Journal of Advanced College of Engineering and Management

Comparative Analysis of LSTM, Transformer, and TCN in Predicting Peptide-MHC Class II Binding Affinity

[1]Dhiraj Pyakurel, [1]Aayush Dangol, [1]Abiral Bhattarai, [1]Dipesh Thapa

[1]*Department of Electronics and Computer Engineering, Advanced College of Engineering and Management, Kathmandu, Nepal*

Email: dhiraj@acem.edu.np

DOI: 10.3126/jacem.v11i1.84250

Abstract

Accurate prediction of Major Histocompatibility Complex (MHC) class II-peptide binding affinity remains a critical challenge in immunotherapy and vaccine development due to the complexity of molecular interactions and allele-specific binding preferences. While existing computational approaches have shown promise, identifying the optimal deep learning architecture for this task remains an open question. This study addresses this gap by systematically comparing three state-of-the-art deep learning architectures: Long Short-Term Memory Networks (LSTM), Transformers, and Temporal Convolutional Networks (TCN) using the IEDB2016 dataset containing 134,281 MHC class II-peptide binding records. Our comprehensive evaluation employed five-fold cross-validation and ensemble modeling to assess both regression and classification performance. Results demonstrate that TCN consistently outperforms competing architectures, achieving superior regression metrics with R^2 of 0.6208 (vs. LSTM: 0.5923, Transformer: 0.5706) and classification performance with AUROC of 0.8766 (vs. LSTM: 0.8736, Transformer: 0.8707). Cross-validation confirmed TCN's robustness (average R^2: 0.6229), while ensemble methods further validated its superiority (R^2: 0.6805). These findings establish TCN as the most effective architecture for peptide-MHC binding prediction, offering significant implications for computational immunology and therapeutic design.

Keywords*—Artificial Intelligence, Deep Learning, IEDB, LSTM, Machine Learning, TCN*

1. INTRODUCTION

Major Histocompatibility Complex (MHC) alleles are central to immune function, presenting peptides to T-cells and initiating immune responses. MHC-I alleles present intracellular peptides to CD8+ T-cells, while MHC-II alleles present extracellular peptides to CD4+ helper T-cells. Accurate prediction of peptide–MHC binding affinity is essential for immunotherapy and vaccine design but remains challenging due to allele-specific binding preferences, peptide variability, and the structural complexity of peptide–MHC interactions.

Despite the advancements in computational models and the growing availability of large biological datasets, predicting MHC-peptide binding affinity remains challenging. MHC alleles exhibit unique binding preferences, making it difficult to develop a generalized model that performs well across all allele types. Additionally,

the structural and sequential variability of peptides complicates the creation of a single model capable of accurately predicting binding affinity across diverse peptides. While ML approaches have demonstrated promising results, identifying the most effective DL architecture remains an open research question.

Computational models for MHC-peptide binding can be categorized into two types: allele-specific and pan-specific. Allele-specific models train separate models for every MHC allele, achieving higher accuracy but limited generalization. In contrast, pan-specific models aim for better generalization by training using data from multiple alleles. However, pan-specific models often struggle to capture allele-specific binding preferences, resulting in a trade-off between accuracy and flexibility.

Recent years have seen increasing development of various DL models to improve MHC-peptide binding affinity predictions. NetMHCpan-4.0, a neural network-based pan-specific model, has demonstrated strong performance in predicting binding affinity for both MHC-I and MHC-II peptides [7]. MHCAttnNet, another DL model, employs bidirectional LSTM and cross-attention mechanisms to enhance predictions [15]. DeepMHCII, a Convolutional Neural Network (CNN)-based model, has outperformed traditional ML models in MHCII-peptide binding affinity prediction [20]. Models based on Transformers such as BERTMHC and TransHLA leverage self-attention mechanisms to improve performance across a wide range of MHC alleles [3], [13]. A recent advancement in this field is RPEMHC, a CNN and LSTM-based model that introduces Residue-Residue Pair Encoding (RPE) to improve binding affinity predictions. Unlike conventional methods that encode MHC and peptide sequences separately, RPEMHC uses a residue-residue interaction matrix, demonstrating enhanced performance across MHC-I and MHC-II alleles [19].

The main contributions of this article are summarized as follows:

- **Systematic architectural comparison**: We provide the first comprehensive comparison of LSTM, Transformer, and TCN architectures for MHC-peptide binding prediction, with particular emphasis on the underexplored TCN approach.
- **Robust evaluation framework**: We employ rigorous five-fold cross-validation and ensemble modeling to ensure reliable performance assessment across diverse MHC alleles and peptide sequences, advancing methodological standards in the field for vaccine design, immunotherapy, and peptide-based drug discovery applications.

Through this analysis, we discovered the most effective model for the prediction of binding affinity of MHC peptides, contributing to the larger goal of advancing regulatory immunology. By elucidating the strengths and weaknesses of these DL architectures, this work aims to support progress in vaccine design, immunotherapy, and peptide-based drug discovery.

2. LITERATURE REVIEW

Recent years have seen increasing development of various deep learning models to improve MHC-peptide binding affinity predictions. NetMHCpan-4.0, a neural network-based pan-specific model, has demonstrated strong performance in

predicting binding affinity for both MHC-I and MHC-II peptides, though it struggles with allele-specific optimization and underperforms for rare alleles with limited training data. [1]. MHCAttnNet, another deep learning model, employs bidirectional LSTM and cross-attention mechanisms to enhance predictions [2]. While effective at modeling sequence relationships, MHCAttnNet suffers from computational complexity. DeepMHCII, a Convolutional Neural Network (CNN)-based model, has outperformed traditional machine learning models in MHCII-peptide binding affinity prediction [3]. Convolutional models offer computational efficiency and local feature extraction capabilities but miss long-range dependencies crucial for binding prediction.

Models based on Transformers such as BERTMHC and TransHLA leverage self-attention mechanisms to improve performance across a wide range of MHC alleles [4], [5]. These models excel at capturing global sequence relationships but require substantial computational resources and may struggle with shorter sequences typical in this domain. A recent advancement in this field is RPEMHC, a CNN and LSTM-based model that introduces Residue-Residue Pair Encoding (RPE) to improve binding affinity predictions. Unlike conventional methods that encode MHC and peptide sequences separately, RPEMHC uses a residue-residue interaction matrix, demonstrating enhanced performance across MHC-I and MHC-II alleles [6]. While this innovation improved performance across MHC-I and MHC-II alleles, the method increases model complexity and computational requirements.

Computational models broadly fall into allele-specific and pan-specific categories. Allele-specific models achieve higher accuracy through specialized training but lack generalization capability and require extensive data for each allele. Pan-specific models offer better generalization by training on multiple alleles but often struggle to capture allele-specific binding preferences, resulting in a fundamental accuracy-flexibility trade-off that remains unresolved in current approaches.

Despite these advances, systematic comparison of different architectural paradigms remains limited, particularly for Temporal Convolutional Networks, which have shown promise in other sequence modeling domains but remain unexplored for MHC-peptide binding prediction.

3. METHODOLOGY

A. Proposed Approach

We propose a systematic comparative framework to evaluate three distinct deep learning paradigms for peptide-MHC class II binding affinity prediction. Our approach integrates recurrent (LSTM), attention-based (Transformer), and convolutional (TCN) architectures within a unified evaluation framework, employing consistent preprocessing, training protocols, and assessment metrics to ensure fair comparison.

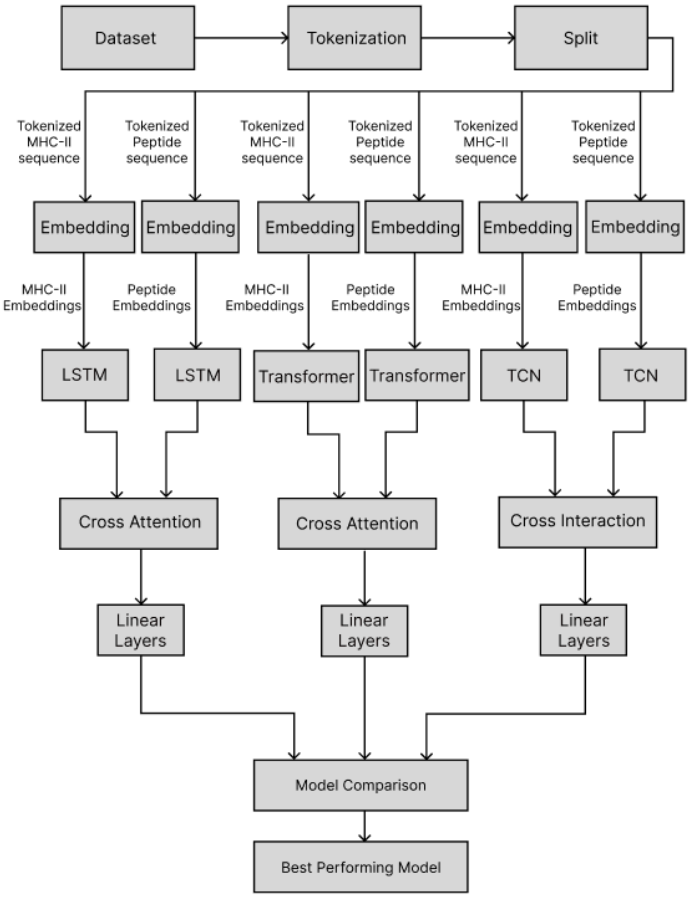

FIG. 1: Methodological framework for comparing the performance of DL techniques in Peptide-MHC Class II binding affinity prediction

B. Dataset and Data Preparation

We utilized the Immune Epitope Database 2016 (IEDB2016) dataset, containing 134,281 entries representing binding affinity values between MHC class II molecules and peptides. The dataset encompasses 80 different MHC class II alleles: 36 HLA-DR, 27 HLA-DQ, 9 HLA-DP, and 8 H-2 molecules.

Original IC50 affinity values (nanomolar units) were normalized using:

$$y = 1 - log(IC50)/log(50000) \quad (1)$$

For classification tasks, we applied a 500nM threshold, classifying peptides below this value as binders and those equal to or above as non-binders.

C. Tokenization and Embeddings

We employed character-level tokenization using the ESM-2 tokenizer, converting MHC and peptide sequences into individual amino acid tokens with unique numerical identifiers. The vocabulary size was 33, including amino acid and special tokens. Self-learned embeddings were used to ensure domain-specific

representation learning rather than relying on potentially mismatched pretrained embeddings.

D. Deep Learning Architectures

1) LSTM Architecture

We implemented the LSTM architecture as described by Hochreiter and Schmidhuber [4], utilizing the standard formulation with forget, input, and output gates for addressing vanishing gradient problems in sequential data. The outputs of the LSTM were then passed to a fully connected layer for prediction. Optimal hyperparameters were embedding dimension of 512, 128 LSTM hidden units, 3 layers, dropout of 0.2, learning rate of 0.0001, and weight decay of 0.0001.

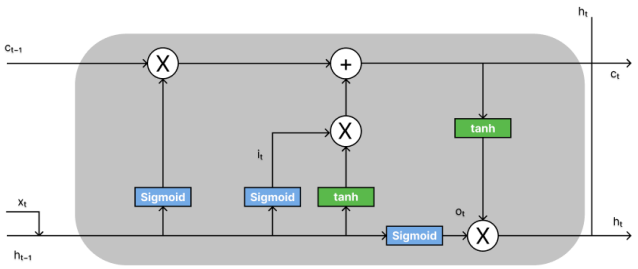

FIG. 2: LSTM architecture illustrating the cell state (C), hidden state (h), output gate (o), input vectors (i), and hyperbolic tangent activation (tanh)

The mathematical formulation of LSTM gates are:

Forget gate: $ft = \sigma(wf \cdot [ht - 1, xt] + bf)$

Input gate: $it = \sigma(wi \cdot [ht - 1, xt] + bi)$

Update vector: $\hat{c} = tanh(wc \cdot [ht - 1, xt] + bc)$

Cell state: $ft \odot Ct - 1 + it \odot \hat{c}t$

Output gate: $\sigma(wo \cdot [ht - 1, xt] + bo)$

2) Transformer Architecture

The Transformer implementation followed Vaswani et al. [14], employing self-attention mechanisms and positional encoding for parallel sequence processing. Optimal hyperparameters were embedding dimension of 256, 3 transformer layers, dropout of 0.1, learning rate of 0.0001, weight decay of 0.0001, and 8 attention heads.

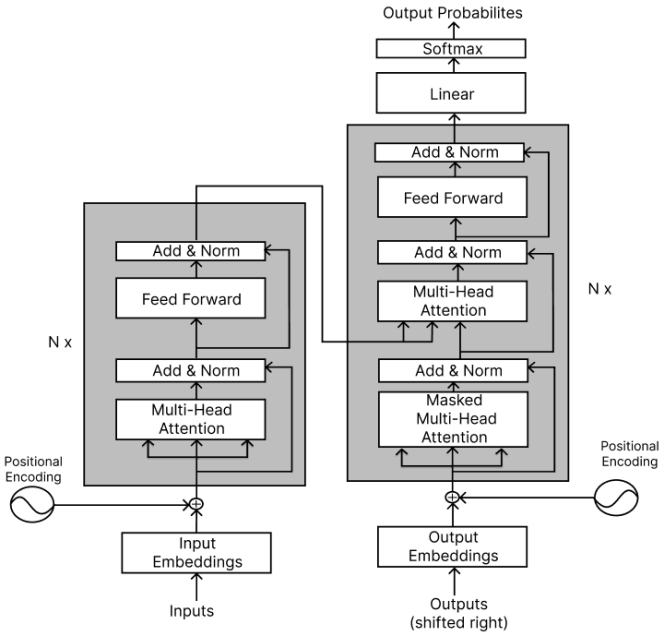

FIG. 3: Architecture of the Transformer model

The self-attention mechanism is calculated using:

$$Attention(Q,\ K,\ V)\ =\ softmax(QK^{\mathrm{T}}/\sqrt{dk})V \quad (2)$$

Multi-head attention is formulated as:

$$MultiHead(Q,\ K,\ V)\ =\ Concat(h1,\ \ldots,\ hn)Wo \quad (3)$$

Positional encoding equations:

$$PE(pos,\ 2i)\ =\ sin(pos/1000\hat{}(2i/dmodel)) \quad (4)$$

$$PE(pos,\ 2i\ +\ 1)\ =\ cos(pos/1000\hat{}(2i/dmodel)) \quad (5)$$

3) TCN Architecture

We implemented TCN following Bai et al. [1], utilizing one-dimensional causal convolution with dilation and residual connections.

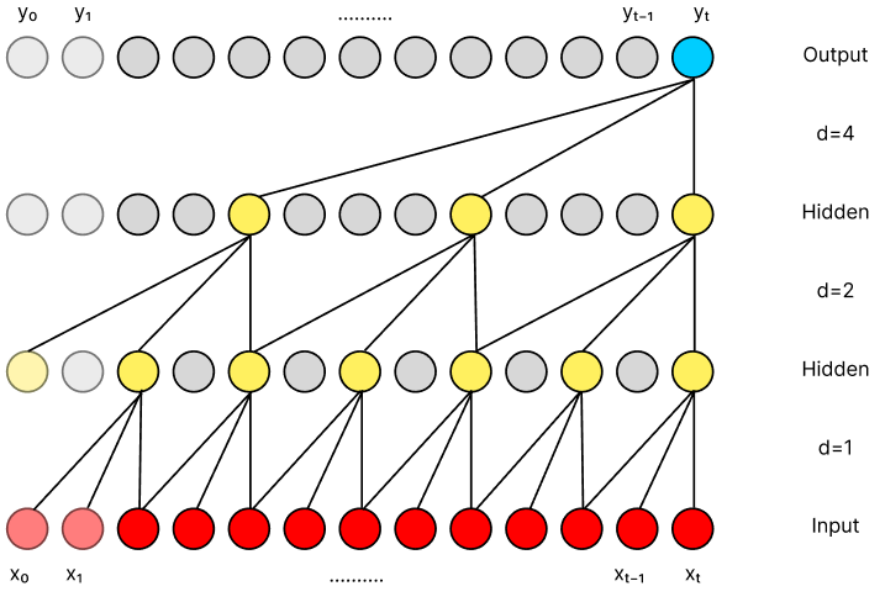

FIG. 4: Architecture of the TCN model

The dilated causal convolution is formulated as:

$$yt = \Sigma(i = 1\ to\ k)\ wi \cdot xt - d{\cdot}i \quad (6)$$

where yt is the output at time t, wi represents filter weights, xt is input at time t, d is the dilation rate, and k is the filter size. Optimal hyperparameters included: embedding dimension of 256, 3 TCN layers, dropout of 0.1, learning rate of 0.0001, kernel size of 3, and weight decay of 0.001.

E. Interaction Mechanisms

1) Cross Attention

Cross attention mechanism works similarly to the self attention mechanism but instead of calculating the relationship for tokens of a same sequence, cross attention calculates the relationship between tokens of two different sequences. We used cross attention for the Transformer and LSTM models.

2) Cross Interaction

The cross interaction mechanism models the interaction between two sequences. The MHC vectors and peptide vectors from the TCN encoder were concatenated and passed onto the cross interaction layer where 1D convolution was applied to capture the interaction between the sequences.

F. Evaluation Metrics

For regression tasks, we used R-square (R^2), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE).

$$R^2 = [(\Sigma(Bi' - B')(Bi - B)) / (\sqrt{\Sigma(Bi' - B')^2}\sqrt{\Sigma(Bi - B)^2})]^2 \quad (7)$$

$$RMSE = \sqrt{[\Sigma(Bi' - Bi)^2 / n]} \quad (8)$$

$$MAE = (1/n)\,\Sigma|Bi - Bi'| \quad (9)$$

$$MSE = (1/n)\,\Sigma(Bi - Bi')^2 \quad (10)$$

For classification tasks, we employed Area Under the Receiver Operating Characteristic Curve (AUROC), F1 score, Precision, Recall, and Area Under the Precision-Recall Curve (AUCPR).

$$TPR = TP/(TP + FN) \quad (11)$$

$$FPR = FP/(FP + TN) \quad (12)$$

$$Precision = TP/(TP + FP) \quad (13)$$

$$Recall = TP/(TP + FN) \quad (14)$$

$$F1 - score = 2 \cdot (Precision \cdot Recall)/(Precision + Recall) \quad (15)$$

4. RESULTS AND DISCUSSION

A) Regression Metrics

The test metrics for the regression models showed that the TCN model outperformed the others, achieving the highest R^2 (0.6208) and the lowest MSE (0.0260), RMSE (0.1614), and MAE (0.1229), indicating superior predictive accuracy and robustness. The LSTM and Transformer models showed competitive performance but were less accurate, with higher errors and lower R^2 values.

Table 1: Test Metrics of LSTM, Transformer, and TCN Models

Model	R^2	MSE	RMSE	MAE
LSTM	0.5923	0.0280	0.1674	0.1279
Transformer	0.5706	0.0295	0.1717	0.1717
TCN	0.6208	0.0260	0.1614	0.1229

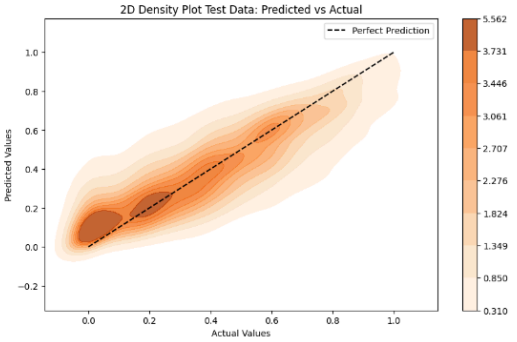

FIG. 5: 2-D Density Plot: Actual vs Predicted for TCN Predictions

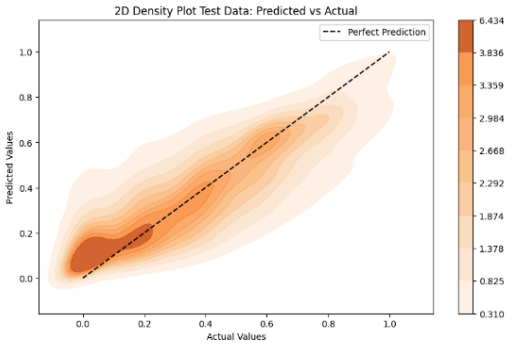

FIG. 6: 2-D Density Plot: Actual vs Predicted for LSTM Predictions

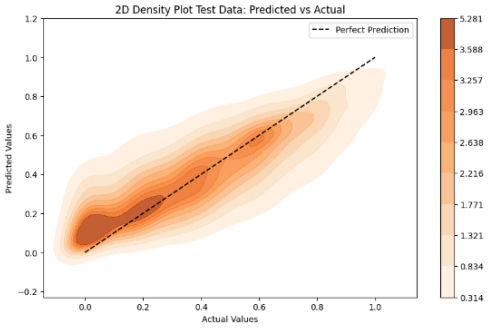

FIG. 7: 2-D Density Plot: Actual vs Predicted for Transformer Predictions

The 5-fold cross-validation results demonstrate the robustness and generalizability of the models. The TCN model consistently outperformed the others, achieving the highest average R^2 (0.6229) and the lowest average MSE (0.0258), RMSE (0.1606), and MAE (0.1208). The ensemble model results further validate the TCN model's performance, with the highest R^2 (0.6805) and the lowest MSE (0.0220), RMSE (0.1482), and MAE (0.1118).

B) Classification Metrics

The classification performance evaluation showed that the TCN model achieves the highest AUROC (0.8766) and AUCPR (0.8462), indicating strong overall predictive capability. LSTM outperforms others in F1-score (0.7603) and recall (0.8117), suggesting better sensitivity. The Transformer model balances precision (0.7309) and recall (0.7844).

Table 2: Classification Performance on Test Data

Model	AUROC	F1	Precision	Recall	AUCPR
LSTM	0.8736	0.7603	0.7150	0.8117	0.8406
Transformer	0.8707	0.7567	0.7309	0.7844	0.8365
TCN	0.8766	0.7498	0.7693	0.7312	0.8462

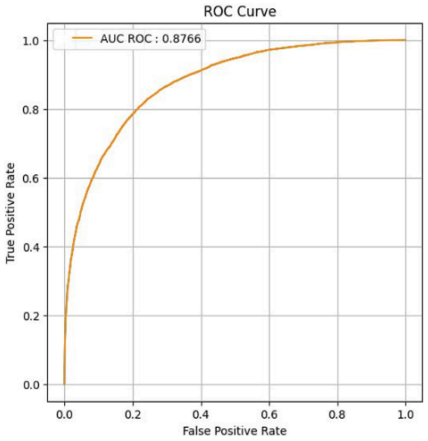

FIG. 8: ROC curve for TCN

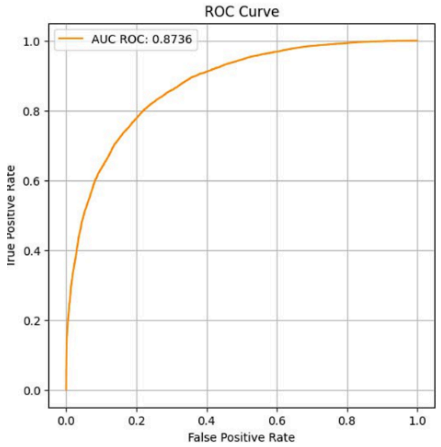

FIG. 9: ROC curve for LSTM

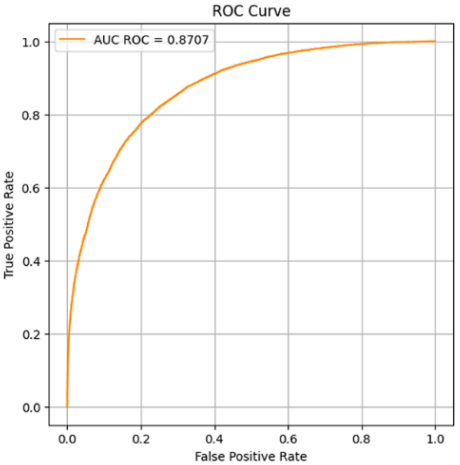

FIG. 10: ROC curve for Transformer

The 5-fold cross-validation results for the classification task demonstrate that the TCN model consistently achieved the highest AUROC values across all folds. The ensemble models results show that the TCN model achieved the highest AUROC (0.8997), F1 (0.7859), Precision (0.7645), and Recall (0.8085), outperforming the LSTM and Transformer models across all metrics.

5. CONCLUSIONS

This systematic comparison of LSTM, Transformer, and TCN architectures for peptide-MHC class II binding affinity prediction establishes TCN as the superior approach for both regression and classification tasks. TCN's consistent outperformance across all evaluation metrics, confirmed through rigorous cross-validation and ensemble modeling, addresses the critical need for reliable computational tools in immunological research.

The superior performance of TCN can be attributed to its ability to capture long-range dependencies through dilated causal convolutions while maintaining computational efficiency through parallel processing. Unlike sequential processing in LSTM or the computational overhead of multi-head attention in Transformers, TCN effectively balances model complexity with predictive accuracy.

These findings have significant implications for computational immunology, offering researchers a reliable tool for vaccine design, immunotherapy development, and peptide-based drug discovery. The established superiority of TCN architecture provides a foundation for future developments in MHC-peptide binding prediction.

SUGGESTIONS AND RECOMMENDATIONS

Future research should investigate the integration of structural information with TCN-based sequence modeling to further enhance prediction accuracy. Additionally, exploring ensemble methods that leverage TCN's strengths alongside complementary architectures could yield improved results.

The development of allele-specific fine-tuning strategies for TCN architecture represents a promising direction for addressing the accuracy-generalization trade-off. Furthermore, expanding evaluation to include more recent binding affinity datasets and exploring applications to related immunological prediction tasks such as T-cell epitope prediction would strengthen the field's computational foundations.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Advanced College of Engineering and Management for providing computational resources and support for this research work.

REFERENCES

[1] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," in Advances in Neural Information Processing Systems, vol. 31, 2018.
[2] P. J. Ballester and J. B. O. Mitchell, "A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking," Bioinformatics, vol. 26, no. 9, pp. 1169–1175, 2010.
[3] J. Cheng, K. Bendjama, K. Rittner, and B. Malone, "Bertmhc: improved mhc–peptide class ii interaction prediction with transformer and multiple instance learning," Bioinformatics, vol. 37, no. 22, pp. 4172–4179, 2021.
[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[5] K. K. Jensen, M. Andreatta, P. Marcatili, S. Buus, J. A. Greenbaum, Z. Yan, A. Sette, B. Peters, and M. Nielsen, "Improved methods for predicting peptide binding affinity to mhc class ii molecules," Immunology, vol. 154, no. 3, pp. 394–406, 2018.
[6] J. Jiménez, M. Škalič, G. Martínez-Rosell, and G. De Fabritiis, "Kdeep: Protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks," Journal of Chemical Information and Modeling, vol. 58, no. 2, pp. 287–296, 2018.

[7] V. Jurtz, S. Paul, M. Andreatta, P. Marcatili, B. Peters, and M. Nielsen, "Netmhcpan-4.0: improved peptide–mhc class i interaction predictions integrating eluted ligand and peptide binding affinity data," The Journal of Immunology, vol. 199, no. 9, pp. 3360–3368, 2017.
[8] G. B. Li, L. L. Yang, W. J. Wang, L. L. Li, and S. Y. Yang, "Id-score: A new empirical scoring function based on a comprehensive set of descriptors related to protein–ligand interactions," Journal of Chemical Information and Modeling, vol. 53, no. 3, pp. 592–600, 2013.
[9] Y. Li, M. A. Rezaei, C. Li, X. Li, and D. Wu, "Deepatom: A framework for protein-ligand binding affinity prediction," in 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2019, pp. 288–291.
[10] H. Öztürk, A. Özgür, and E. Özkırımlı, "Deepdta: deep drug–target binding affinity prediction," Bioinformatics, vol. 34, no. 17, pp. i821–i829, 2018.
[11] T. Rose, N. Monti, N. Anand, and T. Shen, "Plapt: Protein-ligand binding affinity prediction using pretrained transformers," bioRxiv, pp. 2024–02, 2024.
[12] F. Soleymani, E. Paquet, H. L. Viktor, W. Michalowski, and D. Spinello, "Protinteract: A deep learning framework for predicting protein–protein interactions," Computational and Structural Biotechnology Journal, vol. 21, pp. 1324–1348, 2023.
[13] L. Tianchi, X. Wang, W. Nie, M. Huo, and L. Shuaicheng, "Transhla: A hybrid transformer model for hla-presented epitope detection," GigaScience, 2025.
[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems, vol. 30, 2017.
[15] G. Venkatesh, A. Grover, G. Srinivasaraghavan, and S. Rao, "Mhcattnnet: predicting mhc-peptide bindings for mhc alleles classes i and ii using an attention-based deep neural model," Bioinformatics, vol. 36, no. Supplement_1, pp. i399–i406, 2020.

[16] R. Vita, J. A. Overton, J. A. Greenbaum, J. Ponomarenko, J. D. Clark, J. R. Cantrell, D. K. Wheeler, J. L. Gabbard, D. Hix, A. Sette, et al., "The immune epitope database (iedb) 3.0," Nucleic Acids Research, vol. 43, no. D1, pp. D405–D412, 2015.
[17] H. Wang, "Prediction of protein–ligand binding affinity via deep learning models," Briefings in Bioinformatics, vol. 25, no. 2, pp. bbae081, 2024.
[18] R. Wang, J. Jin, Q. Zou, K. Nakai, and L. Wei, "Predicting protein–peptide binding residues via interpretable deep learning," Bioinformatics, vol. 38, no. 13, pp. 3351–3360, 2022.
[19] X. Wang, T. Wu, Y. Jiang, T. Chen, D. Pan, Z. Jin, J. Xie, L. Quan, and Q. Lyu, "Rpemhc: improved prediction of mhc–peptide binding affinity by a deep learning approach based on residue–residue pair encoding," Bioinformatics, vol. 40, 2024.
[20] R. You, W. Qu, H. Mamitsuka, and S. Zhu, "Deepmhcii: a novel binding core-aware deep interaction model for accurate mhc-ii peptide binding affinity prediction," Bioinformatics, vol. 38, no. Supplement_1, pp. i220–i228, 2022.