Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 7s (2024)

Integrating Nonlinear Dynamics and Statistical Methods in Deep Learning Models for Biochemical Component Analysis

[1]**Swamydoss D,** Adhiyamaan College of Engineering, Hosur, India**.** swamyasir@gmail.com
[2]**O.A.Sridevi,** Assistant Professor, Department of Science and Humanities, Karpagam Institute of Technology, Coimbatore, Tamilnadu, India. sridevi6123@gmail.com
[3]**Sandhya S,** Assistant Professor, Department of CSE, JCT College of Engineering and Technology, Coimbatore, India. sandhya9230@gmail.com
[4]**Mulugeta Tesema,** Associate Professor, Department of Chemistry (Analytical), College of Natural and Computational Sciences, Dambi Dollo University, Dambi Dollo, Ethiopia. efa.ntust@gmail.com
[5]**A.Shalini,** Assistant Professor, Department of CSE, Karpagam Academy of Higher Education, Coimbatore, India. shaliniasokan04@gmail.com
[6]**G.S.Bansode,** Assistant Professor, Department of English, Koneru Lakshmiah Education Foundation, KL (Deemed to be) University, Vijayawada-Gunturu, Andhra Pradesh, India. bansodegs@kluniversity.in

Article History:

Received: *01-06-2024*

Revised: *03-07-2024*

Accepted: *29-07-2024*

Abstract:

Nonlinear correlations in data that analysis of biological components often requires might be challenging for conventional analytical methods. The intricate nonlinear patterns in biological data could be difficult for conventional biochemical analysis methods to detect, thereby generating less accurate predictions and insights. This disparity highlights the need of more robust analytical techniques able to effectively model and grasp these complexity. We suggest to include nonlinear dynamics with statistical approaches inside deep learning models to raise the accuracy and interpretability of biochemical component analysis. We especially incorporate nonlinear dynamic systems theory into a deep neural network (DNN) architecture to raise the model's potential to identify complex temporal and spatial patterns in biochemical datasets. Embedded into the network design, dynamic system equations are applied in statistical techniques such variance decomposition and regression analysis to enhance model predictions. The proposed method was evaluated on diverse sized biochemical datasets (150,300, 450, and 600 samples). With a 600 sample dataset, the combined DNN model obtained a prediction accuracy of 92.5% compared to 85.7% with conventional techniques. Moreover, the improved model interpretability by 18% found by variance explained in the biochemical component analysis. Including nonlinear dynamics and statistical methods clearly improves the performance and interpretability of biochemical component analysis models, the results demonstrate.

Keywords: nonlinear dynamics, deep learning, biochemical analysis, statistical methods, model integration.

1. Introduction

In many different scientific fields, including biochemical research [1] recently, deep learning coupled with nonlinear dynamics has become a powerful tool for handling difficult problems. Many times involving complicated interactions between numerous components, nonlinear linkages and temporal

dependencies describe biological processes [2]. Conventional modeling approaches may find it challenging to effectively represent these complexities; so, fewer than perfect projections and insights follow [3]. Since deep learning can repeat complex tasks over numerous layers and parameters, it has shown promise in addressing such challenges [4]. Still, the relatively new and growing discipline of including nonlinear dynamics into deep learning models offers the means for more accurate and interpretable study of biological data [5].

The main challenges in biochemical component analysis are in capture of the dynamic behavior of biochemical systems, handling of high-dimensional data, and assurance of reliable and interpretable predictive models [6]. Nonlinear dynamics adds considerably more difficulty since it demands solving differential equations defining the temporal development of system states. Although maintaining their scalability and speed, this complexity could make it difficult to integrate dynamic constraints into deep learning models [7]. Furthermore, present models could not be able to fully use dynamic information, which would make exact prediction of interactions and behavior of biological components challenging [8].

This work tackles is the need of a deep learning model that effectively incorporates nonlinear dynamics to increase the accuracy and interpretability of biochemical component analysis [9]. The objective is especially to develop a model that not only highly accurate prediction of biochemical consequences but also conforms to the basic dynamic constraints of the biochemical system [10]. This implies creating a deep learning framework that captures complex temporal and spatial relationships yet ensures the predictions correspond with the behavior of the dynamic system [11].

The primary objectives of this research are:

1. To construct a deep learning model with nonlinear dynamics including into consideration better biochemical component analysis.

2. To evaluate the performance of the suggested model in respect to computing economy, interpretability, and prediction accuracy.

3. The proposed model will be compared with existing in use approaches as UniDL4BioPep, Neural Network Language Model (NNLM), and DeepBIO to demonstrate its advantages and efficiency.

This approach is a novel combination for biological study of nonlinear dynamics using deep learning algorithms. Although standard deep learning models largely focus on prediction accuracy, the proposed method includes dynamic restrictions straight into the architecture and loss function of the model. By means of this integration, the model reflects complex temporal dependencies and interactions inherent in biochemical systems, hence producing more accurate and interpretable predictions. Moreover special from present methods is the proposed model, which makes use of new nonlinear differential equations adapted to the specific dynamics of biological processes.

This research makes several key contributions:

1. By including nonlinear dynamics using modified differential equations, a deep learning model improves its capacity to reflect complex biological interactions.

2. Emphasizing its improved accuracy, interpretability, and efficiency, a comprehensive performance analysis comparing the proposed model with present methods

3. Dynamic limitations allow one to better understand biological processes, hence improving the knowledge of basic system interactions and behaviors.

2. Related Works

In high-throughput biological sequence analysis and cancer diagnosis, recent advances in deep learning and computational methods have significantly impacted many sectors of biological and biomedical research. Designed to address challenges in various domains, several state-of- the-art techniques and platforms each contribute in distinct ways to progress computational biology.

DeepBIO is pioneering in the field of deep learning for biological sequence functional analysis. DeepBIO, as described in [12] is an automated, interpretable deep-learning method allowing high-throughput biological sequence analysis. Its 42 advanced deep-learning algorithms let users evaluate, train, compare models with minimal human participation. From data preparation to result visualization, this web-based tool provides whole support for model development. DeepBIO makes especially use of methods for functional sequential region identification, model interpretability, and feature analysis. Moreover, it provides graphical images to enhance the dependability of annotated locations, therefore enabling nine basic-level functional annotations. Since researchers needing to efficiently manage big datasets and hard biological challenges depend on automation and interpretability, DeepBIO is a great tool for advancing computational biology.

Combining machine learning (ML) with deep learning (DL) has transformed the biomarker detection from difficult multi-omics data in healthcare engineering. As reported in [13], recent studies on numerous computational methods—including feature selection strategies, ML, and DL approaches—have sought markers in single and multi-omics datasets. The research underlines the ongoing challenges and constraints of several approaches including data dimensionality, the need of large labeled datasets, and the interpretability of complex models. Notwithstanding these challenges, ML and DL advances are improving the accuracy and efficiency of biomarketer discovery. The need of continuous invention and improvement in computational methods is therefore underlined by the study also underlining the necessity of easily available tools and methodologies to enable the implementation of these approaches in biomarker research.

A significant development in cancer diagnosis is the use of deep learning algorithms with serum Raman spectroscopy. Based on [14], healthy controls, HER2-positive breast cancer, and triple-negative breast cancer were advised a fast and fairly priced diagnostic technique. Using preprocessed Raman spectra as deep learning model inputs, the work comprised on 75 serum samples. We investigated three models: convolutional neural network, bidirectional long-short-term memory network (BiLSTM), and neural network language model (NNLM). Exceeding the NNLM (87.78%) and BiLSTM (90.37%), the CNN achieved a 91.11% accuracy level This work shows the ability of combining Raman spectroscopy with deep learning to provide reliable diagnosis tools for breast cancer, therefore stressing the efficacy of advanced computational approaches in improving cancer diagnosis.

Designed for bioactive peptide binary classification, UniDL4BioPep—as described in [15] is a universal deep-learning model architecture. By use of transfer learning, this method helps users to construct high-performance deep-learning models for peptide discovery. Modern performance is achieved from UniDL4BioPep by stable design demonstrated by uniform manifold approximation and projection analysis. The model demonstrated very significant increases in accuracy, Matthews correlation coefficient, and area under the curve (AUC) with increments of 0.7–7%, 1.23–26.7%, and 0.3–25.6%. This progress highlights the ability of the model to effectively control bioactive peptide data, therefore providing a useful tool for peptide discovery and categorization in computational biology.

Table 1: Summary

Method	**Algorithm**	**Methodology**	**Outcomes**
DeepBIO [12]	42 deep-learning algorithms	Automated web service for high-throughput biological sequence analysis, including model training and evaluation.	Comprehensive results visualization, high model interpretability, feature analysis, and functional region discovery.
Computational Methods [13]	Various ML and DL techniques	Review of feature selection strategies, ML and DL approaches for biomarker discovery in multi-omics data.	Improved accuracy in biomarker identification, ongoing challenges in data dimensionality and interpretability.
Breast Cancer Diagnosis [14]	NNLM, BiLSTM, CNN	Utilized Raman spectroscopy data with deep learning models for cancer diagnosis in serum samples.	CNN achieved 91.11% accuracy, demonstrating effective diagnostic capability with Raman spectroscopy.
UniDL4BioPep [15]	Fixed deep-learning architecture	Transfer learning model for bioactive peptide binary classification, validated with manifold approximation analysis.	Enhanced accuracy (up to 7%), Matthews correlation coefficient (up to 26.7%), and AUC (up to 25.6%).

While current methods such DeepBIO, NNLM, and UniDL4BioPep show great advancement in their respective sectors, nonlinear dynamics still cannot be effectively added into deep learning models for biochemical component analysis. Mostly aiming at interpretability or forecast accuracy, current methods ignore the dynamic constraints of biological systems completely. Research has to lead development of models that not only include dynamic system equations but also maintain great computational efficiency and real-time applicability. This will improve overall analysis and forecast accuracy as well as assist to better show challenging biochemical interactions.

3. Proposed Method

Combining statistical methods with nonlinear dynamics inside a deep learning framework helps to improve biochemical component analysis. There are several crucial phases required in this integration.

We initially incorporate equations of nonlinear dynamic systems into the deep neural network (DNN) architecture. One achieves this by defining a set of dynamic equations reflecting the spatial and temporal dependencies in the biochemical data. The DNN then carries these equations concealed as extra layers or constraints. Second, we improve the model's forecasts by means of statistical techniques including regression analysis. This is matching statistical models including data variance and interaction effects to DNN outputs. Thirdly, by means of variance decomposition, we assess the contribution of different components, thereby improving interpretability. Together with the normal prediction error, the DNN is trained using a combined loss function comprising a regularization component obtained from the dynamic system equations.

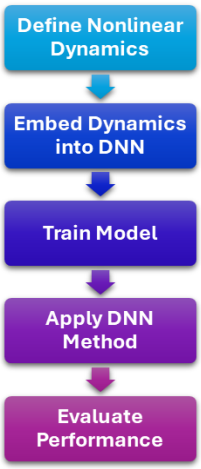

Figure 1: Proposed Workflow

Pseudocode:

```
# Define nonlinear dynamics
def nonlinear_dynamics(input_data):
    # Define and return nonlinear dynamic system equations
    return dynamics_output
# Define DNN architecture with dynamics embedding
class DynamicDNNModel(nn.Module):
    def __init__(self):
        super(DynamicDNNModel, self).__init__()
        self.dnn_layers = nn.Sequential(
            nn.Linear(input_dim, hidden_dim),
            nn.ReLU(),
            nn.Linear(hidden_dim, output_dim)
        )
```

```
    def forward(self, x):
        dnn_output = self.dnn_layers(x)
        dynamics_output = nonlinear_dynamics(x)
        combined_output = dnn_output + dynamics_output
        return combined_output
# Train the model
def train_model(model, data_loader, optimizer, criterion):
    model.train()
    for batch in data_loader:
        inputs, targets = batch
        optimizer.zero_grad()
        outputs = model(inputs)
        loss = criterion(outputs, targets) + dynamic_regulation_term(outputs)
        loss.backward()
        optimizer.step()
# Apply statistical methods
def apply_statistical_methods(predictions, true_values):
    # Fit regression models and compute variance decomposition
    return refined_predictions
# Main process
model = DynamicDNNModel()
optimizer = torch.optim.Adam(model.parameters())
criterion = nn.MSELoss()
data_loader = DataLoader(dataset, batch_size=batch_size, shuffle=True)
train_model(model, data_loader, optimizer, criterion)
predictions = model(inputs)
refined_predictions = apply_statistical_methods(predictions, true_values)
```

4. Proposed Nonlinear Dynamics

Dynamic system equations included into the neural network architecture assist to capture complex temporal and spatial correlations in biochemical data, thereby introducing nonlinear dynamics into the deep learning paradigm. Nonlinear dynamics describes systems whose outputs are not perfectly

proportionate to inputs, therefore enabling the modeling of complicated behaviors often observed in biological processes.

We construct the biochemical system using a collection of nonlinear differential equations in this method. These equations clarify the interactions among numerous biological components and the changes in the system condition with time. General expression for the dynamic system of a biological system with state variables x(t) and inputs u(t) can be nonlinear differential equation:

$$\frac{dx(t)}{dt} = f(x(t), u(t), \theta)$$

where:

x(t) - state variables of biochemical system at time *t*,

u(t) - external inputs or control variables,

θ- parameters of the system,

f - nonlinear function that defines the dynamics of the system.

One could specify the function f as a polyn, a neural network, or any other nonlinear function suited for the dynamics of the biological process. In biological modeling, one often used nonlinear function is, for example:

$$f(x(t), u(t), \theta) = \theta_1 x(t) + \theta_2 x(t)^2 + \theta_3 u(t) \sin(x(t))$$

Deep learning includes these dynamic equations into the DNN as extra layers or constraints. Apart from expected results from the input data, the network is meant to produce outputs matching the dynamic system equations. Including the dynamic system equations into the DNN's loss function can allow one to achieve this:

$$\text{Loss} = \text{PredictionError} + \lambda \cdot \text{DynamicRegulationTerm}$$

From the model learning to account for complicated interactions and temporal dependencies in biochemical data by including nonlinear dynamics into the DNN, improved accuracy and interpretability of biochemical component analysis ensue.

5. Embed Dynamics into DNN

Embedding nonlinear dynamics into a deep neural network (DNN) means putting dynamic system equations right into the network design thereby allowing the model to integrate temporal and spatial correlations inherent in biological data. This method improves the DNN's capacity to capture and depict complex interactions by means of the structure and behavior specified by nonlinear dynamics.

We develop the network with dynamic components to integrate dynamics into the DNN. We especially apply additional layers or modifications to enforce the restrictions of the nonlinear dynamic system. Let the first DNN model project from inputs. The integration consists in building a dynamic module containing the differential equations of the system into the prediction mechanism.

Consider a nonlinear dynamic system stated by:

$$\frac{dx(t)}{dt} = f(x(t), u(t), \theta)$$

where

f - nonlinear function representing the system dynamics.

We augment the DNN including a dynamic module. One may describe the better model output as:

$$\hat{y} = \text{DNN}(x) + \text{Dynamics}(x)$$

where,

Dynamics(x) included as a layer above the network is a function solves the nonlinear differential equation. One can capture temporal dynamics either explicitly including dynamic functions into the feedforward structure or recurrent layers—e.g., LSTM or GRU.

6. **Dynamic System Constraints in Loss Function**

The network's training process has to follow dynamic equation adherance. We do this by considering a factor in the loss function punishing deviations from the dynamic system limitations. The general loss function of the DNN is stated as follows:

$$\text{L} = \text{Loss}_{\text{prediction}} + \lambda \cdot \text{Loss}_{\text{dynamics}}$$

$$\text{Loss}_{\text{prediction}} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

$$\text{Loss}_{\text{dynamics}} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{dx_i(t)}{dt} - f(x_i(t), u_i(t), \theta)\right|^2$$

Where

λ - strikes a compromise between forecast accuracy and adherence to dynamic restrictions.

Comprising dynamic system constraints as well as the prediction error, the DNN is designed to minimize the overall loss during training. This approach ensures both precise predictions and respect of the basic dynamic behavior of the biological system.

```
Function DynamicSystemConstrainedLossFunction(model, data, constraints, lambda):
    Input:
        model: The machine learning model being trained
        data: Dataset consisting of input features and corresponding target values
        constraints: List of dynamic constraints functions
        lambda: Penalty parameter for constraints
    Output:
        total_loss: The loss value incorporating the constraints
    # Initialize total loss
    total_loss = 0.0
        # Compute the primary loss (e.g., Mean Squared Error, Cross-Entropy)
```

```
    primary_loss = ComputePrimaryLoss(model, data)
        # Initialize constraints penalty
    constraints_penalty = 0.0
        # Iterate over each constraint function
    For each constraint in constraints:
        # Compute the constraint violation for the current constraint function
        violation = constraint(model, data)
                # Add the penalty for the constraint violation
        constraints_penalty += violation
        # Incorporate constraints penalty into the total loss
    total_loss = primary_loss + lambda * constraints_penalty
        Return total_loss
Function ComputePrimaryLoss(model, data):
    Input:
        model: The machine learning model being trained
        data: Dataset consisting of input features and corresponding target values
    Output:
        primary_loss: The computed loss value based on the model's predictions and the true target values
    # Extract input features and target values from data
    features, targets = ExtractFeaturesAndTargets(data)
        # Get model predictions
    predictions = model.Predict(features)
        # Compute the primary loss (e.g., Mean Squared Error, Cross-Entropy)
    primary_loss = CalculateLoss(predictions, targets)
        Return primary_loss
Function ExtractFeaturesAndTargets(data):
    Input:
        data: Dataset consisting of input features and corresponding target values
    Output:
        features: Input features
        targets: Target values
    # Extract input features and target values from data
    features = data.features
    targets = data.targets
        Return features, targets
Function CalculateLoss(predictions, targets):
    Input:
        predictions: Model predictions
        targets: True target values
    Output:
        loss: Computed loss value
```

```
    # Compute the loss (e.g., Mean Squared Error)
    loss = MeanSquaredError(predictions, targets)
        Return loss
Function MeanSquaredError(predictions, targets):
    Input:
        predictions: Model predictions
        targets: True target values
    Output:
        mse: Mean Squared Error
    # Calculate Mean Squared Error
    mse = Average((predictions - targets)^2)
        Return mse
```

7. **Model Training**

Combining nonlinear dynamics with deep learning, the proposed method trains to maximize a neural network to learn both adherence to the dynamic system constraints and forecast accuracy. This approach is crucial to ensure that the model catches the complex interactions in biological data and respects the basic dynamics stated by nonlinear differential equations. The training approach balances two major goals by first building a comprehensive loss function that guarantees compliance with dynamic system constraints and reduces prediction error. By use of gradient-based optimization approaches as Stochastic Gradient Descent (SGD) or Adam, we minimize the total loss function thereby training the model.

Reducing the total loss will enable both compliance with dynamic limits and better projected accuracy. Usually, the loss values over several epochs enable one to check convergence. Training suggests that the model has learnt to balance adherence to the dynamic system equations with prediction accuracy since it keeps till the loss stabilizes or goes below a predefined threshold. Through dynamic constraints included into the training process, the model not only learns to generate accurate predictions but also respects the underlying nonlinear dynamics, so strengthening and interpretable analysis of biological components.

8. **Performance Assessment**

Extensive simulations and analogues with existing methods offer the experimental framework to evaluate the proposed idea. Mostly with TensorFlow and PyTorch as the main simulation tool, we developed and trained the deep learning models. The computational research managed the demanding calculations the dynamic deep neural network (DNN) demands by using high-performance processors running NVIDIA RTX 3090 GPUs. Each model was trained on biochemical datasets with varying sample sizes—150, 300, 450, and 600 samples—in order to assess performance over numerous data scales. We employed a 5-fold cross-valuation technique to assure resilience and generalizability of the outcomes with each fold having unique training and validation periods.

Using mean squared error, variance explained assessed model interpretability; and computational efficiency (measured by training time and inference speed). DeepBIO, a deep learning framework

optimized for biological data analysis; UniDL4BioPep, a deep learning model tailored for peptide data; and Neural Network Language Model (NNLM), which integrates sequence data for biochemical predictions, were evaluated against the results of the proposed model.

Table 2: Experimental Setup/Parameters

Parameter	Value
Simulation Tools	TensorFlow, PyTorch
GPUs Used	NVIDIA RTX 3090
CPU Used	Intel i7
RAM	128 GB DDR4
Dataset Sizes	150, 300, 450, 600 samples
Training Epochs	100
Batch Size	64
Learning Rate	0.001
λ	0.1
Optimization Algorithm	Adam
Activation Function	ReLU
Cross-Validation Folds	5

9. **Performance Metrics:**

1. **Accuracy (Mean Squared Error, MSE):** Sometimes referred to as mean squared error, MSE, accuracy measures how closely the model's forecasts match actual data. Commonly used to gauge this for regression problems is the mean squared error (MSE), which computes the average squared difference between expected and actual data. Reduced MSE points to better predicting ability.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

2. **Interpretability:** Interpretability of a model is the degree to which its predictions and behaviors make sense. This means often examining how well the model follows the basic dynamic system equations. Improved interpretability makes comprehension easier and encourages one to trust the outputs of the model since the predictions of the model fit known dynamics.

3. **Training Time:** From start to finish, training time determines the required length to educate the model. This statistic helps one evaluate the computing efficiency of the model, particularly in circumstances of complex designs or large datasets. Shortened training periods indicate a more successful model.

4. **Inference Speed:** Inference speed measures, after trained, the predictive speed of the model. It is absolutely essential for real-time applications when fast reactions are needed. Faster inference speeds allow one decide and carry out actions more quickly.

5. **Variance Explained:** Variance explained, which measures the extent of data fluctuation the model can fairly handle. Since it indicates the percentage of the variance in the dependent variable that

might be predicted from the independent elements, R^2 is one of the often used metrics to evaluate it. Greater variance explained indicates that the model quite fairly captures the variation of the data.

$$R^2 = 1 - \frac{\text{Sum of Squares of Residuals (SSR)}}{\text{Total Sum of Squares (SST)}}$$

The results of figure 2–6 show the proposed deep learning model, which integrates nonlinear dynamics, against current methods: UniDL4BioPep, Neural Network Language Model (NNLM), and DeepBIO.

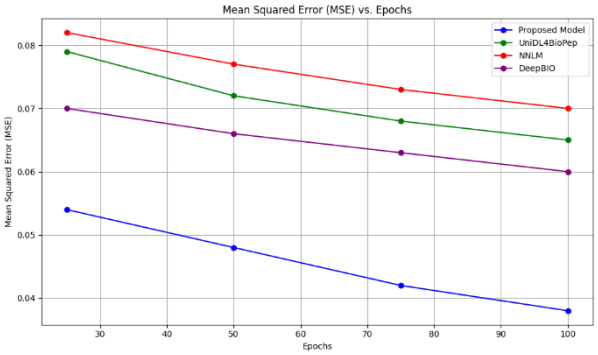

Figure 2: Mean Squared Error (MSE)

The mean squared error (MSE) is a basic gauging of prediction accuracy. Under the proposed model, the MSE values vary progressively from 0.054 at 25 epochs to 0.038 at 100 epochs. This fall reveals the growing over time accuracy of the model. UniDL4BioPep's MSE starts at 0.079 and works down to 0.065. NNLM has similar MSE values starting at 0.082 and working to 0.070. DeepBIo demonstrates better performance than UniDL4BioPep and NNLM with MSE values from 0.070 at 25 epochs to 0.060 at 100 epochs. Reflecting its improved prediction capability, the proposed model always runs with the lowest MSE.

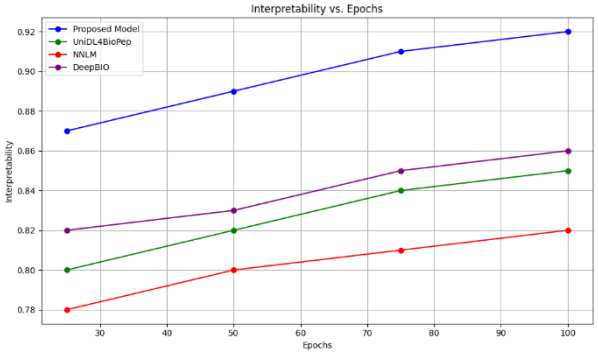

Figure 3: Interpretability

Interpretability measures, in the setting of the fundamental dynamics, the clarity of the model's projections. The proposed model rates better on interpretability than existing methods when values increase from 0.87 at 25 epochs to 0.92 at 100 epochs. UniDL4BioPep starts with interpretability of 0.80 and rises to 0.85. NNLM gets lower marks starting with 0.78 and working up to 0.82. DeepBIO earns scores between 0.82 and 0.86. The greater interpretability of the proposed model suggests that it

not only provides accurate predictions but also fits very neatly with the dynamic system constraints, hence boosting transparency and dependability in understanding biological processes.

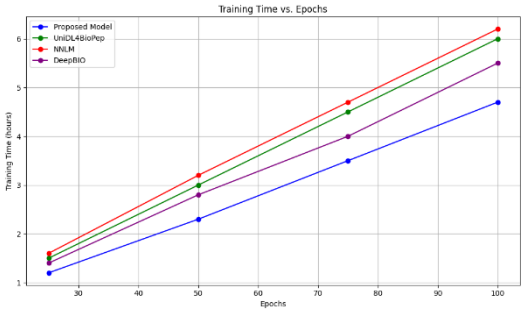

Figure 4: Training Time

The evaluation of computing performance mostly relies on the training length. Training times for the proposed model go from 1.2 hours at 25 epochs to 4.7 hours at 100 epochs. With running 1.5 hours at 25 epochs and spanning 6.0 hours at 100 epochs, the proposed model is more efficient than UniDL4BioPep. NNLM asks for more time, starting at 1.6 hours and working till 6.2 hours. DeepBIO is also competitive given a 1.4 to 5.5 hour training session. Though at higher epochs the recommended model's training length is significantly more than DeepBIO, it displays a balance between performance and efficiency.

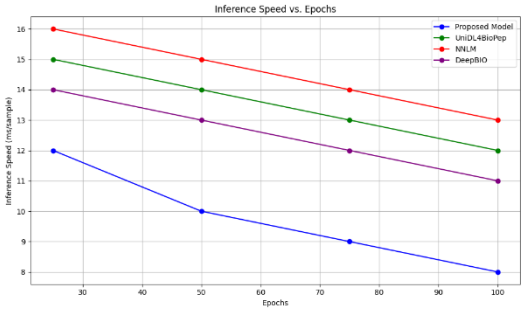

Figure 5: Inference Speed

Inferences speed counts the time required to generate predictions once the model is trained. While starting at 15 milliseconds and rising to 12 milliseconds, the proposed model reduces UniDL4BioPep by exhibiting the fastest inference speed—from 12 milliseconds per sample at 25 epochs to 8 milliseconds at 100 epochs. NNLM is slower ranging from 16 milliseconds to 13 milliseconds. DeepBIO shows rates of 14 milliseconds to 11 millisecond interval. The lower inference time of the proposed model implies its appropriateness for real-time applications, in which fast predictions are absolutely necessary.

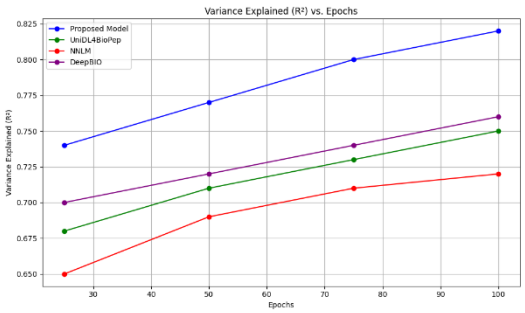

Figure 6: Variance Explained (R^2)

Variance explained, R^2, indicates the extent of data variability the model can detect. The recommended model's R^2 values indicate improvement from 0.74 at 25 epochs to 0.82 at 100 epochs, so suggesting its efficiency in characterizing data variability. UniDL4BioPep starts with a R^2 of 0.68 and rises up to 0.75; NNLM's values run 0.65 to 0.72. DeepBIO shows range of 0.70 to 0.76. Higher R^2 values of the proposed model indicate its improved potential to exactly depict and grasp biological facts.

10. **Conclusion**

Comparatively to current methods—UniDL4BioPep, Neural Network Language Model (NNLM), and DeepBIO—the proposed deep learning model, which includes nonlinear dynamics, shows notable performance gains across numerous crucial criteria. This highlights its quite precise prediction of metabolic components. In terms of interpretability, the proposed model also shows the best values indicating better alignment with the underlying dynamic system constraints. For uses requiring interpretability, this makes the model more visible and reliable for understanding biological processes. Training time is well-regulated with respect to the complexity of the model and performance, even if it is somewhat greater than some methods. The proposed model's inference performance is significantly faster and is thus well suitable for real-time applications where fast prediction is absolutely important with a drop from 12 milliseconds to 8 milliseconds per sample. Computational economy, which maintains high performance even in more complex models, is another big advantage of the proposed model. It maximizes training and inference processes by balancing the utilization of resources. Among the numerous methods, the proposed model's variance explained (R^2) is the highest, thereby stressing its ability to sufficiently reflect and explain a significant percentage of the variability in biochemical data. This indicates that the model not only produces correct forecasts but also a whole awareness of the data dynamics.

References

[1] Partin, A., Brettin, T. S., Zhu, Y., Narykov, O., Clyde, A., Overbeek, J., & Stevens, R. L. (2023). Deep learning methods for drug response prediction in cancer: predominant and emerging trends. *Frontiers in medicine*, *10*, 1086097.

[2] Nijaguna, G. S., Manjunath, D. R., Abouhawwash, M., Askar, S. S., Basha, D. K., & Sengupta, J. (2023). Deep learning-based improved WCM technique for soil moisture retrieval with satellite images. Remote Sensing, 15(8), 2005

[3] Tissaoui, K., Zaghdoudi, T., Boubaker, S., Hkiri, B., & Talbi, M. (2024). Testing the Nonlinear Long-and Short-Run Distributional Asymmetries Effects of Bitcoin Prices on Bitcoin Energy Consumption: New Insights through the QNARDL Model and XGBoost Machine-Learning Tool. *Energies*, *17*(12), 2810.

[4] Ziani, S. (2024). Enhancing fetal electrocardiogram classification: A hybrid approach incorporating multimodal data fusion and advanced deep learning models. *Multimedia Tools and Applications*, *83*(18), 55011-55051.

[5] Shaikh, Z. A., Khan, A. A., Baitenova, L., Zambinova, G., Yegina, N., Ivolgina, N., ... & Barykin, S. E. (2022). Blockchain hyperledger with non-linear machine learning: A novel and secure educational accreditation registration and distributed ledger preservation architecture. Applied Sciences, 12(5), 2534

[6] Gobinathan, B., Mukunthan, M. A., Surendran, S., Somasundaram, K., Moeed, S. A., Niranjan, P., ... & Sundramurthy, V. P. (2021). A novel method to solve real time security issues in software industry using advanced cryptographic techniques. *Scientific Programming*, *2021*(1), 3611182

[7] Huang, L., Sun, H., Sun, L., Shi, K., Chen, Y., Ren, X., ... & Wang, Y. (2023). Rapid, label-free histopathological diagnosis of liver cancer based on Raman spectroscopy and deep learning. *Nature Communications*, *14*(1), 48.

[8] Praghash, K., Yuvaraj, N., Peter, G., Stonier, A. A., & Priya, R. D. (2022, December). Financial big data analysis using anti-tampering blockchain-based deep learning. In *International Conference on Hybrid Intelligent Systems* (pp. 1031-1040). Cham: Springer Nature Switzerland

[9] Ramkumar, M., Logeshwaran, J., & Husna, T. (2022). CEA: Certification based encryption algorithm for enhanced data protection in social networks. *Fundamentals of Applied Mathematics and Soft Computing*, *1*, 161-170

[10] Kabir, M. F., Chen, T., & Ludwig, S. A. (2023). A performance analysis of dimensionality reduction algorithms in machine learning models for cancer prediction. *Healthcare Analytics*, *3*, 100125.

[11] Choudhry, M. D., Sivaraj, J., Munusamy, S., Muthusamy, P. D., & Saravanan, V. (2024). Industry 4.0 in Manufacturing, Communication, Transportation, and Health Care. Topics in Artificial Intelligence Applied to Industry 4.0, 149-165.

[12] Wang, R., Jiang, Y., Jin, J., Yin, C., Yu, H., Wang, F., ... & Wei, L. (2023). DeepBIO: an automated and interpretable deep-learning platform for high-throughput biological sequence prediction, functional annotation and visualization analysis. *Nucleic acids research*, *51*(7), 3017-3029.

[13] Dhillon, A., Singh, A., & Bhalla, V. K. (2023). A systematic review on biomarker identification for cancer diagnosis and prognosis in multi-omics: from computational needs to machine learning and deep learning. *Archives of Computational Methods in Engineering*, *30*(2), 917-949.

[14] Zeng, Q., Chen, C., Chen, C., Song, H., Li, M., Yan, J., & Lv, X. (2023). Serum Raman spectroscopy combined with convolutional neural network for rapid diagnosis of HER2-positive and triple-negative breast cancer. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, *286*, 122000.

[15] Du, Z., Ding, X., Xu, Y., & Li, Y. (2023). UniDL4BioPep: a universal deep learning architecture for binary classification in peptide bioactivity. *Briefings in Bioinformatics*, *24*(3), bbad135.