International Journal of Innovative Research and Scientific Studies, 8(3) 2025, pages: 2774-2786

IJIRSS
International Journal of Innovative Research & Scientific Studies

ISSN: 2617-6548

URL: www.ijirss.com

Implementation of RNN-LSTM with L1 regularization for predicting labels from chimpanzee DNA sequences using pseudo-labeling

Sugiyarto Surono[1*], Goh Khang Wen[2], Arif Rahman[3], Lalu M. Irham[4], Sintia Afriyani[5]

[1]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, Ahmad Dahlan University, Yogyakarta, Indonesia.*
[2]*Faculty of Data Science and Information Technology, INTI International University, Putra Nilai, Malaysia.*
[2]*Department of Computer Science, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan.*
[3]*Department of Information Systems, Faculty of Science and Applied Technology, Ahmad Dahlan University, Yogyakarta, Indonesia.*
[4]*Department of Pharmacy, Faculty of Pharmacy, Ahmad Dahlan University, Yogyakarta, Indonesia.*
[5]*Department of Statistics, Faculty of Mathematics and Natural Sciences, Islamic University of Indonesia, Yogyakarta, Indonesia*.

Corresponding author: Sugiyarto Surono (*Email: Sugiyarto@math.uad.ac.id*)

Abstract

Chimpanzee genome research plays a crucial role in understanding evolution, health, and biological functions. However, incomplete labeling of DNA sequence data presents a challenge for accurate genomic classification. This study aims to improve chimpanzee DNA sequence classification by addressing label scarcity and data imbalance through a deep learning approach. A Recurrent Neural Network Long Short-Term Memory (RNN-LSTM) model with L1 Regularization and pseudo-labeling is employed to enhance classification performance. The workflow includes numerical encoding of DNA sequences, pseudo-labeling to augment training data, and model training using Stochastic Gradient Descent (SGD) optimization. Performance evaluation is conducted using classification accuracy and AUC metrics. Results show that the proposed approach achieves high classification accuracy, with an AUC ranging from 0.94 to 0.99, significantly improving the handling of imbalanced datasets. The integration of pseudo-labeling effectively leverages unlabeled DNA sequences, leading to a more robust genomic classification model. These findings highlight the potential of combining RNN-LSTM with L1 Regularization and pseudo-labeling to address incomplete labeling in genomic datasets. The study advances genomic classification techniques and supports Goal 3: Good Health and Well-being of the Sustainable Development Goals (SDGs) by enhancing DNA sequence classification accuracy, facilitating early disease detection, precision medicine, and evolutionary studies.

Keywords: Chimpanzee genome analysis, Goal 3, Good health and well-being (SDGs), L1 regularization feature selection, Pseudo-labeling in genomics, RNN-LSTM for DNA sequence classification.

DOI: 10.53894/ijirss.v8i3.7083
Funding: This research was funded by the Internal Research Grant for the fiscal year 2024 under (Grant Number: PKLN-288/SP3/LPPM-UAD/XI/2024).
History: Received: 24 March 2025 / **Revised:** 28 April 2025 / **Accepted:** 30 April 2025 / **Published:** 16 May 2025

Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.

Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.
Acknowledgments: The authors would like to express their gratitude to the Research Center for funding this research through the Internal Research Grant for the fiscal year 2024 under contract number PKLN-288/SP3/LPPM-UAD/XI/2024.
Publisher: Innovative Research Publishing

1. Introduction

Genomic data analysis has become a crucial field in the modern era, particularly in the context of global health [1]. Understanding the relationship between DNA sequences and their biological functions is essential a identifying disease mechanisms and developing more targeted therapies [2]. However, one of the main challenges in genomic analysis is the lack of labelled data, where many DNA sequences do not have clear protein annotations. This limitation can hinder the mapping of DNA sequences to their corresponding proteins, restricting potential applications in biomedical research and evolutionary studies [3]. Therefore, more sophisticated approaches are needed to improve the accuracy of genomic data classification.

Chimpanzee genomes are utilized in this study due to their significant similarity to the human genome, making them an essential subject in genomic research [4]. The study of chimpanzee genomes provides insights into evolution, biological functions, and disease mechanisms relevant to humans [5]. Several gene families analyzed in this research include G protein-coupled receptors (GPCRs), tyrosine kinases, tyrosine phosphatases, synthetases, synthases, ion channels, and transcription factors. These genes play a critical role in various physiological and pathological processes related to human health, such as cell signaling regulation, gene expression, and intercellular communication [6].

However, many prior studies have relied solely on fully labeled datasets, which are less effective for real-world genomic data that is often incomplete [7]. To address this limitation, this study adopts pseudo-labeling, a semi-supervised learning technique that enables models to utilize unlabeled data to enhance classification accuracy and generalization [8]. Building on these advancements, deep learning, particularly Convolutional Neural Networks (CNNs), has become a crucial tool for processing complex data. However, CNNs are vulnerable to noise interference, which can affect classification accuracy [9].

Previous research has demonstrated that deep learning models, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM), have been successfully applied to genomic data analysis. CNNs have shown high accuracy in DNA sequence classification, while LSTM excels in capturing long-term dependencies within sequential data [10]. Deep learning techniques (DL) have also significantly improved the accuracy of DNA sequence predictions and classifications. However, challenges remain, particularly in identifying and predicting splice sites in eukaryotic genomes due to the high rate of erroneous discoveries. To address this issue, a recent study proposed a bidirectional LSTM-Recurrent Neural Network (RNN) combined with a Gated Recurrent Unit (GRU) to recognize and predict splice sites in eukaryotic DNA sequences based on intron length constraints, demonstrating improved performance with increasing training epochs. In addition to genomics, DL has also been widely adopted in medical diagnostics, particularly in the classification of ultrasound images for early detection of diseases [11]. For instance, a recent study utilized VGG19 Net to classify ovarian ultrasound images for detecting Polycystic Ovary Syndrome (PCOS), outperforming traditional machine learning techniques such as Random Forest, Logistic Regression, Bayesian Classifier, Support Vector Machine, and Artificial Neural Network in distinguishing between benign and malignant cysts [12]. Despite these advancements, many prior studies have relied solely on fully labelled datasets, which limits their effectiveness when applied to real-world genomic data that is often incomplete [7]. To overcome this limitation, this study employs pseudo-labelling, a semi-supervised learning technique that allows models to utilize unlabeled data, improving classification accuracy and generalization [8].

This research implements a Recurrent Neural Network - Long Short-Term Memory (RNN-LSTM) model combined with pseudo-labelling to classify chimpanzee DNA sequences obtained from Kaggle and Ensemble Genome Browser, introducing a more innovative approach than previous studies. Instead of relying solely on labelled datasets, as seen in conventional deep learning methods like CNNs or LSTMs without semi-supervised learning, this study incorporates L1 regularization-based feature selection to identify the most relevant genetic features, reducing data dimensionality while preserving essential information. Additionally, pseudo-labelling is integrated as a semi-supervised learning strategy, allowing unlabeled data to be assigned labels based on initial model predictions, which are then incorporated into the training process to enhance model generalization, especially in scenarios with limited labelled data. In terms of deep learning architecture, RNN-LSTM is chosen for its ability to capture both long-term and short-term dependencies in sequential data, which CNN-based methods cannot fully optimize. The combination of L1 regularization, pseudo-labelling, and RNN-LSTM architecture makes the model more accurate, efficient in data processing, and robust in handling imbalanced genomic datasets. This approach also aligns with Goal 3: Good Health and Well-being of the Sustainable Development Goals (SDGs) by improving genomic analysis methods for early disease detection and the development of more precise gene-based therapies.

2. Methods and Materials

The method in this study is a pseudo-labelling approach with the LSTM-RNN model to classify DNA sequences. This process includes data preprocessing, initial model training using labelled data, pseudo-labelling unlabeled data, and retraining the model with a combination of labelled and pseudo-labelled data, as shown in Figure 1.

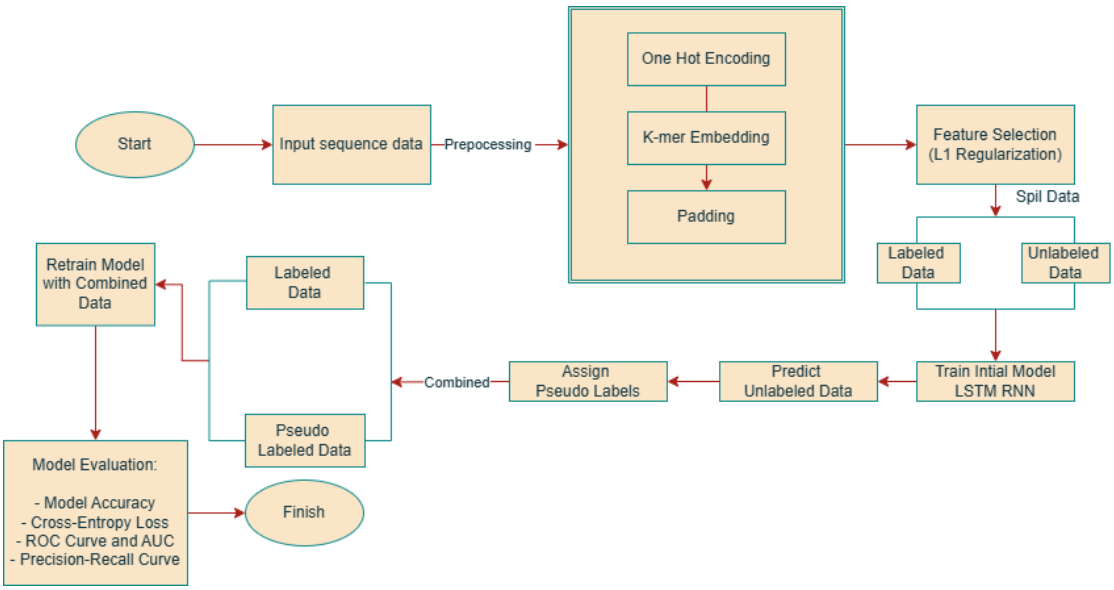

Figure 1.
Workflow of LSTM-RNN Training with Pseudo-Labeling for DNA Sequence Classification.

Figure 1 illustrates the workflow of the research on training an LSTM-RNN model with pseudo-labelling techniques to recognize patterns in DNA sequences and predict protein labels. The study begins with genomic data preprocessing, including One-Hot Encoding, K-mer Embedding, and Padding, followed by Feature Selection using L1 Regularization. The data is then divided into labelled and unlabeled datasets, where the initial LSTM-RNN model is trained using labelled data. This model is subsequently used to predict labels for the unlabeled data, which are then assigned pseudo-labels. The original labelled data and pseudo-labelled data are combined to retrain the model, improving prediction performance. The model is evaluated using metrics such as accuracy, cross-entropy loss, ROC curve & AUC, and precision-recall curve to assess the effectiveness of the pseudo-labelling approach in genomic data classification.

The data used in this study are DNA sequence data, so the initial step is data preprocessing to convert nucleotide sequences into a numerical format that can be understood by the model. The data preprocessing stage is a crucial step in building a machine learning model, especially in DNA sequence analysis. This process aims to clean, encode, and prepare the data for use in model training.

2.1. Data Preprocessing

DNA sequence data preprocessing is a step in bioinformatics analysis aimed at converting nucleotide sequences into numerical representations that can be used by machine learning models [13]. This process includes various techniques such as one-hot encoding, k-mer-based embedding, padding, and array conversion to ensure the data is ready for model training [14]. The detailed steps of DNA sequence data preprocessing are illustrated in Figure 2.

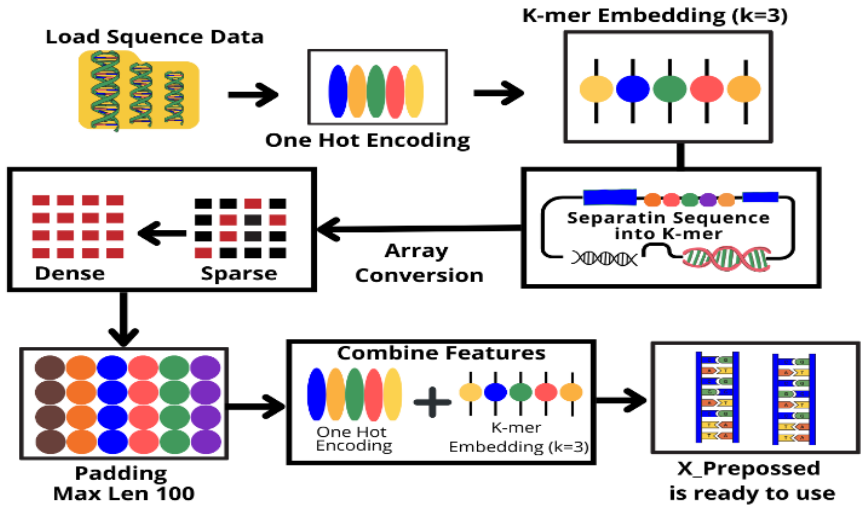

Figure 2.
Preprocessing Pipeline of DNA Sequence Data.

Figure 2 illustrates the preprocessing pipeline of DNA sequence data in this study, starting with loading the sequence data as input. The DNA sequence is then encoded using one-hot encoding to represent nucleotides as binary vectors and transformed into a k-mer (k=3) embedding to capture specific patterns. Next, the data is converted into sparse and dense

representations for efficient processing by the model, and padding is applied to ensure uniform sequence length with a maximum of 100. Features from one-hot encoding and k-mer embedding are combined to produce a richer representation, and once all steps are completed, the data is ready for machine learning model training, aiming to improve efficiency and accuracy in genetic function prediction.

The first step in DNA sequence data preprocessing is One-Hot Encoding. This method converts nucleotides in a DNA sequence (A, T, G, C) into binary vectors, making them easier for machine learning models to process [15]. Each nucleotide is represented by a 4-dimensional vector, where one element is set to 1 (active) and the other three elements are set to 0 (inactive). For example, the nucleotides "A", "T", "G", and "C" are represented as follows:

A = [1, 0, 0, 0].
T = [0, 1, 0, 0].
G = [0, 0, 1, 0].
C = [0, 0, 0, 1].

This transformation can be described as mapping each nucleotide $x \in \{A, T, G, C\}$ to a vector $\vec{v} \in \{0,1\}^4$, where the position corresponding to the nucleotide is set to 1, and the remaining positions are set to 0.

Next, K-mer Embedding is applied, which offers another way to represent DNA sequence data. In this method, the DNA sequence is split into smaller segments of size $k(k-mers)$, capturing local patterns within the sequence. For instance, with $k = 3$, the DNA sequence "ATGCGT" would generate the k-mers: "ATG", "TGC", and "CGT". Each k-mer is then represented as a vector using embedding techniques like Word2Vec or GloVe [16]. This approach allows the model to capture more complex relationships between k-mers, recognizing patterns and similarities in the sequence. Mathematically, this can be viewed as mapping each k-mer $k \in \{k-mer\}$ to a vector $\vec{v} \in R^d$, where each k-mer is represented in a continuous vector space.

Lastly, Padding is applied to ensure that all sequences have the same length [17]. Padding adds a specific symbol (e.g., 'N' or 'X') to shorter sequences until they reach a uniform length. This step is crucial for models that require fixed-length input. For example, if the maximum sequence length is set to 100 and a sequence has a length of 90, padding will add 10 'X' symbols to make the sequence length 100. This ensures that all input sequences are consistent in size. Padding can be represented as a function $P: \{S_i\} \rightarrow \{S_j\}$, where S_i is the original sequence with length n_i, and S_j is the padded sequence with length n_j (where $n_j \geq n_i$) by adding specific symbols (e.g., 'X') until the sequence length reaches n_j.

2.2. L1 Regularization Feature Selection

In this study, L1 Regularization is applied for feature selection on chimpanzee genome data to identify relevant genetic features for predicting specific diseases or traits [18]. L1 Regularization, also known as Lasso (Least Absolute Shrinkage and Selection Operator), is a technique that adds a penalty to the absolute values of the model coefficients [19]. This penalty encourages some of the coefficients to shrink to zero, effectively performing feature selection by eliminating less relevant features. This not only improves the model's performance by reducing overfitting but also enhances its interpretability by highlighting the most important features for prediction. A regression model with L1 regularization can be expressed as follows:

$$\text{Minimize} \quad \text{Loss Function} + \lambda \sum_{i=1}^{n} |w_i| \qquad (1)$$

Where w_i represents the model coefficients and λ is the regularization parameter that controls the degree of feature selection.

2.3. Long Short-Term Memory (LSTM) Architecture within a Recurrent Neural Network (RNN)

The Long Short-Term Memory (LSTM) architecture within a Recurrent Neural Network (RNN) has been employed to analyze chimpanzee DNA sequence data, both labeled and unlabeled. Labelled data consists of DNA sequences classified into specific gene families, while unlabeled data includes sequences without predefined categories. In this model, each DNA sequence is represented as an embedding vector to capture structural relationships and biological context before being processed through LSTM layers [20]. LSTM plays a crucial role in identifying sequential patterns and long-range dependencies between nucleotides by preserving essential information within memory cells. The hidden LSTM layer encodes contextual information from DNA sequences, which is then classified using a softmax layer to predict the corresponding gene family. For unlabeled data, the model can be applied in unsupervised learning to uncover latent patterns and structures within DNA sequences, offering deeper insights into unclassified genetic functions. The LSTM-based approach surpasses conventional RNNs by effectively mitigating the vanishing gradient problem, enabling more efficient long-term information processing in complex biological sequence analysis. Figure 3 is the RNN architecture used in this study.

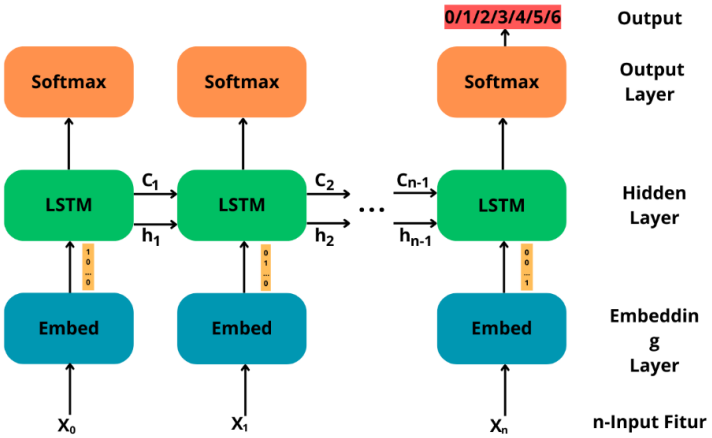

Figure 3.
Architecture of an LSTM Network Within an RNN.

Figure 3 illustrates the architecture of an LSTM network within an RNN designed for DNA sequence analysis. This model consists of several key layers, starting with the Input Layer, represented by symbols, which correspond to nucleotide sequences encoded in numerical form. Next, the Embedding Layer transforms the input into a richer numerical representation, making it easier for the model to process. The Hidden Layer (LSTM Layer) captures long-term sequential dependencies by utilizing memory cells, addressing the vanishing gradient problem commonly found in traditional RNNs, and passing the information to the next unit. The output from the LSTM layer is then processed through the Output Layer (Softmax Layer), which generates probability distributions for each target class, enabling the classification of DNA sequences into specific gene families. The Label Output (0/1/2/3/4/5/6) represents different classification categories, such as G protein-coupled receptors, tyrosine kinases, ion channels, and others. Overall, this model effectively facilitates the analysis of DNA sequences in bioinformatics research by capturing long-term patterns in sequential data.

The LSTM process in RNN involves the input gate storing new information, the forget gate removing irrelevant data, and the output gate generating outputs based on the updated cell state. This mechanism enables the model to capture long-term patterns in DNA sequences, mitigate the vanishing gradient problem, and improve biological data analysis accuracy [21]. Furthermore, numerical DNA representations are processed through an embedding layer, analyzed by LSTM, and classified using a softmax layer to identify specific gene families [22]. The input gate i_t controls the new information that will be stored in the cell state. This is achieved by filtering the information using a sigmoid activation function, which returns values between 0 and 1. The mathematical equation for the input gate is:

$$i_t = \sigma\left(W_{i[h_t-1,x_t]} + b_i\right) \quad (2)$$

Where i_t represents the output of the input gate, σ is the sigmoid activation function, and W_i denotes the weight associated with the input gate. The term h_{t-1} corresponds to the output from the previous time step, while X_t represents the input at the current time step t. Additionally, b_i serves as the bias for the input gate, helping to adjust the activation function's response.

The forget gate (f_t) determines which information should be removed from the cell state. It also utilizes a sigmoid activation function to decide the extent to which information should be forgotten. The mathematical equation for the forget gate is:

$$f_t = \sigma\left(W_f[h_t - 1, x_t] + b_f\right) \quad (3)$$

Where f_t represents the output of the forget gate, $W_f$is the weight associated with the forget gate, and b_f is the bias that helps adjust the activation function's response. This mechanism ensures that the model selectively retains or discards information from the cell state, optimizing long-term sequence processing.

The output gate (o_t) generates the final output based on the cell state and previously processed information. It regulates how much of the updated cell state contributes to the next hidden state. The mathematical equation for the output gate is:

$$o_t = \sigma(W_o[h_t - 1, x_t] + b_o) \quad (4)$$

Where o_t represents the output of the output gate, W_o is the weight associated with the output gate, and b_o is the bias that helps regulate the activation function's response. This gate determines the extent to which the current cell state contributes to the hidden state, ensuring that only the most relevant information is passed forward in the sequence.

After determining the values of the three gates, the next step is to update the cell state. This process involves two key steps:

- Calculating the candidate cell state ($C_t^{\sim}$) value, which represents potential new information to be added:

$$C_t^{\sim} = tan\,h(W_c[h_t - 1, x_t] + b_c) \quad (5)$$

- Updating the cell state (C_t) by combining the retained past state with the newly selected information:

$$C_t = f_t \cdot C_t - 1 + i_t \cdot C_t^{\sim} \quad (6)$$

This mechanism allows the model to maintain long-term dependencies while filtering out irrelevant data, ensuring effective sequential learning.

Finally, the final output (h_t) is calculated using the cell state and the output gate:

$$h_t = o_t \cdot tanh(C_t) \quad (7)$$

With the LSTM architecture incorporating three main gates, the model can effectively store and manage long-term information while filtering out irrelevant data during the learning process. This enables LSTM to capture sequential dependencies more efficiently, making it well-suited for tasks involving long-range patterns in time-series and sequence data

2.4. Pseudo-Labelling

Pseudo-labelling is a semi-supervised learning technique used to enhance the performance of LSTM-RNN models in DNA sequence classification by leveraging unlabeled data [23]. The process begins with a pre-trained model predicting labels for unlabeled data, formulated as:

$$\hat{y}_i = f(x_i; \theta) \quad (8)$$

where $\hat{y}_i$ represents the predicted label for the DNA sequence x_i. Next, pseudo-labels are assigned if the confidence score exceeds 90%, calculated using the softmax function:

$$P(y = j|x_i) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad (9)$$

where $z_j$is the logit for class j. If the confidence score meets the threshold, pseudo-labels are assigned based on:

$$\hat{y}_i = \begin{cases} label_{predicted} \\ no\ label \end{cases}, if \quad P\big(y = j(x_i)\big) > 0.9 \quad (10)$$

The combined dataset is then $D_{combined} = D_{label} \cup D_{pseudo}$, where $D_{pseudo} = \{(x_j, \hat{y}_j)\}$. The LSTM-RNN model is retrained using this expanded dataset with the loss function:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\big(f(x_i; \theta)\big) \quad (11)$$

Where represents the total number of data points in the combined dataset. This iterative process continues until accuracy or F1-score stabilizes, enabling the LSTM-RNN model to capture sequential patterns in genomic data more effectively and improve generalization in genetic classification.

2.5. Model Evaluation

Several model evaluation metrics are used in this study to assess the performance of RNN-LSTM in classifying chimpanzee DNA sequences with the Pseudo-Labelling technique.

2.5.1. Accuracy

One key metric is accuracy, which measures the percentage of correct predictions out of total predictions. A high accuracy indicates the model's ability to effectively recognize genetic patterns, even in complex sequences [24]. By incorporating Pseudo-Labelling, the model learns more efficiently from unlabeled data, enhancing overall classification performance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

Where TP and TN represent correct predictions for positive and negative cases, while FP and FN indicate incorrect predictions. These metrics assess the RNN-LSTM model's classification performance in DNA sequence analysis.

2.5.2. Loss Function

The Loss Function, specifically Cross-Entropy Loss, measures the prediction error of the RNN-LSTM model in classifying chimpanzee DNA sequences [25]. It quantifies the difference between predicted probabilities and true labels, ensuring the model effectively distinguishes genetic patterns across categories like G protein-coupled receptors or Tyrosine kinase. A lower loss value indicates improved classification performance. With Pseudo-Labelling, the model learns from unlabeled data, leading to a gradual loss reduction. The Loss Curve shows a stable decline, demonstrating effective parameter optimization and improved accuracy without overfitting [26]. The mathematical formula for Cross-Entropy Loss in multi-class classification is:

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \quad (13)$$

Where N represents the total number of samples, y_i is the actual class label (1 if correct, 0 otherwise), and $\hat{y}_i$ is the predicted probability for the class i. This ensures that the model learns to minimize the difference between actual and predicted labels, improving classification accuracy in DNA sequence analysis.

2.5.3. ROC (Receiver Operating Characteristic) Curve

The ROC (Receiver Operating Characteristic) Curve illustrates the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) across different classification thresholds. The AUC (Area Under the Curve) measures the model's ability to distinguish between positive and negative classes, with higher values (closer to 1) indicating better classification performance. AUC is computed as the integral area under the ROC curve, providing a comprehensive

evaluation of the model's discrimination power [27]. A higher AUC value signifies that the model effectively differentiates between classes, making it a crucial metric in assessing the performance of DNA sequence classification using RNN-LSTM. The Formula is:

$$TPR = \frac{TP}{TP + FN} \tag{14}$$

$$FPR = \frac{FP}{FP + TN} \tag{15}$$

$$AUC = \int_0^1 TPR(FPR)\ d(FPR) \tag{16}$$

Equation 14 defines the True Positive Rate (TPR) as a measure of the model's sensitivity, while Equation 15 calculates the False Positive Rate (FPR) to assess misclassification of negative samples. Equation 16 describes the Area Under the Curve (AUC) as the integral of TPR over FPR, reflecting the model's ability to distinguish between positive and negative classes. The higher the AUC, the better the classification performance.

2.5.4. Precision-Recall Curve

The Precision-Recall Curve is used to evaluate model performance, especially in imbalanced datasets. Precision measures the accuracy of positive predictions, while Recall assesses how well the model identifies actual positive samples. Average Precision (AP) represents the mean precision across different recall levels [28]. A higher AP value indicates that the model effectively balances precision and recall, maintaining high precision even as recall increases. This metric is particularly useful for assessing classification performance when class distribution is skewed, ensuring the model's reliability in DNA sequence analysis using RNN-LSTM. The formula to get AP is.

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

$$AP = \int_0^1 Precision(Recall)\ d(Recall) \tag{19}$$

Equation 17 defines Precision as the ratio of correctly predicted positive samples to all predicted positives, while Equation 18 represents Recall as the proportion of actual positives correctly identified. Equation 19 calculates Average Precision (AP) as the integral of Precision over Recall, reflecting the model's ability to balance accuracy and completeness, particularly in imbalanced datasets.

3. Results and Discussion

In this study, the data was sourced from two public repositories, Kaggle and Ensemble Genome Browser, which provide chimpanzee DNA sequence datasets in both labelled and unlabeled forms. The labelled data consists of DNA sequences classified based on specific gene families, such as G protein-coupled receptors, tyrosine kinase, tyrosine phosphatase, synthetase, synthase, ion channels, and transcription factors. Each label represents a specific protein produced by the DNA sequence, serving as the basis for training the Recurrent Neural Network - Long Short-Term Memory (RNN-LSTM) model. The collected labelled data is sufficient to ensure a balanced category distribution, allowing the model to effectively recognize genetic patterns. The dataset details are presented in Table 1.

Table 1.
Characteristics of the Chimpanzee DNA Sequence Dataset.

No	*Sequence*	Class
1.	ATGCCCC…G	4
2.	ATGAACGAA…A	4
...	…	…
1.682	ATGTTG…A	3

The dataset in Table 1 consists of 1,682 rows with two columns: Sequence and Class, where the DNA sequence length varies between 200 and 2,000 nucleotides. This length range has been verified to comply with genomic standards. With sufficient data and an appropriate format, this study aims to develop an accurate RNN-LSTM-based protein prediction model and enhance classification performance through pseudo-labelling.

3.1. Preprocessing Data Sequence

The modelling process in this study begins with a preprocessing stage to convert DNA sequences into a numerical format that can be processed by the model. The preprocessing steps include One-Hot Encoding, which converts nucleotide characters into numerical representations, and k-mer embedding with $k = 3$, where sequences are divided into consecutive substrings (k-mers) and transformed into vectors using CountVectorizer. The k-mer embedding results are then processed with padding using pad_sequences to ensure uniform vector length, with a maximum length of 100. Finally, features from One-Hot Encoding and k-mer embedding are combined into a single matrix to ensure an optimal representation of the DNA sequence data before being used for model training. The results of this preprocessing stage are presented in Table 2.

Table 2.
Results of Pre-Processed Data.

Sequence	One-Hot Encoding	K-mer Embedding (k=3)	Padded K-mer Embedding
ATGCGT	[1, 0, 0, 0, 0, 0, 1, 0, ...]	[ATG, TGC, GCG, CGT] → [3, 5, 2, 4]	[3, 5, 2, 4, 0, 0, 0, ...]
CGTACA	[0, 1, 0, 0, 1, 0, 0, 1, ...]	[CGT, GTA, TAC, ACA] → [4, 6, 7, 8]	[4, 6, 7, 8, 0, 0, 0, ...]
..	..	..	..
TACGGA	[0, 0, 1, 1, 0, 0, 1, 0, ...]	[TAC, ACG, CGG, GGA] → [7, 9, 10, 11]	[7, 9, 10, 11, 0, 0, 0, ...]

Table 2 presents the process of converting DNA sequences into numerical features, starting with One-Hot Encoding, followed by K-mer Embedding (k=3), which is transformed into numerical vectors, and finally Padded K-mer Embedding, where padding is applied to ensure a uniform length (maximum 100).

3.2. Feature Selection Was Performed Using Lasso Regression

Based on the preprocessing results above, feature selection was performed using Lasso Regression with L1 regularization. Lasso applies a penalty to the absolute values of the coefficients in the loss function, aiming to shrink some coefficients to zero and eliminate irrelevant features. The Lasso loss function can be calculated as shown in Equation (1 Table 3 presents the results of feature selection after applying Lasso Regression, where irrelevant features were removed, and significant features were selected for the subsequent model.

Table 3.
Feature Selection Outcomes for Genomic Classification.

Feature	Coefficient	Selected (1 = Yes, 0 = No)
ACG	534	1
ATC	0	0
...	...	...
TCG	0	0

Table 3 shows a reduction in features from 1775 to 62. Features with near-zero coefficients were removed as they were deemed irrelevant, while significant features were retained for further analysis. This selection process simplifies the model, reduces overfitting risk, and improves efficiency and prediction accuracy

3.3. LSTM-RNN with Pseudo-Labelling

After feature selection to retain relevant attributes, a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) model is applied to analyze and predict genomic data. By leveraging pseudo-labelling, the model can identify patterns in previously unlabeled genomic data, which are typically difficult to use in conventional training. This technique enhances the training dataset by adding predicted labels to unlabeled data, ultimately improving prediction accuracy.

Initially, genomic data is divided into two main groups: labelled data (80%) for training the model and unlabeled data (20%), which receives pseudo-labels after training. The labelled data is further split into training data (70%) and testing data (30%), where the training set is used to optimize model weights through backpropagation with the Adam optimizer, while the test set evaluates performance. The model structure is shown in Table 4.

Table 4.
Structural Configuration of the LSTM-RNN Model.

Layer	Units	Activation	Description
LSTM 1	128	-	return_sequences=True
LSTM 2	64	-	-
Dropout	-	-	Rate = 0.3
Dense	32	ReLU	Fully connected layer
Output Layer	7	Softmax	Probability prediction

In the initial training phase, the model was trained using labelled training data, leveraging the LSTM architecture to recognize patterns in genomic data. The structural configuration of the LSTM-RNN model is presented in Table 4, detailing the number of units, activation functions, and specific layer descriptions used in the architecture. Training was conducted for 50 epochs using the sparse categorical cross-entropy loss function, which is suitable for multi-class classification, and the Adam optimizer with a learning rate of 0.001 to accelerate convergence. This training process aimed to optimize the

model's ability to understand relationships between processed genomic features. Table 5 presents the initial training results, demonstrating the model's effectiveness in identifying patterns within the training data.

Table 5.
Results of Initial Model Training.

Sample ID	Original Data (Features)	True Label	Model Prediction (Label)
1	[0.23, 0.56, 0.98]	Class 1	Class 1
2	[0.12, 0.34, 0.45]	Class 2	Class 2
…	…	…	…
3	[0.45, 0.23, 0.76]	Class 3	Class 3

Table 5 presents the results of the initial model training, where the model's predictions are compared against the true labels for a subset of the training data. This evaluation demonstrates the model's effectiveness in learning patterns from labeled genomic sequences before being applied to unlabeled data. After the initial training, the model is used to predict unlabeled data by generating pseudo-labels. Based on learned patterns, the model assigns a class prediction to each sample without an original label. This process utilizes the argmax function, selecting the class with the highest probability as the pseudo-label. As a result, previously unusable data can now contribute to further model training. Table 6 presents the pseudo-labelling results.

Table 6.
Results of Pseudo-Labeling Approach.

Sample ID	Unlabeled Data (Features)	Model Prediction (Pseudo-Label)
4	[0.67, 0.45, 0.23]	Class 1
5	[0.33, 0.56, 0.89]	Class 2
…	…	..
6	[0.12, 0.54, 0.34]	Class 3

Table 6 presents the results of the pseudo-labeling approach, where the model assigns predicted labels to previously unlabeled data based on learned patterns. These pseudo-labels are then integrated into the training dataset to enhance model performance. After obtaining pseudo-labels, the previously unlabeled data is combined with the labeled training data to enrich the training dataset, allowing the model to be trained on a larger and more diverse dataset. This integration aims to enhance the model's ability to recognize genomic patterns by leveraging information from both types of data. Once the combined dataset is formed, the model is retrained for 20 epochs using reshaped data that matches the model's input dimensions, with a batch size of 32. This retraining process enables the model to utilize more information from both original labelled data and pseudo-labelled data, ultimately improving accuracy and performance in identifying genomic patterns.

3.4. Model Evaluation

The next step involves evaluating the model's performance in predicting unseen data and assessing its generalization capability across a broader dataset. The model's evaluation is conducted by analyzing the accuracy trends for both training and validation data as the number of epochs increases. The expectation is that, as training progresses, the model will better recognize patterns in the training data and generalize them effectively to the validation data. In this study, the evaluation is performed using the training history obtained through the model.fit() method, where training accuracy and validation accuracy are extracted from history. history['accuracy'] and history. history['val_accuracy'], respectively. The accuracy data is then plotted against the number of epochs to observe the model's learning trends. This approach systematically analyzes the model's performance, allowing for the identification of potential overfitting or underfitting issues that could impact its generalization to new data.

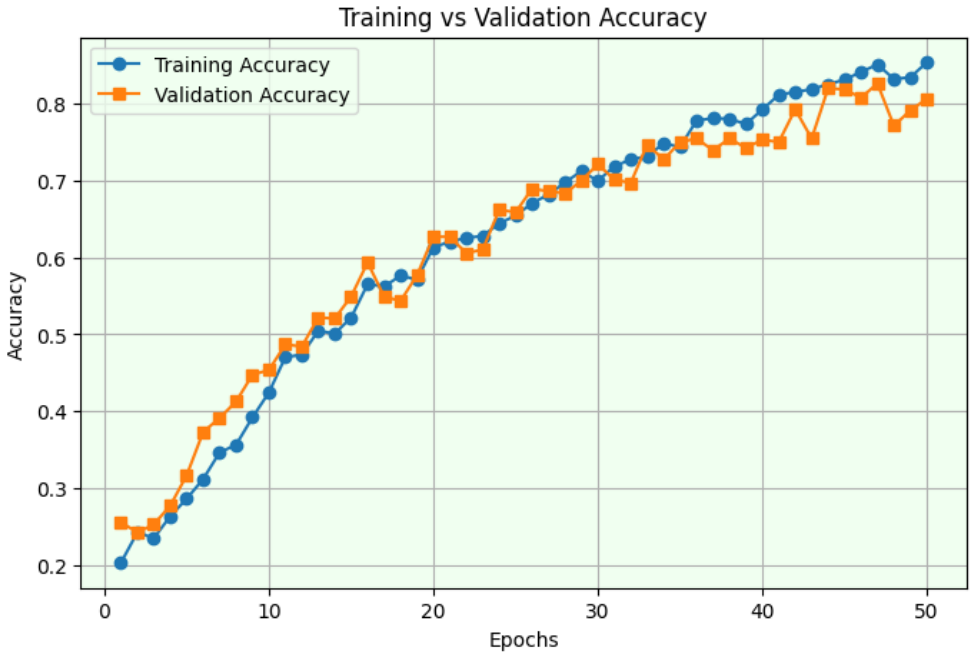

Figure 4.
Training and Validation Accuracy Graph.

Figure 4 illustrates the gradual increase in training accuracy from 20.27% at epoch 1 to 85.41% at epoch 50, while validation accuracy follows a similar trend, starting at 25.57% in epoch 1 and reaching 80.67% at epoch 50. By epoch 10, training accuracy reaches 42.44%, while validation accuracy is at 45.36%, indicating that the model has begun to effectively learn patterns within the data. At epoch 30, training accuracy reaches 69.98%, with validation accuracy at 72.15%, signifying model stability. Despite minor fluctuations in validation accuracy, the overall trend continues to rise without clear signs of overfitting, as validation accuracy remains aligned with training accuracy.

Following the accuracy evaluation, further assessment is conducted using the ROC Curve and AUC to measure the model's ability to differentiate between classes. The ROC Curve examines the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) across various classification thresholds, while AUC quantifies how well the model distinguishes between classes. In this study, AUC calculation involves first binarizing labels using label_binarize(), making it applicable to multi-class classification. The AUC score is then computed using roc_auc_score() with a one-vs-rest (OVR) approach, allowing a comprehensive evaluation of the model's performance in distinguishing each class. This analysis provides deeper insights into model prediction quality beyond accuracy alone, helping to identify areas for further improvement.

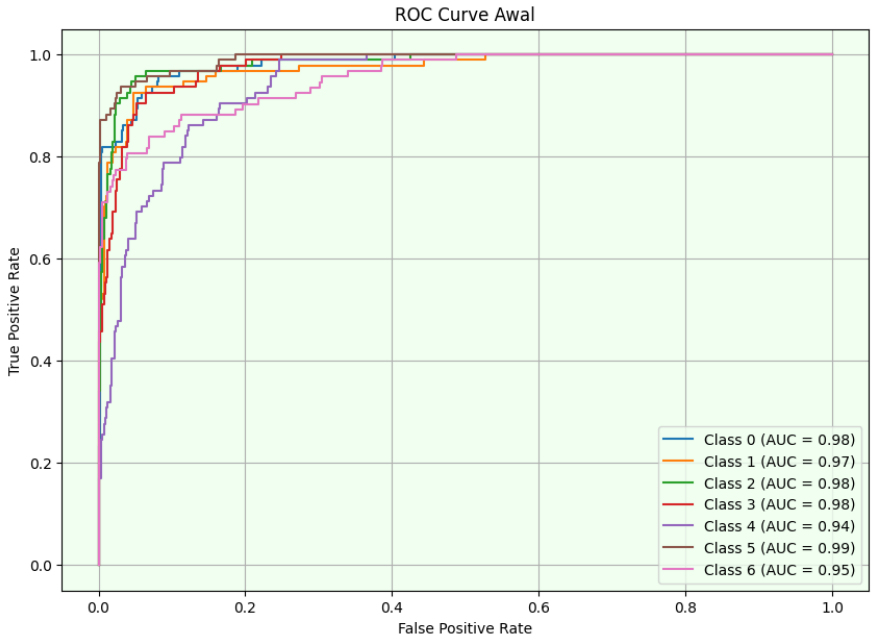

Figure 5.
Initial Results of the ROC Curve Analysis.

Figure 5 illustrates that the model demonstrates strong classification capability, with AUC values ranging from 0.94 to 0.99 for each class. A higher AUC value indicates better differentiation between positive and negative classes. The ROC curve mostly lies near the upper-left corner, signifying a high True Positive Rate (TPR) and a low False Positive Rate (FPR). This suggests that the model effectively classifies samples with minimal errors, although slight variations in performance across different classes are observed.

While the ROC Curve and high AUC values indicate strong classification performance, additional evaluation is conducted through the Loss Curve and Accuracy Curve to ensure training stability and prevent overfitting. The Loss Curve tracks changes in loss values during training and validation, while the Accuracy Curve displays accuracy trends for both datasets. These evaluations utilize training history data, including train_loss, val_loss, train_acc, and val_acc, with the optimal epoch determined using np.argmin(val_loss) + 1. This analysis ensures that the model is assessed not only based on classification performance but also on the stability of its learning process.

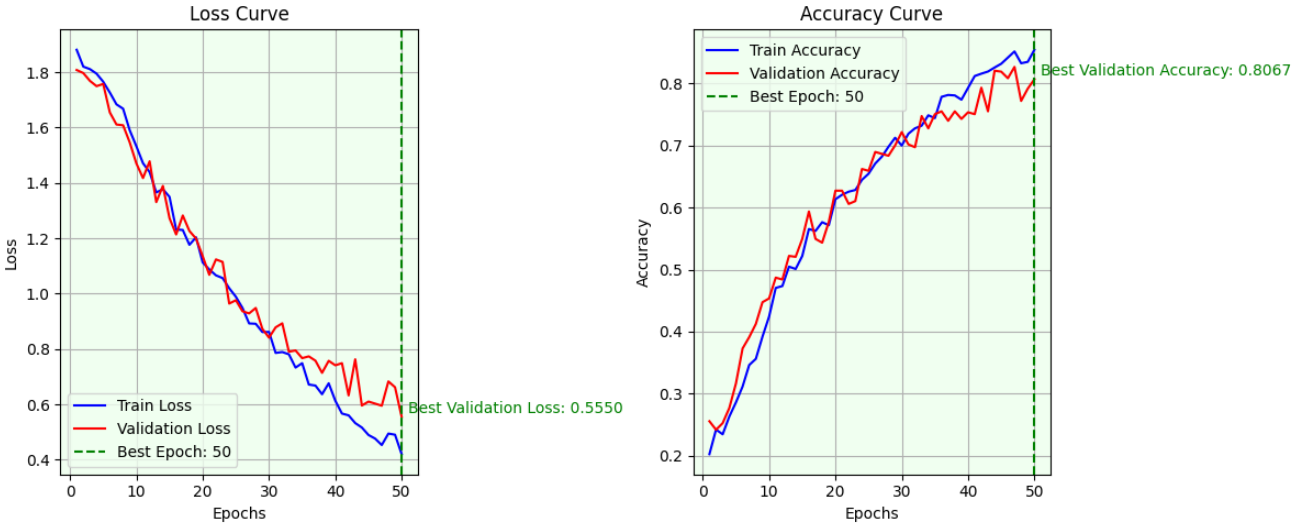

Figure 6.
Loss and Accuracy Curves.

Analyzing Figure 6 allows us to assess whether the model experiences overfitting. Based on the Loss Curve and Accuracy Curve, the model demonstrates a stable learning trend, with a decreasing loss and increasing accuracy as the number of epochs progresses. The optimal epoch is determined at epoch 50, where the validation loss reaches 0.5550 and the validation accuracy reaches 0.8067. The accuracy curve indicates a gradual improvement in performance, while the loss curve shows that the model successfully minimizes errors. The alignment between training and validation loss, as well as training and validation accuracy, suggests that the model does not suffer from overfitting and generalizes well to validation data.

Following the evaluation through the Loss Curve and Accuracy Curve, a Precision-Recall Curve analysis is conducted to assess the model's ability to handle imbalanced datasets. Accuracy alone is insufficient to determine whether the model effectively classifies minority classes. The Precision-Recall Curve illustrates the trade-off between precision and recall across different classification thresholds, while Average Precision (AP)—calculated as the area under the curve—evaluates the balance between the two. Using precision_recall_curve() and average_precision_score(), this evaluation helps determine whether the model accurately identifies classes with fewer samples.

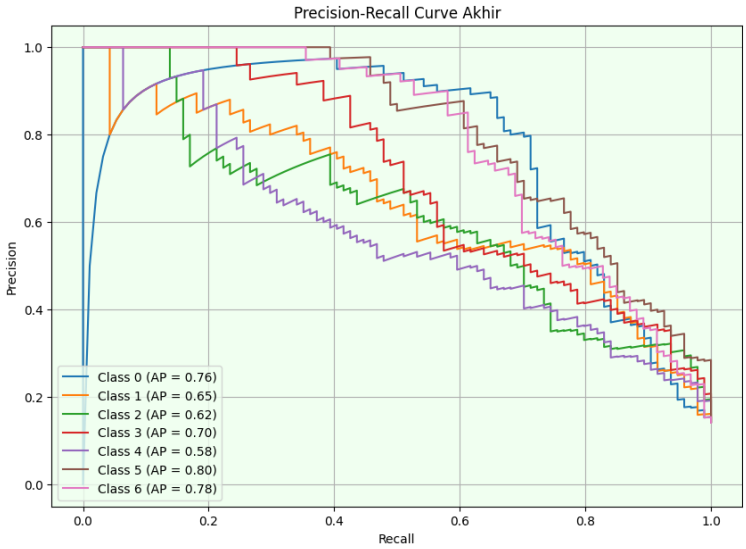

Figure 7.
Precision-Recall Curve Analysis Result.

Figure 7 shows that the Average Precision (AP) values range between 0.58 and 0.80. Classes with higher AP values, such as class 5 (AP = 0.80) and class 6 (AP = 0.78), demonstrate better precision retention as recall increases, indicating that the model is more consistent in identifying samples from these classes. Conversely, class 4 (AP = 0.58) exhibits lower performance, suggesting that the model struggles to maintain precision as recall increases. Overall, higher AP values indicate that the model effectively manages the trade-off between precision and recall for specific classes.

The discussion of these results highlights the effectiveness of the RNN-LSTM model with pseudo-labeling in improving chimpanzee DNA sequence classification accuracy, demonstrating its ability to achieve high performance despite data imbalance. The model's stability, indicated by converging accuracy and loss curves, as well as a well-balanced precision-recall curve, confirms its reliability in genetic pattern recognition. Compared to traditional CNN-based methods, which are more susceptible to noise, the semi-supervised learning approach successfully utilizes unlabeled data to enhance generalization. The integration of LSTM, pseudo-labeling, and L1 Regularization has proven to be a valuable strategy in genomic classification tasks. These findings emphasize the model's potential applications in bioinformatics and healthcare, particularly in disease-related genetic pattern identification, evolutionary studies, and gene-based therapies. Additionally, the proposed approach can be extended to other species' genomic studies with limited labeled data, contributing to global health advancements through artificial intelligence-driven genomic research, aligning with the Sustainable Development Goals (SDGs).

4. Conclusion

This study implemented an RNN-LSTM model with Pseudo-Labelling to classify chimpanzee DNA sequences obtained from Kaggle and the Ensemble Genome Browser. The dataset included labeled DNA sequences categorized into various gene families, such as G protein-coupled receptors, tyrosine kinases, and others. Pseudo-Labelling was applied to utilize unlabeled data by predicting its class probabilities using the initial model and incorporating them into training to enhance model generalization. Model evaluation demonstrated strong performance in recognizing genetic patterns. Training accuracy increased from 20.27% at epoch 1 to 85.41% at epoch 50, while validation accuracy followed a similar trend, rising from 25.57% to 80.67%. At epoch 10, training accuracy reached 42.44%, and validation accuracy was 45.36%, indicating that the model had started learning meaningful patterns. By epoch 30, training accuracy had reached 69.98%, and validation accuracy was 72.15%, signifying model stability. Despite minor fluctuations in validation accuracy, the overall trend continued to rise, with no clear signs of overfitting. The ROC curve confirmed strong classification capability, with AUC values ranging from 0.94 to 0.99, indicating that the model effectively differentiates between positive and negative classes with minimal error. The loss and accuracy curves demonstrated stable learning behavior, with validation loss reaching 0.5550 at epoch 50, reinforcing that the model successfully minimizes errors while maintaining generalization. The Precision-Recall Curve analysis showed that the Average Precision (AP) values ranged between 0.58 and 0.80, with higher AP values in certain classes (e.g., class 5 at 0.80 and class 6 at 0.78), demonstrating better precision retention as recall increased. Conversely, lower AP values (e.g., class 4 at 0.58) suggested challenges in maintaining precision. Overall, the results indicate that the combination of RNN-LSTM and Pseudo-Labelling enhances classification accuracy, improves generalization, and effectively handles imbalanced datasets. This approach has significant potential for genetic pattern identification in DNA sequences, especially when labeled data is scarce, contributing to advancements in bioinformatics, disease-related genetic research, evolutionary studies, and personalized medicine.

References

[1] A. Calcino *et al.*, "Harnessing genomic technologies for one health solutions in the tropics," *Globalization and Health*, vol. 20, no. 1, pp. 1-6, 2024. https://doi.org/10.1186/s12992-024-01083-3

[2] Z. Ahmed, S. Zeeshan, D. Mendhe, and X. Dong, "Human gene and disease associations for clinical-genomics and precision medicine research," *Clinical and Translational Medicine*, vol. 10, no. 1, pp. 297-318, 2020. https://doi.org/10.1002/ctm2.28

[3] N. Sapoval *et al.*, "Current progress and open challenges for applying deep learning across the biosciences," *Nature Communications*, vol. 13, no. 1, pp. 1–12, 2022. https://doi.org/10.1038/s41467-022-29268-7

[4] F. Mora-Bermúdez *et al.*, "Differences and similarities between human and chimpanzee neural progenitors during cerebral cortex development," *Elife*, vol. 5, p. e18683, 2016. https://doi.org/10.7554/eLife.18683

[5] M. V. Suntsova and A. A. Buzdin, "Differences between human and chimpanzee genomes and their implications in gene expression, protein functions and biochemical properties of the two species," *BMC Genomics*, vol. 21, no. Suppl 7, pp. 1–12, 2020. https://doi.org/10.1186/s12864-020-06962-8

[6] C. Li *et al.*, "Roles and mechanisms of exosomal non-coding RNAs in human health and diseases," *Signal Transduction and Targeted Therapy*, vol. 6, no. 1, pp. 1–31, 2021. https://doi.org/10.1038/s41392-021-00779-x

[7] B. Rhead, P. E. Haffener, Y. Pouliot, and F. M. De La Vega, "Imputation of race and ethnicity categories using genetic ancestry from real-world genomic testing data," *Pacific Symposium on Biocomputing*, vol. 29, pp. 433–445, 2024. https://doi.org/10.1142/9789811286421_0033

[8] A. Khan, M. A. Shaaban, and M. H. Khan, "Improving pseudo-labelling and enhancing robustness for semi-supervised domain generalization," presented at the Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP 2024), SciTePress, 2024.

[9] K. W. Goh *et al.*, "Comparison of activation functions in convolutional neural network for poisson noisy image classification," *Emerging Science Journal*, vol. 8, no. 2, pp. 592-602, 2024. https://doi.org/10.28991/ESJ-2024-08-02-019

[10] L. Koumakis, "Deep learning models in genomics; are we there yet?," *Computational and Structural Biotechnology Journal*, vol. 18, pp. 1466-1473, 2020. https://doi.org/10.1016/j.csbj.2020.06.017

[11] P. J. Canatalay and O. N. Ucan, "A bidirectional LSTM-RNN and GRU method to exon prediction using splice-site mapping," *Applied Sciences*, vol. 12, no. 9, p. 4390, 2022. https://doi.org/10.3390/app12094390

[12] M. Praneesh, N. Nivetha, S. S. Maidin, and W. Ge, "Optimized deep learning method for enhanced medical diagnostics of polycystic ovary syndrome detection," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1399-1411, 2024. https://doi.org/10.47738/jads.v5i3.368

[13] A. Yang, W. Zhang, J. Wang, K. Yang, Y. Han, and L. Zhang, "Review on the application of machine learning algorithms in the sequence data mining of DNA," *Frontiers in Bioengineering and Biotechnology*, vol. 8, pp. 1–13, 2020. https://doi.org/10.3389/fbioe.2020.01032

[14] P. S. Hossain and K. Kim, "Multi-label deep learning models for virus genome DNA sequence classification," Master's Thesis, Bio-AI Convergence Graduate School, Chungnam National University, Daejeon, South Korea, 2023.

[15] A. El-Tohamy, H. A. Maghwary, and N. Badr, "A deep learning approach for viral DNA sequence classification using genetic algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 8, pp. 530–538, 2022. https://doi.org/10.14569/IJACSA.2022.0130861

[16] Z. Du, Y. He, J. Li, and V. N. Uversky, "Deepadd: Protein function prediction from k-mer embedding and additional features," *Computational Biology and Chemistry*, vol. 89, p. 107379, 2020. https://doi.org/10.1016/j.compbiolchem.2020.107379

[17] A. Lopez-del Rio, M. Martin, A. Perera-Lluna, and R. Saidi, "Effect of sequence padding on the performance of deep learning models in archaeal protein functional prediction," *Scientific Reports*, vol. 10, no. 1, p. 14634, 2020. https://doi.org/10.1038/s41598-020-71450-8

[18] Z. Chen *et al.*, "Feature selection may improve deep neural networks for the bioinformatics problems," *Bioinformatics*, vol. 36, no. 5, pp. 1542-1552, 2020. https://doi.org/10.1093/bioinformatics/btz763

[19] M. Abraruddin, J. G. Prasath, and M. Tech Scholar, "Early prediction and risk analysis of type 2 diabetes mellitus using the nonlinear least absolute shrinkage and selection operator (LASSO) regression technique," *Journal of Emerging Technologies and Innovative Research*, vol. 7, no. 7, pp. 203–210, 2020. https://doi.org/10.17229/Journal.24022

[20] A. Alshammari, "Ensemble recurrent neural network with whale optimization algorithm-based DNA sequence classification for medical applications," *Soft Computing*, pp. 1-14, 2023. https://doi.org/10.1007/s00500-023-08435-y

[21] H. Iuchi *et al.*, "Representation learning applications in biological sequence analysis," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 3198-3208, 2021. https://doi.org/10.1016/j.csbj.2021.05.039

[22] G.-S. Han, Q. Li, and Y. Li, "Nucleosome positioning based on DNA sequence embedding and deep learning," *BMC Genomics*, vol. 23, no. Suppl 1, pp. 1–10, 2022. https://doi.org/10.1186/s12864-022-08508-6

[23] P. Shaeri and A. Katanforoush, "A semi-supervised fake news detection using sentiment encoding and LSTM with self-attention," in *Proceedings of the 2023 13th International Conference on Computer and Knowledge Engineering (ICCKE), IEEE*, 2023, pp. 590–595.

[24] Ž. Vujović, "Classification model evaluation metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 599-606, 2021. https://doi.org/10.14569/IJACSA.2021.0120670

[25] C. John, J. Sahoo, M. Madhavan, and O. K. Mathew, "Convolutional neural networks: A promising deep learning architecture for biological sequence analysis," *Current Bioinformatics*, vol. 18, no. 7, pp. 537-558, 2023. https://doi.org/10.2174/1574893618666230320103421

[26] Y. Chen, K. Xu, P. Zhou, X. Ban, and D. He, "Improved cross entropy loss for noisy labels in vision leaf disease classification," *IET Image Processing*, vol. 16, no. 6, pp. 1511-1519, 2022. https://doi.org/10.1049/ipr2.12402

[27] F. S. Nahm, "Receiver operating characteristic curve: Overview and practical use for clinicians," *Korean Journal of Anesthesiology*, vol. 75, no. 1, pp. 25-36, 2022. https://doi.org/10.4097/kja.21209

[28] C. K. Williams, "The effect of class imbalance on precision-recall curves," *Neural Computation*, vol. 33, no. 4, pp. 853-857, 2021. https://doi.org/10.1162/neco_a_01362