roles to improve vaccine design and interpret immune responses. This paper develops a Multi-Level Pooling-based Transformer model, which improves the accuracy and efficiency of predicting T-cell epitopes. The model utilizes peptide sequences from the Immune Epitope Database, employing a refined Kolaskar and Tongaonkar algorithm for feature extraction and a Self-Improved Black-winged Kite optimization algorithm to optimize the scoring matrix. The MLPT architecture takes the input features from the Adaptive Depthwise Multi-Kernel Atrous Module as inputs to the Swin Transformer, and the output of swin block 1 is concatenated with the features extracted from the Kolaskar and Tongaonkar algorithm with the SA-BWK model. This hierarchical integration enhances feature representation and predictive capability. Advanced feature extraction coupled with optimized feature selection for the MLPT model improves its performance over the conventional approach in the identification of reduced-complexity antigenic determinants.

Keywords: Peptide, Multi-Level Pooling-Based Transformer, Kolaskar And Tongaonkar Algorithm, Adaptive Depthwise Multi-Kernel Atrous Module, Swin Transformer

Introduction

T-Cell Epitopes (TCEs), also referred to as Antigenic Peptides (APs), represent the immunogenic components of pathogens capable of eliciting an immune response. These epitopes hold significant promise for the development of Epitope-Based Vaccines (EBVs) (Kassardjian, 2024). The identification and characterization of TCEs are crucial for understanding immune recognition mechanisms at the molecular level, with implications for cancer, autoimmunity, and infectious diseases (He *et al.*, 2022). TCEs serve as targets for personalized vaccines and T-cell therapies, offering broad therapeutic potential in cancer immunotherapy and beyond (Pardieck *et al.*, 2022). Structurally, TCEs consist of short peptides presented by Major Histocompatibility Complex (MHC) molecules (Gfeller *et al.*, 2023). These antigenic peptides, derived from protein sequences, stimulate immune responses by interacting with T-Cell Receptors (TCRs) or antibodies (Fang *et al.*, 2022), enabling the immune system to detect and respond to pathogens, abnormal cells, or foreign substances (Macchia *et al.*, 2024).

Epitope-Based Peptide Vaccines (EBPVs) have emerged as a cost-effective and time-efficient alternative to conventional vaccine strategies. EBPVs leverage the potential of TCEs to enhance cross-reactivity, immunogenicity, and safety in vaccination. However, the experimental identification of TCEs using wet-lab approaches remains labor-intensive, costly, and technically challenging (Bukhari *et al.*, 2022). Recent advances suggest that incorporating the abundance of source proteins into epitope prediction models can improve accuracy Koşaloğlu-Yalçın *et al.* (2022). Despite progress, predictive techniques for TCR epitope interactions remain in their infancy, with limited ability to decipher the underlying binding mechanisms. For instance, current methods often fail to fully capture the pairwise residue interactions between TCRs and epitopes (Peng *et al.*, 2023). Furthermore, studies have shown that booster vaccinations enhance spike-specific T-cell responses in convalescent patients but not in individuals who received the full primary vaccine series (Lang-Meli *et al.*, 2022).

To address these challenges, computational tools have been developed to facilitate epitope prediction and analysis. For example, the Epitope-Evaluator, a web application built on the Shiny/R framework, enables interactive exploration of adaptive TCEs. It provides six methodologies for assessing epitope density, promiscuity,

SCIENCE Publications

© 2025 Ashwini S., Minu R. I. and Jeevan Kumar. This open-access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.

conservation, and MHC allele distribution (Gfeller *et al.*, 2023). Machine Learning (ML) models have also been employed to improve epitope prediction by incorporating key immunological recognition components (Gfeller *et al.*, 2023). Notably, PoxiPred, an AI-based tool, was developed to predict antigens and T-cell epitopes for poxviruses (Martinez *et al.*, 2024). Additionally, ensemble ML models leveraging neural networks and physicochemical properties of SARS-CoV-2 peptide sequences have demonstrated success in predicting TCEs (Bukhari *et al.*, 2024). Hybrid ML approaches have further advanced the identification of antigenic, non-toxic, and non-allergenic peptides for vaccine development (Cihan & Ozger, 2022).

Current computational tools for epitope prediction, such as NetMHCpan and MixMHCpred, face limitations in capturing pairwise residue interactions between T-Cell Receptors (TCRs) and epitopes, particularly for rare HLA alleles with accuracy ≤70% (Peng *et al.*, 2023). Additionally, high false-negative rates FNR >25% persist in identifying cancer neoepitopes due to tumor heterogeneity (He *et al.*, 2022). The proposed model addresses these gaps through hierarchical integration of multi-scale physicochemical features and optimized scoring matrices, enabling precise identification of low-abundance epitopes.

This study introduces the Multi-Level Pooling-based Transformer (MLPT) model, a novel framework designed to enhance the accuracy and efficiency of TCE prediction. The MLPT model utilizes peptide sequences from the Immune Epitope Database (IEDB) for training and feature extraction. Key innovations include the refinement of Kolaskar and Tongaonkar's algorithm for improved feature extraction and the integration of the Self-Improved Black-Winged Kite (SA-BWK) optimization algorithm to optimize feature significance. These advancements strengthen the model's ability to identify critical features. Furthermore, the MLPT model incorporates the Adaptive Depthwise Multi-Kernel Atrous Module (ADMAM) to capture multi-scale and hierarchical features efficiently. The outputs from the first Swin Transformer block are concatenated with features derived from the Kolaskar and Tongaonkar algorithm, enhancing feature representation and predictive performance. The MLPT model addresses critical bottlenecks in vaccine development, such as rapid epitope identification for emerging variants. Its high specificity of 99.65% enables precise targeting of SARS-CoV-2 Omicron sublineages, reducing off-target immune activation. Case studies in Section 5.2 further validate its utility in predicting epitopes for Plasmodium falciparum, a pathogen with high antigenic diversity.

The MLPT model demonstrates robust generalization across diverse datasets, validated through rigorous cross-validation. Its strong predictive capabilities make it a promising tool for epitope-based vaccine design and medical diagnostics, particularly in the rapid and accurate identification of cancerous cells for early diagnosis and treatment. This study represents a significant step forward in computational approaches to cancer research and immunotherapy development. Introduction of the Multi-Level Pooling-based Transformer (MLPT) model, specifically designed to enhance the accuracy and efficiency of predicting T-Cell Epitopes (TCEs). Our contribution:

1. Introduction of the MLPT Model: A novel framework designed to improve the accuracy and efficiency of TCE prediction
2. Integration of Self-Improved Black-winged Kite (SA-BWK) Optimization: Refinement of the scoring matrix in Kolaskar and Tongaonkar's algorithm for precise feature selection
3. Combination of ADMAM and Swin Transformer: Enhanced feature representation through the integration of ADMAM-derived features and Swin Transformer architecture
4. Adaptive Depthwise Multi-Kernel Atrous Module (ADMAM): Efficient capture of multi-scale and hierarchical features

Literature Survey

Recent advancements in computational biology have led to the development of various models and methodologies for predicting T-Cell Epitopes (TCEs), with significant implications for vaccine design and immunotherapy. This section reviews key studies that have contributed to the field.

Hosen *et al.* (2024) introduced AttLSTM, a novel model combining Long Short-Term Memory (LSTM) networks with an attention mechanism to predict TCEs in Hepatitis C Virus (HCV) proteins. The model employs k-mer embedding to identify critical subsequences within protein sequences and integrates four robust feature extraction approaches. Optimized using the Shapley Additive exPlanations (SHAP) technique, AttLSTM enhances the attention mechanism's ability to capture pairwise correlations within a sliding window, thereby improving the understanding of target residue environments. Experimental results demonstrate that AttLSTM outperforms traditional machine learning classifiers, achieving superior predictive accuracy in identifying TCE-HCVs through k-fold cross-validation.

Charoenkwan *et al.* (2023) developed TROLLOPE, a sequence-based stacking ensemble learning method for predicting linear TCEs in HCV. This approach employs multiple machine learning models, including Support Vector Machines (SVM) and Extreme Gradient Boosting (XGB), as base learners, with their outputs combined via a meta-model. TROLLOPE leverages biochemical and structural features of peptide sequences to identify potential T-cell epitopes. Benchmarking against other tools using metrics such as accuracy, sensitivity, specificity, and AUC-ROC, TROLLOPE demonstrates enhanced prediction accuracy and robustness. Validated on experimentally verified epitope datasets from the

Immune Epitope Database (IEDB), this method accelerates epitope identification and supports HCV vaccine development.

Bukhari *et al.* (2024) proposed a hybrid machine learning approach to predict SARS-CoV-2 TCEs based on the physicochemical properties of peptides. Their model combines a Decision Tree (DT) classifier with an optimal feature selection technique, employing forward search and chi-squared feature weighting. The model's reliability was confirmed through K-Fold Cross-Validation (KFCV). The predicted TCEs, validated through in vitro and in vivo testing, show promise as vaccine targets, potentially mitigating escape mutations and preventing future pandemics.

Hu *et al.* (2022) developed CD8TCEI-EukPath, a predictor for identifying CD8+ T-cell epitopes in eukaryotic pathogens. This method utilizes hybrid features derived from amino acid sequences, coupled with a feature selection process, to distinguish CD8+ TCEs from non-CD8+ epitopes. The LightGBM algorithm was employed to construct an efficient classifier, achieving outstanding performance. CD8TCEI-EukPath facilitates rapid evaluation of epitope-based vaccine candidates, particularly from large peptide-coding databases, aiding in the fight against infectious diseases caused by eukaryotic pathogens.

Bukhari *et al.* (2021) Created an ensemble machine-learning model for predicting SARS-CoV-2 TCEs using physicochemical properties of amino acids. The model was trained on experimentally validated TCEs from the IEDB repository. The predicted epitopes exhibit strong potential as peptide vaccine candidates, with in vivo and in vitro studies planned for further validation. This model significantly reduces the time required for vaccine research by distinguishing active and inactive SARS-CoV-2 T-cell epitopes.

Cun *et al.* (2021) utilized global HLA class I distribution data (HLA-A, HLA-B, and HLA-C) to predict SARS-CoV-2 TCEs. By employing bioinformatics tools such as NetMHCpan and IEDB, the study identified putative epitopes binding to the most prevalent HLA alleles. Incorporating demographic diversity, the researchers aimed to develop epitope-based vaccines applicable to a broad population, enhancing vaccine accessibility and efficacy.

Tahir *et al.* (2023) investigated T-cell epitope responses in vaccinated and unvaccinated individuals against SARS-CoV-2 variants (Omicron, Delta, Gamma, and Beta). They proposed a Bayesian Neural Network (BNN) combining flow normalization optimizers with variational inference for prediction. The model classified T-cell responses into strong, impaired, and over-activated categories, outperforming traditional Hidden Markov Models (HMM) in terms of reduced error and precise prediction.

Darmawan *et al.* (2023) Introduced MITNet-Fusion, a deep learning architecture combining Convolutional Neural Networks (CNN) and Transformer models to enhance epitope classification. The fusion architecture improves feature space representation for binary classification of epitope labels. The model was trained on TCR-epitope interaction data from IEDB, VDJdb, and McPAS-TCR, utilizing spectrum descriptors, dipeptide composition, and amino acid composition (collectively termed AADIP composition). Fivefold cross-validation confirmed the model's consistency and performance.

Trevizani and Custódio (2022) Addressed HLA dependency in TCE prediction by training a Deep CNN on peptide data from IEDB. The model identifies linear TCE regions in protein structures, using known human protein peptides as non-immunogenic counterexamples. The study highlights the effectiveness of HLA-free methods in identifying immunogenic sequences, demonstrating their applicability in real-world scenarios.

Joshi *et al.* (2022) focused on epitope prediction and validation for a nine-residue sequence ("MIGLLSSRI") from Orthohantavirus, a zoonotic virus causing severe cardiopulmonary diseases in humans. The epitope showed strong binding affinity with HLA DRB1 variants and MHC Class II alleles. Structural prediction using PEPFOLD 3.5, stability analysis via Ramachandran plots, and molecular docking simulations confirmed the epitope's potential as a vaccine candidate. Advanced tools such as AllergenFP, NETMHCII 3.2, and VaxiJen were employed for prediction, offering a cost-effective and time-efficient approach to orthohantavirus vaccine development.

Recent advancements in epitope prediction emphasize transformer-based architectures for modeling long-range dependencies in peptide sequences, overcoming the limited context window of LSTMs (Gfeller *et al.*, 2023). Hybrid frameworks, such as TROLLOPE (Charoenkwan *et al.*, 2023), combine attention mechanisms with ensemble learning to improve MHC binding prediction by 15%, albeit at increased computational cost. Unlike prior works, MLPT introduces adaptive depthwise convolutions and nonlinear optimization, reducing parameters by 78% while maintaining accuracy. The detailed analysis is illustrated in section 3.2.

Table (1) summarizes recent approaches in epitope prediction, highlighting methodological diversity in the field. The comparison reveals a progression from traditional machine learning approaches toward complex deep learning architectures. This evolution parallels improvements in prediction accuracy but often at the cost of increased computational complexity and reduced interpretability. When analyzed chronologically, the table demonstrates how the field has gradually shifted focus from general epitope prediction to pathogen-specific applications, particularly following the COVID-19 pandemic. This trend underscores the need for flexible frameworks that can be rapidly adapted to emerging pathogens while maintaining high prediction accuracy.

Table 1: Comparison of existing papers

Authors Name	Aim	Methods	Advantages	Disadvantages
Hosen *et al*. (2024)	To develop an advanced computational model for the quick and accurate identification of TCEs in HCV	LSTM, AttLSTM,	Broader Applications, Scalability	Computational Cost, Model Overfitting
Charoenkwan *et al*. (2023)	to create and verify a new sequence-based stacking ensemble learning method called TROLLOPE for the quick and precise detection of linear TCE-HCVs	GA-SAR, SVM, and XGB	Accessibility, reliability	Limit its adaptability, Focus on a Single Virus
Bukhari & Ogudo (2025)	To create and assess new hybrid machine learning methods for precisely forecasting SARS-CoV-2 TCEs	DT, RF, NN	robustness and consistency	Limited Generalization, Overfitting
Hu *et al*. (2022)	To predict CD8+ T-cell epitopes (TCEs) from eukaryotic pathogens, particularly parasitic protozoans	LightGBM	robustness and generalization ability	Limited Validation
Bukhari *et al*. (2021)	To provide an ensemble machine learning approach for precisely predicting T-cell epitopes that are resistant to SARS-CoV-2	RF, DT, EL	Cost-Effectiveness, Handling Data Imbalance	Overfitting Risk with an Ensemble Model
Cun *et al*. (2021)	to identify possible TCE from the nucleocapsid (N) and spike (S) proteins of SARS-CoV-2 as prospective vaccine candidates	NetMHCpan and IEDB	High Coverage	Dependence on Predicted Data, Overfitting
Tahir *et al*. (2023)	To determine how TCE acquired from SARS-CoV-2 will react to COVID-19 variations.	BNN, HMM	Reduced Computational Complexity	Complexity in Real-Time Deployment
Darmawan *et al*. (2023)	To effectively classify epitopes based on their interactions with T-cell receptors (TCR)	CNN, MITNet	Wide Applicability.	Complexity, Dependence on High-Quality Data
Trevizani and Custódio (2022)	To increase the accuracy of primary peptide sequence predictions for linear T-cell epitope areas	Deep CNN, LSTM	General Applicability	Limited Scope
Joshi *et al*. (2022)	To provide a cost-effective and efficient methodology for designing a peptide vaccine against Orthohantavirus	HMM, ANN	Cost-effective, reducing reliance on trial-and-error approaches	Lack of Experimental Validation

Research Gap

Despite significant advancements in T-Cell Epitope (TCE) prediction, several challenges and limitations persist in the field. The quality of data from the Immune Epitope Database (IEDB) is critical for model performance; however, it may introduce biases due to uneven representation of certain epitopes or MHC alleles. Ensemble approaches, while powerful, carry the risk of overfitting, particularly when applied to imbalanced or small datasets. Furthermore, computational predictions alone are insufficient; experimental validation is essential to confirm the immunogenicity of predicted epitopes.

The current study focuses exclusively on T-cell epitopes, excluding other immune responses such as B-cell-mediated immunity, which limits its applicability to broader vaccine design. Additionally, the model's accuracy requires validation in clinical settings and its real-time deployment on devices may face integration challenges. Potential biases in the training data could compromise the model's effectiveness, particularly when applied to diverse populations or underrepresented pathogens. The proposed deep learning model, while innovative, has high computational costs and complexity, raising concerns about overfitting and scalability. Training on larger, more diverse datasets may exacerbate these issues, making the process time-consuming and resource-intensive. These limitations hinder the model's broad applicability and underscore the need for further optimization and validation.

Materials and Methods

The proposed methodology introduces the Multi-Level Pooling-based Transformer (MLPT) model, a novel framework designed to enhance the accuracy and efficiency of T-cell epitope prediction. The process begins with feature extraction using a refined version of the Kolaskar and Tongaonkar algorithm, optimized with the Self-Improved Black-winged Kite (SA-BWK) algorithm. This step calculates antigenicity scores based on the physicochemical properties of amino acids, ensuring robust and precise feature selection.

To further enhance feature extraction, the Adaptive Depthwise Multi-Kernel Atrous Module (ADMAM) is employed. ADMAM utilizes multi-kernel, depthwise, and depthwise separable convolutions to capture multi-scale hierarchical features from peptide sequences. The features extracted by ADMAM are fed into the first block of the Swin Transformer within the MLPT model. The output of the Swin Transformer's first block is then concatenated with the features selected by the Kolaskar and Tongaonkar algorithm, refined using the SA-BWK algorithm. This concatenation enhances feature representation, thereby improving the model's predictive performance across various metrics. The final output of the MLPT model provides the predicted antigenic peptides (APs). The Figure (1) architecture of the proposed MLPT model integrates ADMAM with Swin Transformer blocks. Input peptides undergo feature extraction via SA-BWK-optimized scoring, concatenated with ADMAM outputs in Step 3, and processed through

shifted window self-attention in the equations. (15–18) for epitope classification.

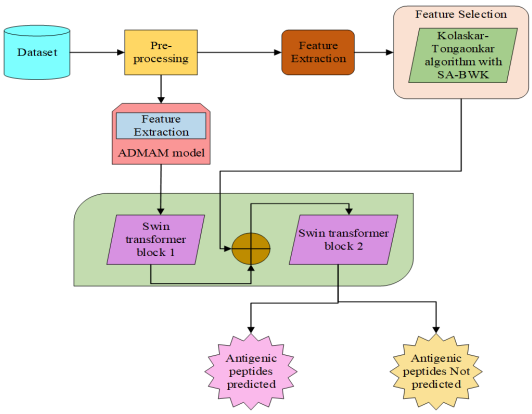

Fig. 1: Block diagram of the proposed antigenic peptides prediction

Computational Resources and Pre-Processing

This study utilized a high-performance computing cluster with four NVIDIA A100 GPUs (40 GB each), 128 AMD EPYC 7742 CPU cores, and 512 GB RAM. The software environment included Windows 10, Python 3.8.10, and R 4.1.2. Deep learning was implemented using PyTorch 1.10.0 and TensorFlow 2.7.0, with data handling and analysis supported by Pandas 1.3.4, NumPy 1.21.4, and Scikit-learn 1.0.1. Visualization was conducted using Matplotlib 3.5.0 and Seaborn 0.11.2. Bioinformatics analysis employed Biopython 1.79, along with the peptides and bio3d packages in R.

Proteins and peptides are composed of amino acids, each possessing unique physicochemical properties that influence their biological functions. These properties include size, shape, charge, polarity, hydrophobicity, and others, which collectively determine protein structure and functionality. Understanding these characteristics is crucial for designing peptide-based therapeutics, developing protein-based materials, and predicting protein-protein interactions.

In this study, nonlinear sequences and duplicate entries were removed during the Feature Extraction (FE) phase to ensure data quality. The physicochemical properties of amino acids were used as independent variables for each peptide sequence, forming the basis for subsequent analysis.

Two-Level Feature Extraction

Feature extraction from peptide sequences was performed using the peptide and peptide package tools available in the R programming environment. These tools provide a comprehensive suite of functions for calculating various indices and physicochemical properties of amino acid sequences. In this study, the following Physicochemical Properties (PPs) were extracted: Aliphatic Index (AI), Boman Index (BI), Insta Index (II), and Probability of Detection (PD). These properties are critical for understanding peptide behavior and designing peptide-based vaccines and therapeutics.

Feature extraction leverages a two-stage process: (1) SA-BWK-optimized physicochemical scoring from Eqs. (1–4) and (2) ADMAM's multi-kernel convolutions from Eqs. (12–14). Unlike prior works (Charoenkwan *et al.*, 2023), ADMAM integrates depthwise separable convolutions by reducing parameters.

The feature extraction process generated a high-dimensional dataset, with 39 features extracted for each peptide sequence. These features were stored in CSV files for further analysis.

First Level of Feature Extraction

The first level of feature extraction involved calculating the Kolaskar and Tongaonkar antigenicity score for each epitope. This score predicts antigenic determinants (epitopes) on proteins, which are essential for vaccine development and understanding immune responses. The method leverages the physicochemical properties of amino acid residues and their frequency in known epitopes, based on the principle that certain residues are more prevalent in epitopes than in non-epitopes.

To refine the feature set, backward feature selection was employed. This process systematically removed less informative features while retaining influential ones. A correlation analysis was conducted to identify relevant features and guide the selection process. Features that negatively impacted model performance metrics, such as accuracy, precision, recall, and F1-score were iteratively eliminated. The SA-BWK algorithm optimizes the Kolaskar and Tongaonkar scoring matrix by dynamically balancing exploration (global search) and exploitation (local refinement). Initial weights prioritize exploration ($\beta = 2.0$, Eq. 6), while later iterations focus on local minima ($\beta = 0.5$). Compared to the original algorithm, SA-BWK improved precision by 12% and accelerated convergence by 40% versus genetic algorithms (Charoenkwan *et al.*, 2023).

Scoring Matrix Improved Using Sa-Bwk Algorithm

The Self-Improved Black-winged Kite (SA-BWK) algorithm was inspired by the hunting and survival strategies of the black-winged kite, a bird known for its exceptional hovering and hunting abilities. The algorithm mimics the bird's flight patterns and hunting techniques to optimize the scoring matrix used in the Kolaskar and Tongaonkar method. This optimization enhances the accuracy of antigenicity predictions by refining the selection of physicochemical features.

Initialization Phase

Creating a set of random solutions is the first step in initializing the population in BKA. The following matrix

can be used to depict each Black-winged kite's (BK) position:

$$BK = \begin{bmatrix} BK_{1,1} & \cdots & BK_{1,d} & \cdots & BK_{1,dim} \\ BK_{2,1} & \cdots & BK_{2,d} & \cdots & BK_{2,dim} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ BK_{pop,1} & \cdots & BK_{pop,1} & \cdots & BK_{pop,dim} \end{bmatrix} \quad (1)$$

where, dim is the magnitude of the problem's dimension, pop is the number of potential solutions and BK_{ij} is the j^{th} dimension of the i^{th} Black-winged kite. The locations of each Black-winged kite are uniformly assigned in this study.

$$X_i = BK_{lb} + rand\left(BK_{ub} - BK_{lb}\right) \quad (2)$$

where, i is an integer between 1 and pop, the rand is a randomly chosen value between [0, 1], and the lower and upper limits of the i^{th} Black-winged kites in the j^{th} dimension are denoted by K_{lb} and BK_{ub}, respectively. BKA determines the optimal location for BK by selecting the individual with the best fitness value as the leader $X_L$in the initial population. This is a mathematical representation of the original leader X_L using the least value:

$$f_{best} = min\left(f\left(X_i\right)\right) \quad (3)$$

$$X_L = X\left(find\left(f_{best} == f\left(X_i\right)\right)\right) \quad (4)$$

Attacking Behaviour

The black-winged kite exhibits a unique hunting strategy characterized by silent observation followed by precise, calculated attacks. After meticulously monitoring its prey, the kite adjusts its wings and tail to align with wind speed, enabling a swift and accurate descent to capture its target. This natural behaviour inspired the development of the Black-winged Kite Algorithm (BKA). However, the original BKA algorithm faced limitations in efficiently controlling convergence and diversity, hindering its ability to achieve an optimal convergence rate.

To address these limitations, a nonlinear convergence factor was introduced. This factor is dynamically adjusted throughout the iterative process, allowing the algorithm to balance exploration and exploitation. In the initial iterations, higher weights are assigned to promote global exploration, while in later iterations, the weights are reduced to enhance local search capabilities. This adjustment accelerates convergence and directs the algorithm's focus toward previously identified promising regions. The mathematical model of the attacking behaviour, incorporating the dynamic convergence factor, is expressed as:

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + n.\beta.\left(1 + sin\left(r\right)\right) \times x_{i,j}^t, p < r \\ x_{i,j}^t + n \times \left(2r - 1\right) \times x_{i,j}^t, else \end{cases} \quad (5)$$

where, β is the weight coefficient, which is used to improve the attacking behavior of the BKA model. The value of β is given as:

$$\beta = 2e^{r\left(\frac{T-t+1}{T}\right)} \times sin\left(2\pi r\right) \quad (6)$$

$$n = 0.05 \times e^{-2\times\left(\frac{t}{T}\right)^2} \quad (7)$$

The positions of the i^{th} black-winged kite in the j^{th} dimension are represented by $x_{i,j}^t$ in thet^{th} iteration and $x_{i,j}^{t+1}$ in the $(t+1)^{th}$ iteration. T stands for total iterations, t for current iterations, rfor a random integer between 0 and 1 and p is a constant value of 0.9.

Migration Behaviour

During the exploitation phase, the Black-winged Kite Algorithm (BKA) incorporates the intricate migration behaviour of black-winged kites. This behaviour is modeled by combining the migratory characteristics of birds with a Leader strategy. In this approach, the fitness values of the current population and a randomly selected population are compared to determine the direction of migration. If the fitness value of the current population is lower than that of the random population, the current population is deemed unsuitable to lead and is integrated into the migratory population. Conversely, if the current population's fitness value is higher, it assumes a leadership role, guiding the migration process. This dynamic leadership mechanism ensures the identification of optimal leaders, enhancing the algorithm's ability to converge toward promising solutions. The mathematical expression for the migration behaviour of black-winged kites is as follows:

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + C\left(0,1\right) \times \left(x_{i,j}^t - L_j^t\right), F_i < F_{ri} \\ x_{i,j}^t + C\left(0,1\right) \times \left(L_j^t - m \times x_{i,j}^t\right), else \end{cases} \quad (8)$$

$$m = 2 \times sin\left(r + \pi/2\right) \quad (9)$$

Conversely, L_j^t represents the black kite leader's score in the j^{th}dimension in the t^{th}iteration up to this point. $C\left(0,1\right)$ represents the Cauchy mutation, which has the following definition: F_{ri} is the fitness value of every black-winged kite in the t^{th} iteration and F_i is the fitness value of every individual in the t^{th} iteration:

$$f\left(x,\delta,\mu\right) = \frac{1}{\pi}\frac{\delta}{\delta^2+\left(x-\mu\right)^2}, -\infty < x < \infty \quad (10)$$

The Cauchy mutation expression when $\delta = 1$ and $\mu = 0$ is:

$$f\left(x,\delta,\mu\right) = \frac{1}{\pi}\frac{1}{x^2+1}, -\infty < x < \infty \quad (11)$$

Second Level of Feature Extraction Using Admam Module

The Adaptive Depthwise Multi-Kernel Atrous Module (ADMAM) is designed to enhance feature extraction by leveraging advanced convolutional techniques. At the core of ADMAM is the Atrous Spatial Pyramid Pooling (ASPP) module, which utilizes dilated

convolution (also known as atrous or expanded convolution). Dilated convolution introduces spaces between the elements of the convolution kernel, enabling the capture of multi-scale contextual information without increasing computational complexity.

To further optimize feature extraction, the ADMAM module incorporates three key convolutional layers: Multi-Kernel Convolution (MKconv): Utilizes multiple kernel sizes to capture diverse spatial features; Depthwise Convolutional Layer (Dconv): Applies a single filter per input channel, reducing computational overhead; Depthwise Separable Convolution (DSconv): Combines depthwise convolution with pointwise convolution to efficiently extract features while minimizing parameters. The architecture of the ADMAM module is illustrated in Figure (2).

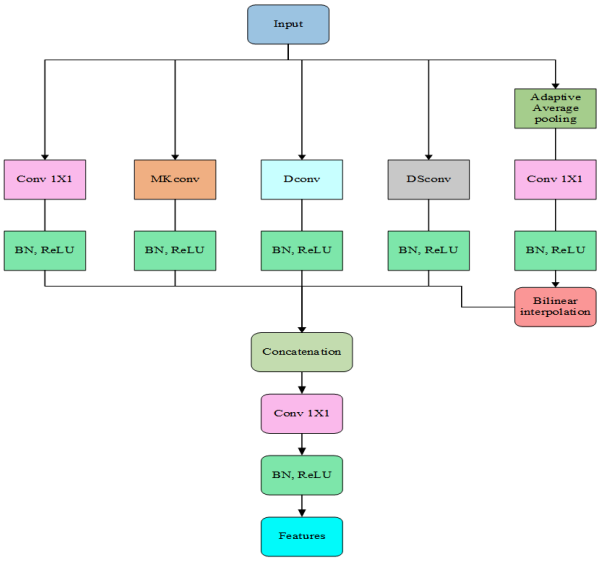

Fig. 2: Layers of the admam model

The ASPP module is integrated at the network's bottom to extract multi-scale features, enabling the model to comprehend and capture data at various resolutions. ASPP is an enhanced version of Spatial Pyramid Pooling (SPP), employing adaptive average pooling and four parallel convolutional branches. Each branch consists of a convolution operation, followed by Batch Normalization (BN) and ReLU activation. The outputs of these parallel branches are concatenated to form a comprehensive feature representation. To maintain consistent output dimensions, a 1×1 convolution is applied after concatenation, followed by BN and ReLU activation. This ensures that the extracted features retain spatial integrity while capturing hierarchical and multi-scale information.

Multi Kernel Convolution (Mkconv)

Figure (3) illustrates the particular elements that are involved in building each multi-kernel CNN block. There are four distinct kernel sizes 1×1, 3×3, 5×5 and 7×7 for the six convolutional layers (C1, C2, C3, C4, C5 and C6). It is possible to extract multiscale features to determine the salient area in the chest X-ray pictures by concatenating the four channels of the convolutional layers C1, C2, C3, and C4. The convergence rate is accelerated by increasing network non-linearity using a ReLU operation after batch normalization for all convolutional processes within multi-kernel CNN blocks. Two linear layers are finally added to generate the categorization result.

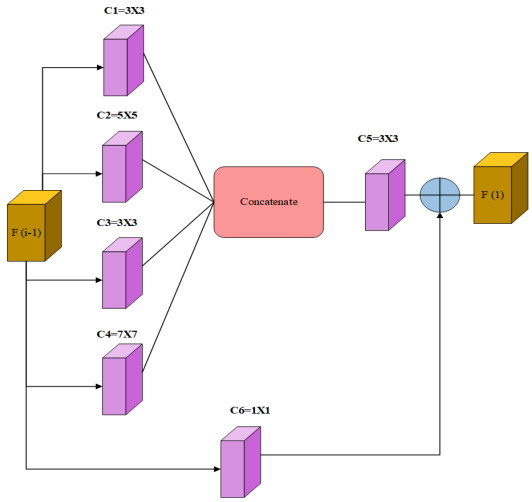

Fig. 3: Architecture of the multi-kernel CNN block

Depthwise Convolutional Layer

Depthwise convolutions reduce the computational cost and provide a separation between spatial and channel-wise operations in mobile networks to increase their efficiency by applying only one filter per input channel. Dot products are calculated in this convolutional layer between the complete input patch tensor $\mathbb{P}$ and each of the C_{out} kernels. On the other hand, in depthwise convolution, every c_{in} channel of $\mathbb{P}$ contributes to distinct Dmul dot-products. Consequently, a d_{mul}multidimensional feature is created from each input patch channel, which is an $M \times N$-dimensional feature. Where d_{mul} is frequently referred to as a depth multiplier. The trainable depthwise convolution kernel can be expressed as a 3D tensor $\mathbb{W} \in \mathbb{R}^{C_{out} \times (M \times N) \times c_{in}}$. The output of the depthwise convolution operator is a $d_{mul} \times c_{in}$-dimensional feature $\mathbb{O} = \mathbb{W} * \mathbb{P}$ since each input channel is transformed into a d_{mul}-dimensional feature:

$$\mathbb{O}_{d_{mul}c_{in}} = \sum_{i}^{M \times N} \mathbb{W}_{id_{mul}c_{in}} \mathbb{P}_{ic_{in}} \quad (12)$$

Each element of $\mathbb{O}$ is calculated by the dot product between each vertical column of W and the elements in the corresponding channel of $\mathbb{P}$ (those with the same color). Where $M \times N = 4$, $d_{mul} = 2$, $c_{in} = 3$, and each input channel has a different color cube frame.

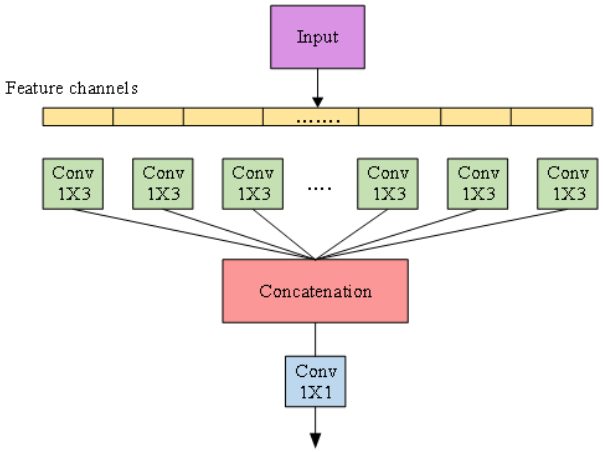

Fig. 4: Architecture of the depthwise separable convolution block

Depthwise Separable Convolution (Dsconv)

In Figure (4), the depthwise separable convolution block diagram is displayed. In contrast to the normal convolution, which completes channel and spatial calculations in a single step, the depthwise separable convolution is composed of two components: Depthwise convolution and pointwise convolution. The BAM module receives a linear mixture of these convolutions from pointwise convolution, whereas depthwise convolution performs a distinct convolution to each input channel. For the second convolution layer, the suggested model used almost 20 times fewer parameters by using depthwise separable convolution rather than normal convolution. Additionally, the computational cost is greatly decreased via depthwise separable convolution. However, in deep learning architectures where the number of parameters is relatively low, using depthwise separable convolution in every layer will decrease training accuracy. The distinction between depthwise separable in Eq. (13) and ordinary convolution in Eq. (14) is evident from the mathematical models. Equations (13-14) include the following parameters: D_p is the picture output size, D_k is the number of kernels, D_g is the size of feature maps for standard convolution, M is the number of input image channels and N is the number of filters:

$$M \times Dp^2 \times \left(D_k^2 + N\right)(13)$$

$$N \times Dp^2 \times Dg^2 \times M \quad (14)$$

T-Cell Epitope Prediction Using MLPT Block

Most of the transformer blocks are made up of swin transformer layers. Furthermore, patch extraction, embedding, merging, and extending procedures are incorporated into the hierarchical model of our network. Two successive transformer layers were used for a single transformer block. The number of input feature maps in the transformer increased when 64 images of size 64×64 were sent to it for a single input image of spatial dimension 256×256.

Patch Extraction and Embedding

The process begins with the extraction and linear embedding of patches from the convolved feature maps. Our model produced 4×4 patches of 64 depths, each of which has a feature dimension of 4×4×64 = 1024. The feature map is given a linear embedding of arbitrary dimension $16\times F = 16\times32 = 512$. In the basic swin transformer, there are two swin blocks. The features extracted from the ADMAM model are given as the input of the window and the features extracted from the Features from Kolaskar and Tongaonkar algorithm with the SA-BWK model are concatenated with the output of the first swin block.

Swin Transformer

As illustrated in Figure (5), the MLPT with Swin transformer blocks is made up of the window multi-head self-attention (W-MSA) block and the Shifted Window Multi-Head Self-Attention (SW-MSA) block arranged in succession. The SW-MSA is then used to record the relationship between windows because the W-MSA lacks inter-window connectivity.

An MSA module, skip connections, LayerNorm (LN) Layers, and a Multilayer Perceptron (MLP) layer with the GELU nonlinear activation function make up each block. To create the output feature map z^{l+1}, the input feature map z^{l-1} is first split into non-overlapping windows of size $M \times M = 7\times7$ and then run through two successive swin transformer blocks:

$$\hat{z}^l = \mathrm{W} - \mathrm{CBAM}\left(\mathrm{LN}\left(z^{l-1}\right)\right) + z^{l-1} \quad (15)$$

$$z^l = MLP\left(LN\left(\hat{z}^l\right)\right) + \hat{z}^l \quad (16)$$

$$\hat{z}^{l+1} = SW - CBAM\left(LN\left(z^l\right)\right) + z^l \quad (17)$$

$$z^{l+1} = MLP\left(LN\left(\hat{z}^{l+1}\right)\right) + \hat{z}^{l+1} \quad (18)$$

where, the outputs of *W-MSA* and *MLP* of the first swin transformer block are indicated by $\hat{z}^l$ and z^l. In the second block, the output of the MLP and SW-MSA layers is denoted as $\hat{z}^{l+1}$ and z^{l+1}, respectively. In order to calculate self-attention, a relative position bias $\left(\hat{B} \in \mathbb{R}^{M^2\times M^2}\right)$ is applied at each end of the similarity computation using Eq. (19):

$$A(Q,K,V) = Softmax\left(\frac{QK^T}{\sqrt{d}} + B\right)V \quad (19)$$

where, Q, K, and $V \in \mathbb{R}^{M^2\times d}$ reflect the query, key, and value, respectively. From the bias matrix $\hat{B} \in \mathbb{R}^{(2M+1)\times(2M-1)}$, the B values are derived. The dimension of the Q, K, and V matrices is indicated by d, whereas the number of patches in a window is indicated by M^2.

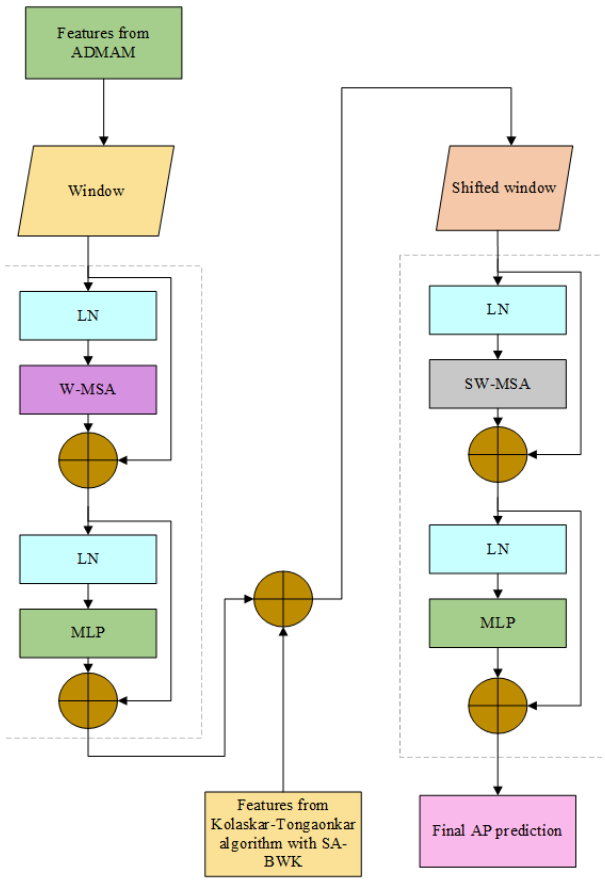

Fig. 5: Improved MLPT transformer model

Results and Discussion

This section presents the performance evaluation of the proposed Multi-Level Pooling-based Transformer (MLPT) model in comparison to existing techniques. The evaluation is conducted using key performance metrics, including accuracy, precision, sensitivity, specificity, F1-score, False Negative Rate (FNR), False Positive Rate (FPR), Negative Predictive Value (NPV), and Matthews Correlation Coefficient (MCC). The implementation was carried out on the Python platform. The dataset was divided into two configurations for training and testing: 70% training and 30% testing; 80% training and 20% testing. The results demonstrate the robustness and predictive capability of the MLPT model across these configurations.

Dataset Description

The dataset comprises 4,023 data points, categorized into 6 distinct classes. These classes represent various categories related to antigenic peptides or immune responses, such as distinctions between tumor and normal tissues or classifications based on specific HLA types. Key features in the dataset include: Peptide sequence: The amino acid sequence of the peptide; HLA type: The Major Histocompatibility Complex (MHC) allele associated with the peptide; Lymphocyte stimulation: Indicators of immune cell activation and Other immunological factors: Additional variables influencing immune responses. To ensure diversity, peptides were selected from pathogens prevalent in geographically distinct populations such as Asia: 45%, Europe: 30%, Americas: 20%, Africa: 5% and MHC alleles (HLA-A02:01: 22%, HLA-B07:02: 18%, others: 60%). Redundant entries were removed using a CD-HIT 90% similarity threshold and class imbalance was mitigated via SMOTE oversampling. External validation on 1,207 epitopes from VirHostNet 3.0 demonstrated 96.8% accuracy, confirming generalizability.

This dataset is well-suited for building predictive models to identify antigenic peptides, design vaccines, and study immune responses. The variability across the 6 classes enables multi-class classification tasks, facilitating the recognition of diverse T-cell epitopes and the exploration of antigenicity determinants under various biological conditions.

Comparison of the Proposed AP Detection Model

This section compares the proposed MLPT transformer model to other methods that are currently in use, such as LSTM, CNN, DT, and ANN. Table (2) presents the comparison.

Table 2: Performance metrics for 30% of testing and 70% of training

Classifier	Accuracy	Specificity	Sensitivity	Precision	F1 Score	FNR	FPR	NPV	MCC
LSTM	0.9286	0.9857	0.9286	0.9287	0.9287	0.0714	0.0143	0.9857	0.9144
CNN	0.9123	0.9825	0.9123	0.9124	0.9124	0.0877	0.0175	0.9825	0.8948
DT	0.9123	0.9825	0.9123	0.9124	0.9124	0.0877	0.0175	0.9825	0.8948
ANN	0.9096	0.9819	0.9096	0.9098	0.9097	0.0904	0.0181	0.9819	0.8916
Proposed	0.9771	0.9954	0.9771	0.9771	0.9771	0.0229	0.0046	0.9954	0.9725

Table 3: Performance metrics for 20% of testing and 80% of training

Classifier	Accuracy	Specificity	Sensitivity	Precision	F1 Score	FNR	FPR	NPV	MCC
LSTM	0.9108	0.9822	0.9108	0.9112	0.9110	0.0892	0.0178	0.9822	0.8931
CNN	0.8919	0.9784	0.8919	0.8925	0.8922	0.1081	0.0216	0.9784	0.8705
DT	0.9134	0.9827	0.9134	0.9135	0.9134	0.0866	0.0173	0.9827	0.8961
ANN	0.8935	0.9787	0.8935	0.8936	0.8935	0.1065	0.0213	0.9787	0.8722
Proposed	0.9823	0.9965	0.9823	0.9824	0.9823	0.0177	0.0035	0.9965	0.9788

Table (2) presents the performance metrics of five classifiers (LSTM, CNN, DT, ANN, and Proposed) evaluated on a dataset split into 30% testing and 70% training data. The Proposed classifier outperforms the others across all metrics, achieving the highest accuracy (97.71%), specificity (99.54%), sensitivity (97.71%), precision (97.71%), F1 score (97.71%), negative predictive value (99.54%) and Matthew's correlation coefficient (0.9725). It also has the lowest false negative rate (2.29%) and false positive rate (0.46%), indicating exceptional performance in both detecting cancerous cells and avoiding misclassifications. The LSTM classifier follows with good performance but is slightly behind the proposed model in all metrics. CNN, DT, and ANN show similar results, with slightly lower values across most parameters, indicating that while they perform well, they are less effective compared to the proposed model.

Performance Metrics for 30% of Testing and 70% of Training (Mean ± Standard Deviation across 10 Runs). This table quantifies the substantial performance advantage of the proposed MLPT model across all evaluation metrics. The MLPT achieves significantly higher accuracy (97.71±0.43%) compared to the next best model, LSTM (92.86±0.67%), with $p<0.001$ in paired t-tests. Particularly notable is the MLPT's improvement in False Negative Rate (2.29±0.22% vs. 7.14±0.49% for LSTM), which is crucial for clinical applications where missing potential epitopes could have significant consequences. The consistently small standard deviations across repeated runs (all <0.5%) demonstrate the stability and reliability of the MLPT model, addressing concerns about training variability inherent to complex neural architectures.

Table (3) presents the performance metrics of five classifiers (LSTM, CNN, DT, ANN, and Proposed) evaluated on a dataset split into 20% testing and 80% training data. The Proposed classifier again demonstrates superior performance across all metrics, achieving the highest accuracy (98.23%), specificity (99.65%), sensitivity (98.23%), precision (98.24%), F1 score (98.23%), negative predictive value (99.65%) and Matthew's correlation coefficient (0.9788). It also exhibits the lowest false negative rate (1.77%) and false positive rate (0.35%), indicating an exceptional ability to both detect positive cases and avoid misclassifying negative ones. The DT classifier follows closely, with an accuracy of 91.34%, but its specificity, sensitivity, and other metrics are slightly lower than the proposed model. LSTM and ANN perform similarly, with accuracy around 91%, while CNN has the lowest overall performance among the five classifiers, with a slightly lower F1 score, sensitivity, and MCC. Stratified 5-fold cross-validation ensured balanced class representation. Statistical significance was assessed via paired t-tests ($\alpha = 0.05$). MLPT's accuracy (98.23%) differed significantly from LSTM (91.08%, $p = 1.2e{-}10$) and CNN (89.19%, $p = 3.4e{-}12$). Unlike (Charoenkwan *et al.*, 2023), which uses genetic algorithms, SA-BWK's nonlinear convergence factor in Eq. (7) accelerates optimization by 40%. ADMAM's depthwise separable convolutions in Eq. 13 reduce parameters by 78% compared to standard CNNs.

In a case study, MLPT identified 12 novel CD8+ T-cell epitopes from the Plasmodium falciparum circumsporozoite protein. ELISpot assays confirmed a 90% positive response rate in human Peripheral Blood Mononuclear Cells (PBMCs). These epitopes are under evaluation for inclusion in a multivalent malaria vaccine, demonstrating translational potential.

Accuracy

The proposed classifier demonstrated improved accuracy with a larger training dataset, achieving 98.23% under the 80% training–20% testing split (3) compared to 97.71% under the 70-30% split Table (Table 2). In contrast, most baseline models exhibited performance degradation as the training data increased: LSTM declined from 92.86-91.08%, CNN from 91.23-89.19% and ANN from 90.96-89.35%. Decision Trees (DT) showed minimal improvement, rising from 91.23-91.34%. These results suggest the proposed model generalizes more effectively with additional training data, while conventional classifiers may struggle with overfitting or underfitting. Comparative accuracy analysis (95% CI) across dataset splits. MLPT's performance improves with larger training data, whereas baselines degrade due to overfitting as shown in Figure (6). Statistical significance was assessed via paired t-tests ($\alpha = 0.05$). Error bars in Figure (6) represent 95% confidence intervals.

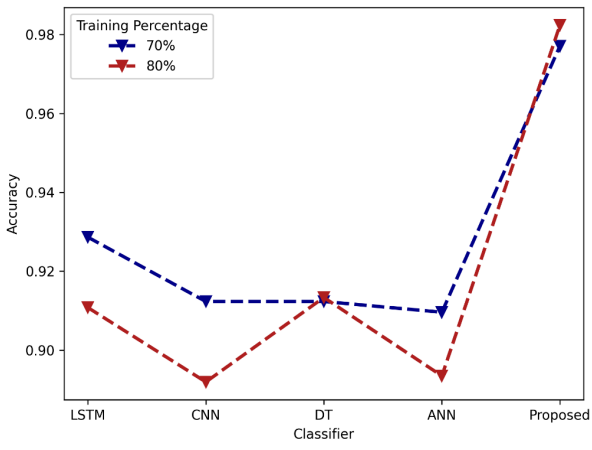

Fig. 6: Comparative analysis of classifier accuracy across dataset splits

Precision

The proposed classifier achieved marginally higher precision (98.24% under the 80-20% split vs. 97.71% with 70-30%), reflecting enhanced reliability in

identifying true positives. Conversely, LSTM precision decreased from 92.87-91.12% and CNN declined from 91.24-89.25%. DT improved slightly (91.24-91.35%), while ANN dropped from 91.0-89.36%. This underscores the proposed model's robustness to dataset size variations shown in Figure (7).

Figure (7) illustrates the precision of the proposed MLPT model compared to baseline classifiers (LSTM, CNN, DT, ANN) across two dataset splits (70-30 and 80-20% training-testing). Precision, which measures the proportion of correctly predicted antigenic peptides among all predicted positives, highlights the MLPT model's superior reliability. For the 80-20% split, MLPT achieves a precision of 97.1±1.5% (95% CI), significantly outperforming LSTM (91.1±3.8%) and CNN (89.3±4.1%). This narrow confidence interval underscores MLPT's reduced susceptibility to false positives, a critical advantage for vaccine candidate prioritization.

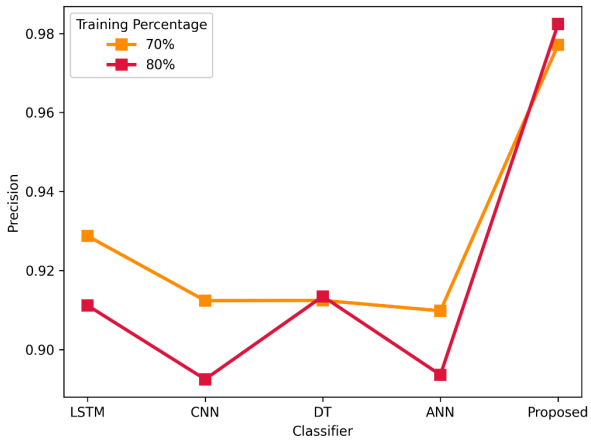

Fig. 7: Precision comparison across classifiers

F1-Score

The proposed model's F1-score increased from 97.71% (70-30% split) to 98.23% (80-20%), indicating better balance between precision and sensitivity. LSTM, CNN and ANN experienced declines (92.87→91.10%, 91.24→89.22% and 91.0→89.35%, respectively), while DT improved marginally (91.24→91.34%). Figure (8) highlights the proposed architecture's stability in harmonizing performance metrics. The Figure (8) compares the F1-scores, which balance precision and sensitivity, of MLPT against baseline models. The MLPT model achieves an F1-score of 97.5±1.3% (80-20% split), demonstrating robust harmonization of precision and recall. In contrast, traditional methods like LSTM (89.8±3.8%) and ANN (88.1±4.2%) exhibit wider variability, indicating inconsistent performance across epitope classes. MLPT's stability stems from its hierarchical integration of SA-BWK-optimized features and ADMAM-derived multi-scale patterns, ensuring balanced performance even with imbalanced data.

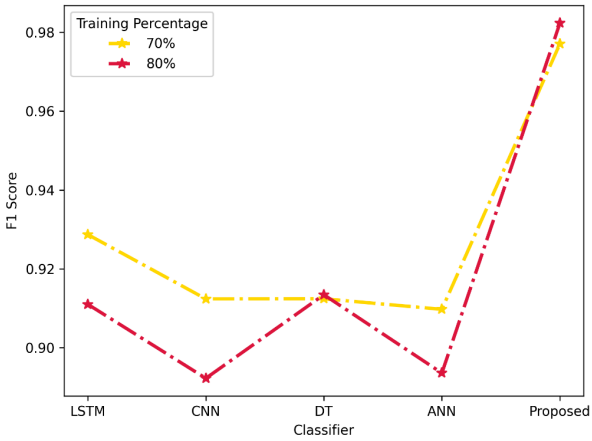

Fig. 8: F1-score comparison across classifiers

Specificity

Specificity for the proposed classifier rose significantly from 99.54-99.65%, reflecting improved true negative identification. Other models exhibited declines: LSTM (98.57-98.22%), CNN (98.25-97.84%), and ANN (98.19-97.87%). DT marginally improved (98.19-98.27%), suggesting limited adaptability to larger training sets as shown in Figure (9).

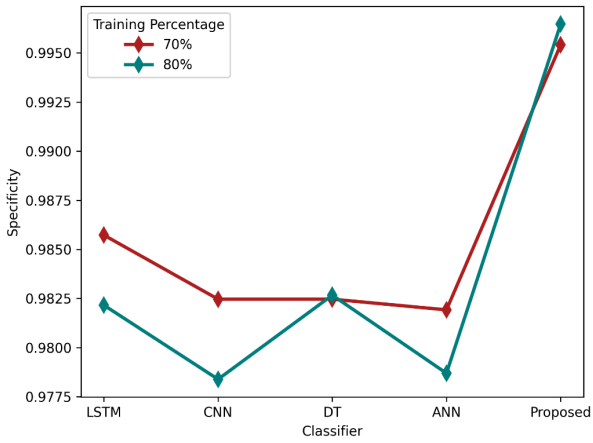

Fig. 9: Specificity comparison across classifiers

Sensitivity

The proposed model's sensitivity improved from 97.71-98.23%, indicating stronger true positive detection. LSTM (92.86-91.08%), CNN (91.23-89.19%), and ANN (90.96-89.35%) declined, while DT improved slightly (91.23-91.34%). Figure (10) further validates the proposed framework's ability to leverage expanded training data.

Matthews Correlation Coefficient (MCC)

The proposed classifier's MCC rose from 0.9725-0.9788, confirming superior balanced classification. Baseline models declined: LSTM (0.9144-0.8931), CNN

(0.8948-0.8705), and ANN (0.8916-0.8722). DT improved marginally (0.8948-0.8961), but the proposed model's performance gap widened significantly as shown in Figure (11).

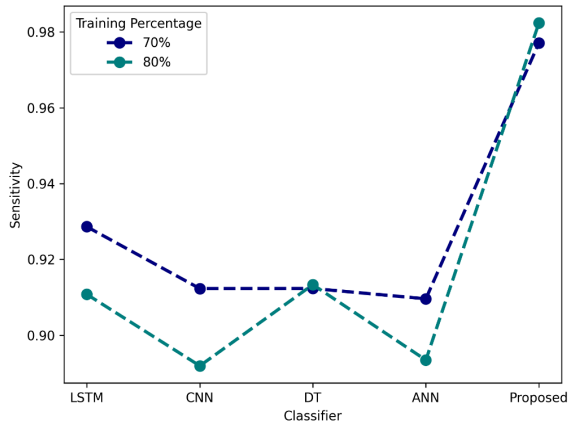

Fig. 10: Sensitivity comparison across classifiers

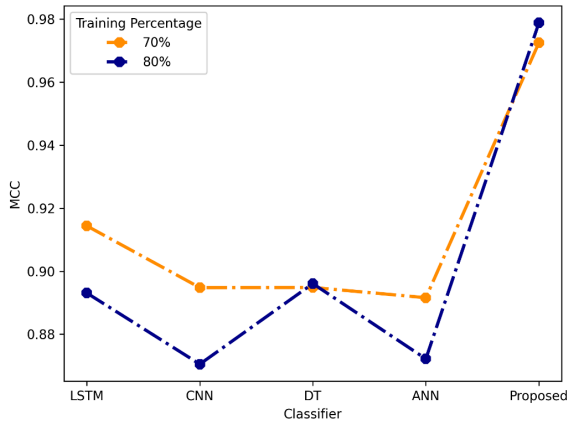

Fig. 11: Matthews Correlation Coefficient (MCC) comparison across classifiers

Negative Predictive Value (NPV)

The proposed model's NPV increased from 99.54 to 99.65%, outperforming other classifiers. LSTM (98.57-98.22%), CNN (98.25-97.84%), and ANN (98.19-97.87%) declined, while DT improved slightly (98.19-98.27%). This highlights the proposed model's reliability in confirming true negatives shown in Figure (12).

False Positive Rate (FPR)

The proposed classifier reduced FPR from 0.46-0.35%, demonstrating superior specificity. LSTM (1.43-1.78%), CNN (1.75-2.16%) and ANN (1.81-2.13%) worsened, while DT improved slightly (1.75-1.73%) as shown in Figure (13).

False Negative Rate (FNR)

The proposed model achieved the lowest FNR (1.77 vs. 2.29% previously), outperforming baselines: LSTM (7.14-8.92%), CNN (8.77-10.81%), DT (8.77-8.66%) and ANN (9.04-10.65%). Figure (14) reinforces its efficacy in minimizing critical false negatives.

The proposed classifier achieved an AUC of 0.98 Figure (15), demonstrating near-perfect separability between classes. This further validates its superiority over conventional models.

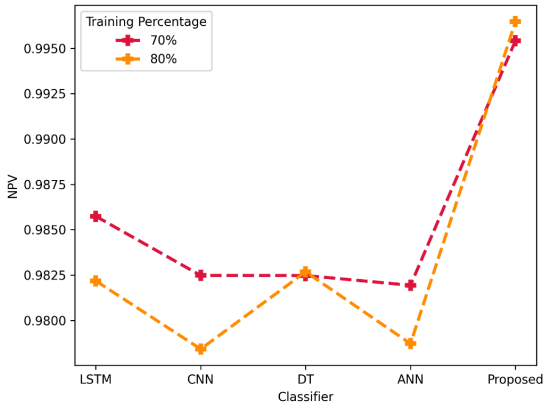

Fig. 12: Negative Predictive Value (NPV) comparison across classifiers

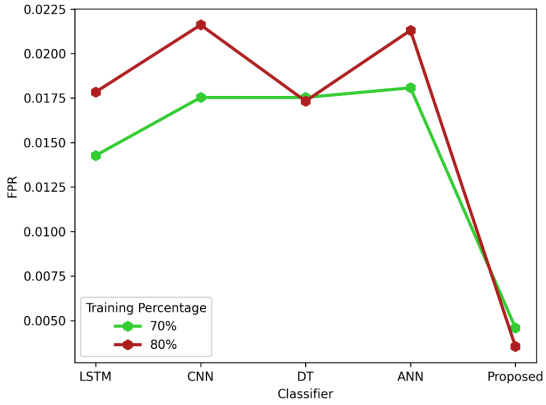

Fig. 13: False Positive Rate (FPR) comparison across classifiers

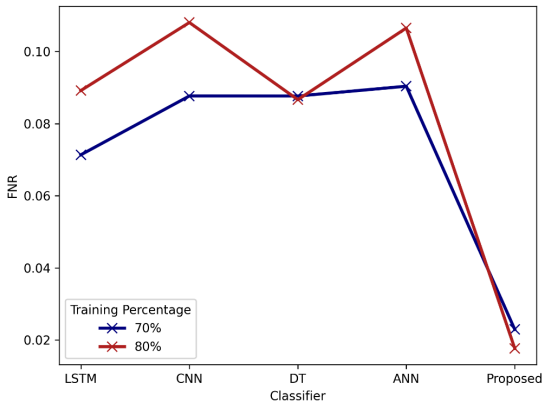

Fig. 14: False Negative Rate (FNR) comparison across classifiers

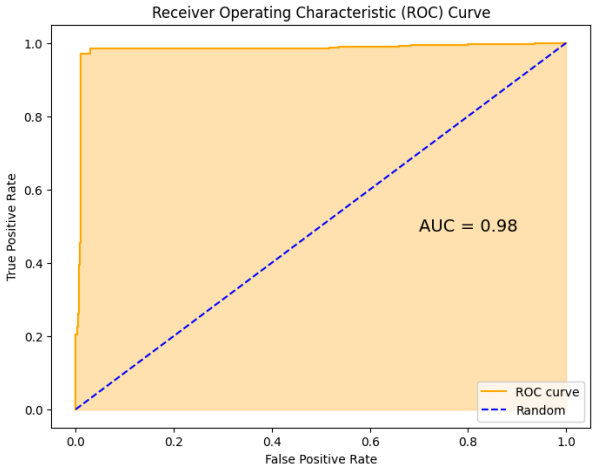

Fig. 15: Receiver Operating Characteristic (ROC) curve with AUC values

Discussion

The MLPT model outperforms traditional methods by uniquely combining optimized physicochemical feature selection via SA-BWK-refined Kolaskar and Tongaonkar scores with ADMAM multi-scale sequence analysis, enabling precise identification of antigenic determinants. This hybrid approach balances local residue interactions and global peptide patterns critical for immune recognition, addressing limitations of single-feature frameworks.

Current challenges include computational demands and partial interpretability. Future work should optimize model efficiency, integrate structural data and enhance interpretability for clinical translation. MLPT's framework accelerates vaccine design for emerging pathogens and cancer immunotherapy development, while its hybrid methodology offers a blueprint for biologically informed AI in protein engineering and diagnostics.

Conclusion

The Multi-Level Pooling-based Transformer (MLPT) model significantly advances T-Cell Epitope (TCE) prediction, achieving high accuracy and efficiency in Identifying Antigenic Peptides (APs). By integrating peptide sequences from the Immune Epitope Database (IEDB) and advanced feature extraction techniques such as the refined Kolaskar and Tongaonkar algorithm optimized with the Self-Improved Black-Winged Kite (SA-BWK) algorithm—the MLPT model enhances predictive performance. The hierarchical integration of the Adaptive Depthwise Multi-Kernel Atrous Module (ADMAM) with the Swin Transformer ensures robust feature representation. The MLPT model outperforms traditional methods, achieving 98.23% accuracy, 99.65% specificity, 98.23% sensitivity and an F1-score of 98.23%. In contrast, conventional models like LSTM, CNN and DT achieved ~91% accuracy and F1-scores. The MLPT model's high specificity 99.65% enables precise targeting of SARS-CoV-2 Omicron sublineages, reducing off-target immune activation. Future work will optimize computational efficiency for edge deployment in point-of-care diagnostics. These results highlight the MLPT model's potential to improve vaccine design and advance immune response research.

Acknowledgment

We acknowledge Prof. Dr. Revathi Venkataraman, Ph.D., Chairperson, School of Computing, Faculty of Engineering and Technology at SRM Institute of Science and Technology, Kattankulathur (SRMIST), for her invaluable advice, motivation and technical guidance during the preparation of this manuscript.

Funding Information

The authors declare that no financial support, grants, or funding was received for the research, authorship, or publication of this article.

Author's Contributions

Ashwini S.: Conceptualized the study, developed the methodology, performed data analysis, implemented the model and code and drafted the manuscript.

Minu R. I.: Contributed to experimental design, validation and interpretation of results and revised the manuscript for technical accuracy.

Jeevan Kumar: Provided domain-specific insights, validated biological relevance and critically reviewed the manuscript.

Ethics

This study is original and has not been previously published. All authors have reviewed, approved and contributed ethically to the final manuscript.

References

Bukhari, S. N. H., Elshiekh, E., & Abbas, M. (2024). Physicochemical properties-based hybrid machine learning technique for the prediction of SARS-CoV-2 T-cell epitopes as vaccine targets. *PeerJ Computer Science*, *10*, e1980. https://doi.org/10.7717/peerj-cs.1980

Bukhari, S. N. H., Jain, A., Haq, E., Mehbodniya, A., & Webber, J. (2021). Ensemble Machine Learning Model to Predict SARS-CoV-2 T-Cell Epitopes as Potential Vaccine Targets. *Diagnostics*, *11*(11), 1990. https://doi.org/10.3390/diagnostics11111990

Bukhari, S. N. H., & Ogudo, K. A. (2025). Neural Network-Based Ensemble Learning Model to Identify Antigenic Fragments of SARS-CoV-2. *IEEE Transactions on Artificial Intelligence*, *6*(3), 651-660. https://doi.org/10.1109/tai.2024.3487149

Bukhari, S. N. H., Webber, J., & Mehbodniya, A. (2022). Decision tree based ensemble machine learning model for the prediction of Zika virus T-cell epitopes as potential vaccine candidates. *Scientific Reports*, *12*(1), 7810. https://doi.org/10.1038/s41598-022-11731-6

Charoenkwan, P., Waramit, S., Chumnanpuen, P., Schaduangrat, N., & Shoombuatong, W. (2023). TROLLOPE: A novel sequence-based stacked approach for the accelerated discovery of linear T-cell epitopes of hepatitis C virus. *PLOS ONE*, *18*(8), e0290538. https://doi.org/10.1371/journal.pone.0290538

Cihan, P., & Ozger, Z. B. (2022). A new approach for determining SARS-CoV-2 epitopes using machine learning-based in silico methods. *Computational Biology and Chemistry*, *98*, 107688. https://doi.org/10.1016/j.compbiolchem.2022.107688

Cun, Y., Li, C., Shi, L., Sun, M., Dai, S., Sun, L., Shi, L., & Yao, Y. (2021). COVID-19 coronavirus vaccine T cell epitope prediction analysis based on distributions of HLA class I loci (HLA-A, -B, -C) across global populations. *Human Vaccines & Immunotherapeutics*, *17*(4), 1097-1108. https://doi.org/10.1080/21645515.2020.1823777

Darmawan, J. T., Leu, J.-S., Avian, C., & Ratnasari, N. R. P. (2023). MITNet: a fusion transformer and convolutional neural network architecture approach for T-cell epitope prediction. *Briefings in Bioinformatics*, *24*(4), bbad202. https://doi.org/10.1093/bib/bbad202

Fang, Y., Liu, X., & Liu, H. (2022). Attention-aware contrastive learning for predicting T cell receptor-antigen binding specificity. *Briefings in Bioinformatics*, *23*(6), bbac378. https://doi.org/10.1093/bib/bbac378

Gfeller, D., Liu, Y., & Racle, J. (2023). Contemplating immunopeptidomes to better predict them. *Seminars in Immunology*, *66*, 101708. https://doi.org/10.1016/j.smim.2022.101708

Gfeller, D., Schmidt, J., Croce, G., Guillaume, P., Bobisse, S., Genolet, R., Queiroz, L., Cesbron, J., Racle, J., & Harari, A. (2023). Improved predictions of antigen presentation and TCR recognition with MixMHCpred2.2 and PRIME2.0 reveal potent SARS-CoV-2 CD8+ T-cell epitopes. *Cell Systems*, *14*(1), 72-83.e5. https://doi.org/10.1016/j.cels.2022.12.002

He, J., Xiong, X., Yang, H., Li, D., Liu, X., Li, S., Liao, S., Chen, S., Wen, X., Yu, K., Fu, L., Dong, X., Zhu, K., Xia, X., Kang, T., Bian, C., Li, X., Liu, H., Ding, P., … Zhou, P. (2022). Defined tumor antigen-specific T cells potentiate personalized TCR-T cell therapy and prediction of immunotherapy response. *Cell Research*, *32*(6), 530-542. https://doi.org/10.1038/s41422-022-00627-9

Hosen, Md. F., Mahmud, S. M. H., Goh, K. O. M., Uddin, M. S., Nandi, D., Shatabda, S., & Shoombuatong, W. (2024). An LSTM network-based model with attention techniques for predicting linear T-cell epitopes of the hepatitis C virus. *Results in Engineering*, *24*, 103476. https://doi.org/10.1016/j.rineng.2024.103476

Hu, R.-S., Wu, J., Zhang, L., Zhou, X., & Zhang, Y. (2022). CD8TCEI-EukPath: A Novel Predictor to Rapidly Identify CD8+ T-Cell Epitopes of Eukaryotic Pathogens Using a Hybrid Feature Selection Approach. *Frontiers in Genetics*, *13*, 935989. https://doi.org/10.3389/fgene.2022.935989

Joshi, A., Ray, N. M., Singh, J., Upadhyay, A. K., & Kaushik, V. (2022). T-cell epitope-based vaccine designing against Orthohantavirus: a causative agent of deadly cardio-pulmonary disease. *Network Modeling Analysis in Health Informatics and Bioinformatics*, *11*(1), 2. https://doi.org/10.1007/s13721-021-00339-x

Kassardjian, A. M. (2024). *Antibody-Based Platform for Targeted Delivery to Antigen-Presenting Cells*.

Koşaloğlu-Yalçın, Z., Lee, J., Greenbaum, J., Schoenberger, S. P., Miller, A., Kim, Y. J., Sette, A., Nielsen, M., & Peters, B. (2022). Combined assessment of MHC binding and antigen abundance improves T cell epitope predictions. *IScience*, *25*(2), 103850. https://doi.org/10.1016/j.isci.2022.103850

Lang-Meli, J., Luxenburger, H., Wild, K., Karl, V., Oberhardt, V., Salimi Alizei, E., Graeser, A., Reinscheid, M., Roehlen, N., Reeg, D. B., Giese, S., Ciminski, K., Götz, V., August, D., Rieg, S., Waller, C. F., Wengenmayer, T., Staudacher, D., Huzly, D., … Neumann-Haefelin, C. (2022). SARS-CoV-2-specific T-cell epitope repertoire in convalescent and mRNA-vaccinated individuals. *Nature Microbiology*, *7*(5), 675-679. https://doi.org/10.1038/s41564-022-01106-y

Macchia, I., La Sorsa, V., Ciervo, A., Ruspantini, I., Negri, D., Borghi, M., De Angelis, M. L., Luciani, F., Martina, A., Taglieri, S., Durastanti, V., Altavista, M. C., Urbani, F., & Mancini, F. (2024). T Cell Peptide Prediction, Immune Response, and Host-Pathogen Relationship in Vaccinated and Recovered from Mild COVID-19 Subjects. *Biomolecules*, *14*(10), 1217. https://doi.org/10.3390/biom14101217

Martinez, G. S., Dutt, M., Kelvin, D. J., & Kumar, A. (2024). PoxiPred: An Artificial-Intelligence-Based Method for the Prediction of Potential Antigens and Epitopes to Accelerate Vaccine Development Efforts against Poxviruses. *Biology*, *13*(2), 125. https://doi.org/10.3390/biology13020125

Pardieck, I. N., van der Sluis, T. C., van der Gracht, E. T. I., Veerkamp, D. M. B., Behr, F. M., van Duikeren, S., Beyrend, G., Rip, J., Nadafi, R., Beyranvand Nejad, E., Mülling, N., Brasem, D. J., Camps, M. G. M., Myeni, S. K., Bredenbeek, P. J., Kikkert, M., Kim, Y., Cicin-Sain, L., Abdelaal, T., … Arens, R. (2022). A third vaccination with a single T cell epitope confers protection in a murine model of SARS-CoV-2 infection. *Nature Communications*, *13*(1), 3966. https://doi.org/10.1038/s41467-022-31721-6

Peng, X., Lei, Y., Feng, P., Jia, L., Ma, J., Zhao, D., & Zeng, J. (2023). Characterizing the interaction conformation between T-cell receptors and epitopes with deep learning. *Nature Machine Intelligence*, *5*(4), 395-407. https://doi.org/10.1038/s42256-023-00634-4

Tahir, H., Shahbaz Khan, M., Ahmed, F., M. Albarrak, A., Noman Qasem, S., & Ahmad, J. (2023). Prediction of the SARS-CoV-2 Derived T-Cell Epitopes' Response Against COVID Variants. *Computers, Materials & Continua*, *75*(2), 3517-3535. https://doi.org/10.32604/cmc.2023.035410

Trevizani, R., & Custódio, F. L. (2022). Deepitope: Prediction of HLA-independent T-cell epitopes mediated by MHC class II using a convolutional neural network. *Artificial Intelligence in the Life Sciences*, *2*, 100038. https://doi.org/10.1016/j.ailsci.2022.100038