Modeling Cellular Differentiation Using LSTM's: A Comparison of Sequential and Non-Sequential Neural Networks

Paavan Bajaj

August 26, 2025

Abstract

To understand how cells become specialized, it is vital to know how their genes turn on and off over time. The main challenge is that biological data often consists of only a few snapshots in time, making it difficult to create predictive models. This paper uses a type of deep learning called a Long Short-Term Memory (LSTM) network to address this. We set up the prediction task in two ways: (1) predicting how individual genes will behave based on learned patterns, and (2) forecasting the entire future state of a cell from its earlier history and timepoints. Using a public dataset, we tested the LSTM against a simpler neural network and found that the LSTM was far more effective, particularly for the more challenging forecasting task. This work offers a guide for how to apply and test different computer-based methods to learn about biology from limited time-series data.

1 Introduction

The growth of a living cell is a key process in biology, where it changes from a flexible state into a specific type of cell. This process, called *cell differentiation*, happens through a series of carefully controlled changes in gene activity over time. Understanding this process is important for studying how organisms grow, how diseases develop, and how doctors might one day use cells to repair tissues. But studying this process is difficult because current technology can only take separate snapshots of gene activity at different times, which makes the data incomplete.

To help with problems like this, researchers use a tool called a *neural network*, which is a computer model inspired by the way the human brain works. Neural networks are made up of layers of "neurons" that take in data, process it, and pass it forward, allowing the model to find patterns that are too complex for humans to see directly. They have been used successfully in areas like image recognition, language translation, and speech recognition. A special type of neural network, called a *Long Short-Term Memory (LSTM)* network, is designed to work with data that comes in sequences. LSTMs can "remember" information from earlier steps and use it to make better predictions later, which makes them especially good at analyzing processes that change over time. This makes them an excellent choice for studying how cells develop across different time points. The figure below shows a simple example of how a basic neural network is structured with an input layer, hidden layers and the final output layer.

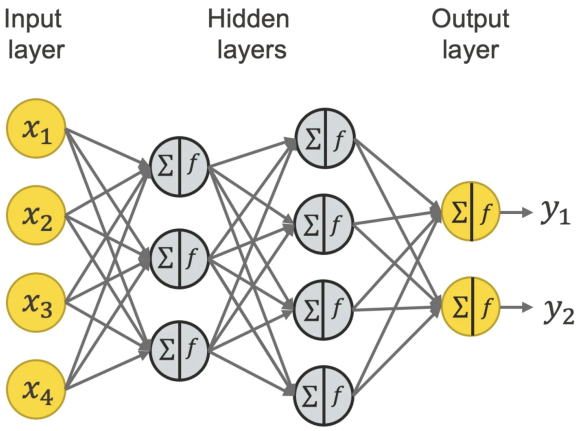

Figure 1: A basic nueral network architecture with input, hidden and output layers.

The main problem is figuring out patterns and making predictions from limited information. Can computer models learn the rules that control differentiation and use them to predict what cells will look like later? This paper explores this by using *Long Short-Term Memory (LSTM) neural networks* on real human stem cell data. LSTMs are good at working with sequence data, such as predicting words in a sentence or forecasting time series, because they can remember earlier information while making new predictions. This makes them well-suited to studying how cells change over time.

Our dataset is structured as a matrix where rows represent *genes* and columns represent *cells* at different times. From this, we tested two approaches:

1. *Gene-Level Prediction:* The model learns from patterns in many genes and then tries to predict the behavior of new genes it has not seen before. This looks for shared rules in how genes change over time.

2. *Cell-State Forecasting:* The harder approach uses data from earlier stages (thousands of genes at once) to predict the full state of a cell at a later time. This tests whether the model can forecast how the entire system develops.

By comparing the LSTM with a simpler feedforward neural network, this study shows how important both model design and experiment design are for getting accurate results from limited time-series biological data.

2 Materials and Methods

2.1 Dataset

This study used a public dataset that followed human embryonic stem cell development over 96 hours (Chu et al., 2016). Data was collected at 6 time points: 0, 12, 24, 36, 72, and 96 hours. In total, the dataset had gene expression values for 19,097 genes across 758 cells. Most cells expressed between 8,000–10,000 genes, though this varied depending on their stage of development.

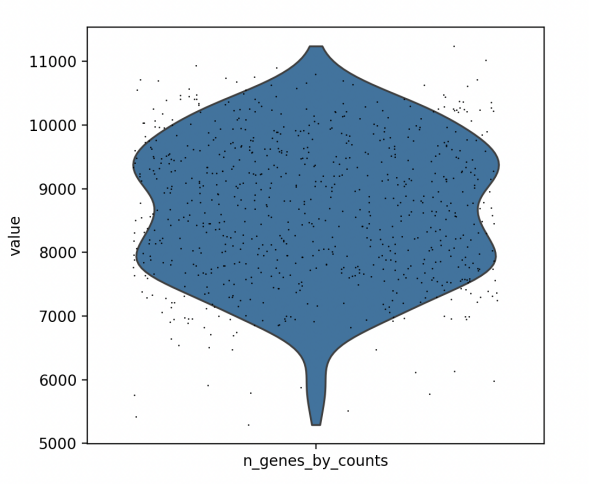

Figure 2: Violin plot showing the distribution of the number of genes expressed per cell.

2.2 Data Preprocessing

Single-cell RNA sequencing data needs cleaning and processing before it can be used. Cells expressing fewer than 500 genes were removed, and genes expressed in fewer than 3 cells were also removed because they did not provide enough useful information.

Normalization was applied so that very highly expressed genes did not dominate the results. A standard log-transformation was used:

$$x_{\text{norm}} = \log(x_{\text{raw}} + 1) \tag{1}$$

After processing, the final dataset included the 2,000 most variable genes across all 758 cells.

2.3 Experimental Design

The way data was split depended on the task.

For *Gene-Level Prediction*, genes were divided into training (70%), validation (15%), and testing (15%). The model used 5 time points as input to predict the last time point for genes it had not seen before.

For *Cell-State Forecasting*, models were trained only on the early time points (0–36 hours) and tested on their ability to predict the full gene expression profile at 96 hours, which was left out of training.

2.4 Neural Network Architectures

Two types of models were tested.

The *Feedforward Network* acted as a baseline. It had three hidden layers with 128, 64, and 32 neurons. It treated the time points as a flat set of inputs without considering their order.

The *Long Short-Term Memory (LSTM) Network* was built for sequence data. It had two layers with 128 and 64 units. It could process time points in order and remember earlier steps to improve predictions. Both models also used Dropout layers (rate = 0.2) to reduce overfitting.

2.5 Training Protocol

Models were trained for up to 250 epochs, with early stopping if the validation results did not improve for 15 epochs. The training used batches of 32 samples and the Adam optimizer. Training results for both models are shown in the following figures.

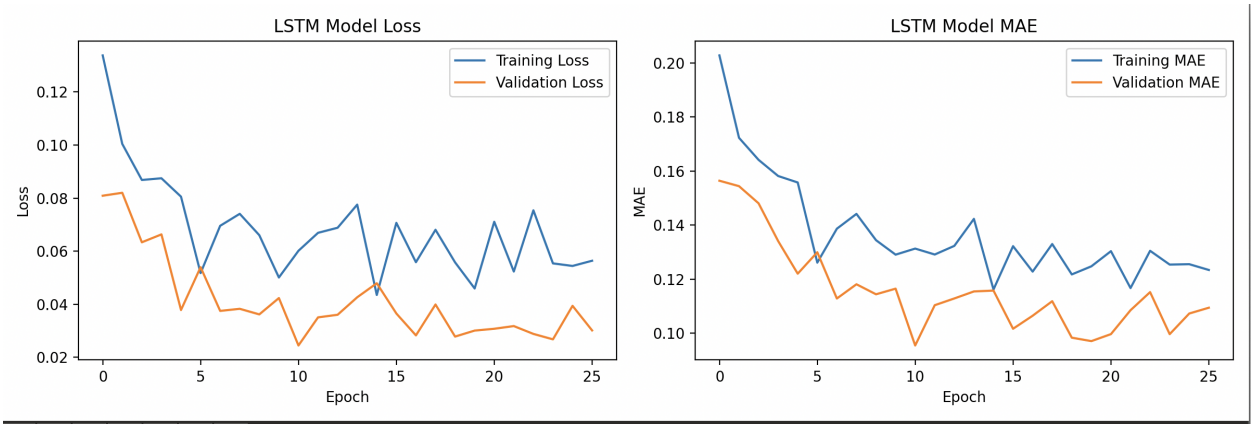

Figure 3: Training performance of the LSTM model for the cell-state forecasting task. The sequential architecture successfully learns from early time points (0–36 hours) to forecast whole-cell expression profiles at 96 hours, demonstrating the model's ability to capture long-term temporal dependencies.

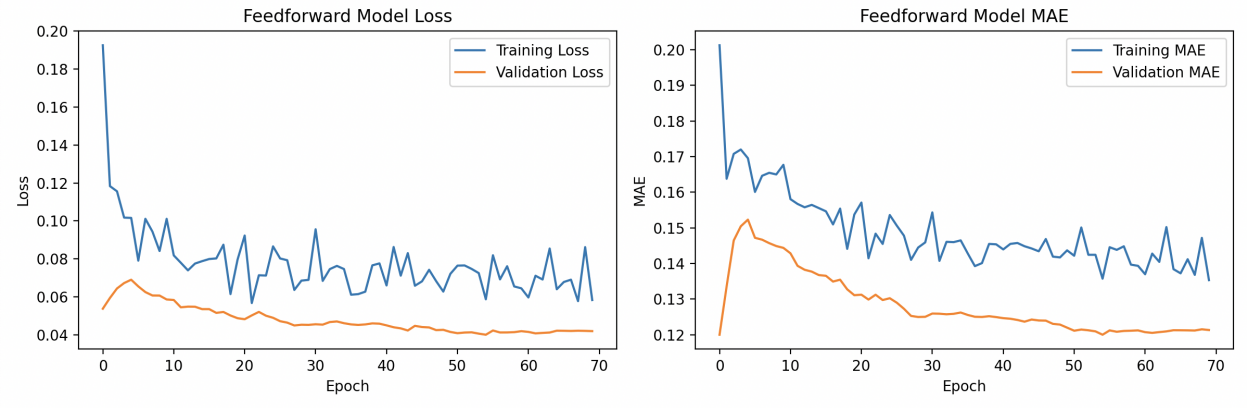

Figure 4: Training performance of the feedforward network for the cell-state forecasting task. Unlike the LSTM, the non-sequential model fails to capture temporal relationships, leading to poor forecasting accuracy for later cell states.

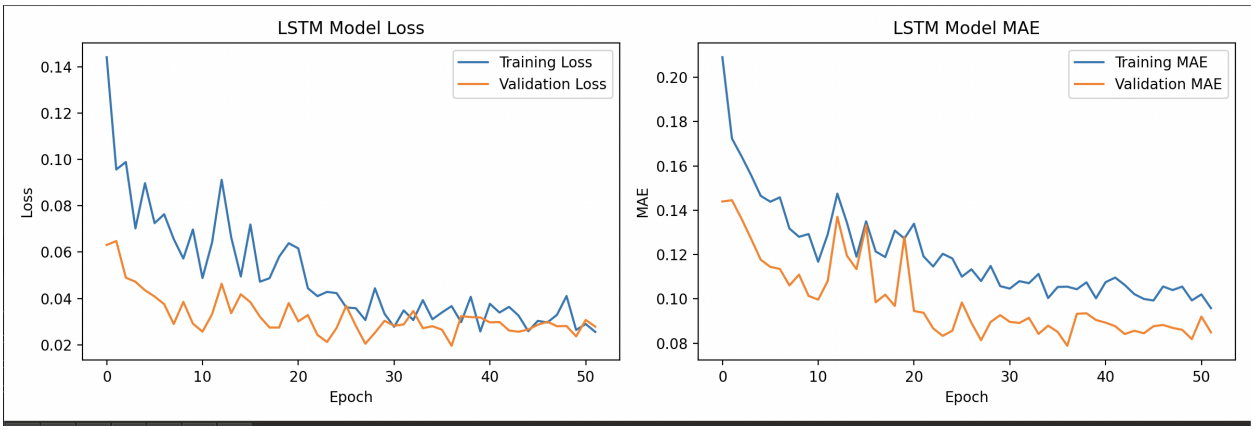

Figure 5: Training performance of the LSTM model for the gene-level prediction task. The sequential model learns temporal expression patterns across genes, enabling accurate predictions for unseen genes.

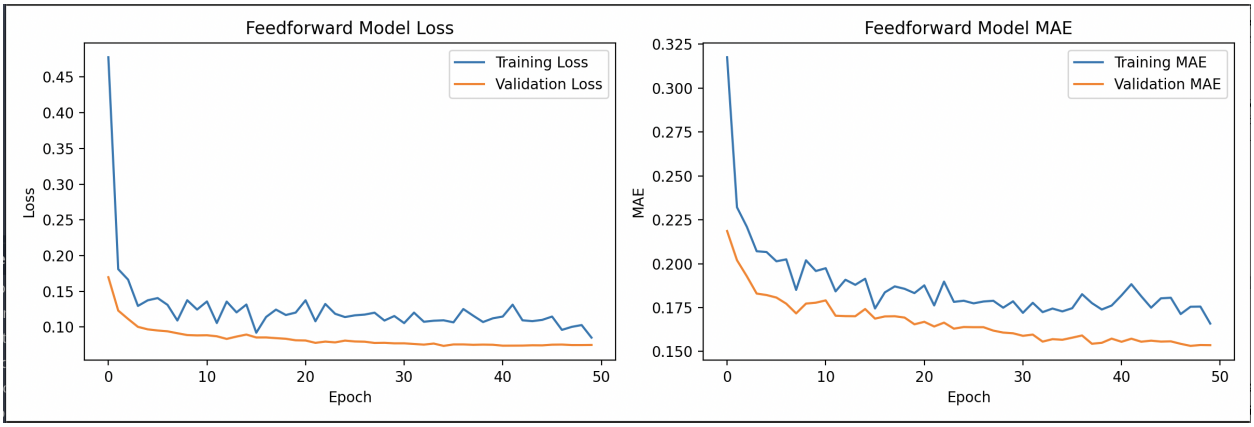

Figure 6: Training performance of the feedforward network for the gene-level prediction task. While able to learn some general patterns, the non-sequential architecture performs less accurately than the LSTM when predicting unseen genes.

2.6 Performance Evaluation

Model performance was measured using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2). Lower RMSE and MAE mean better accuracy, while R^2 closer to 1.0 means the model explains more of the data's variance:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4}$$

3 Results and Discussion

When comparing the LSTM and feedforward networks, the results showed big differences in accuracy. The LSTM consistently performed better on time-series biological data, as shown by lower RMSE/MAE and higher R^2 values (Table 1).

Table 1: Averaged performance metrics for both models across the two data-splitting strategies. Lower RMSE/MAE and higher R^2 indicate better performance.

Splitting Strategy	Model	Avg. RMSE	Avg. MAE	Avg. R^2
Gene-Level Prediction	**LSTM**	**0.095**	**0.060**	**0.858**
	Feedforward	0.130	0.096	0.735
Cell-State Forecasting	**LSTM**	**0.258**	**0.130**	**0.633**
	Feedforward	0.513	0.307	-0.455

3.1 Gene-Level Pattern Recognition

In this task, the models tried to learn from some genes and then predict the patterns of completely new genes.

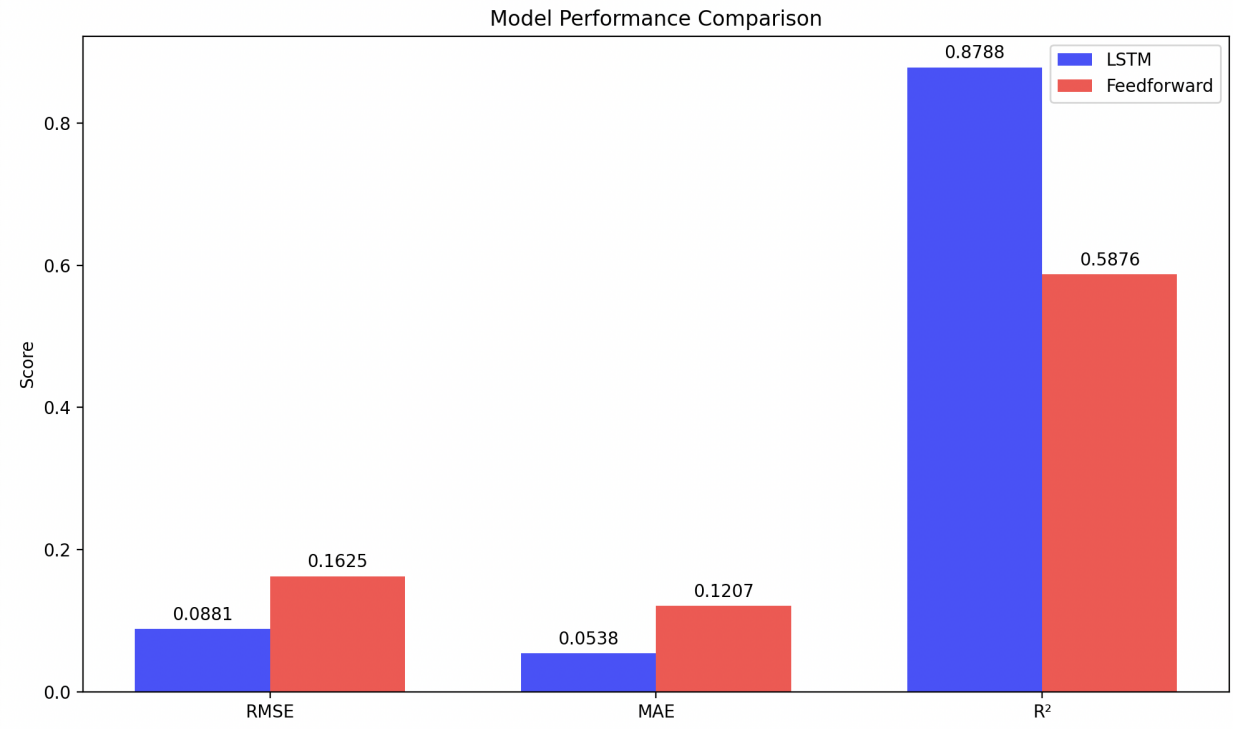

Figure 7: Comparison of training performance between the LSTM and feedforward models for the gene-level prediction task. The LSTM consistently outperforms the feedforward network, capturing temporal gene expression patterns more effectively and yielding higher predictive accuracy on unseen genes.

The LSTM did very well, with an R^2 of 0.8788, meaning it explained about 88% of the changes in unseen genes. The feedforward model reached $R^2 = 0.5876$, showing it was less effective.

These results show two things: (1) there are shared rules in how genes act during differentiation that can be learned by a computer, and (2) LSTMs are much better than feedforward networks because they keep track of the time order of the data.

3.2 Long-term Cellular State Forecasting

This task was harder because it required predicting the full state of cells at 96 hours using only early-stage data.

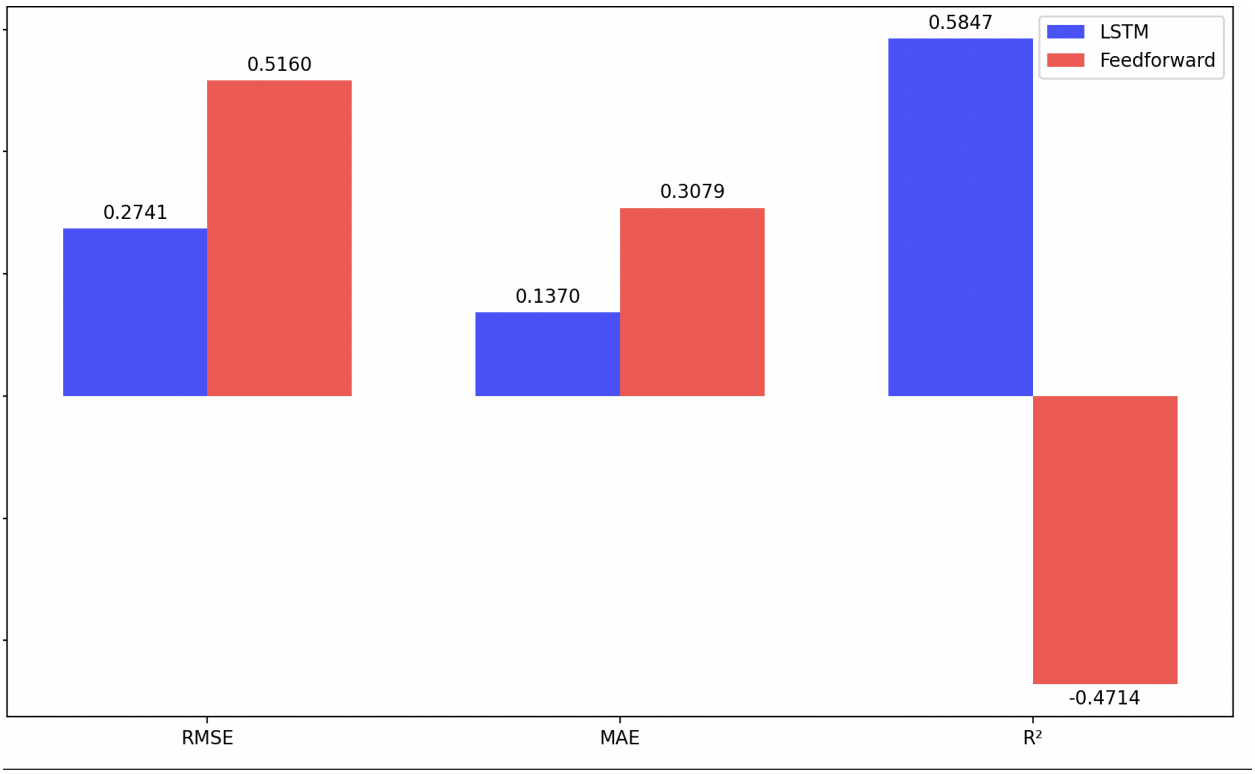

Figure 8: Comparison of training performance between the LSTM and feedforward models for the cell-state forecasting task. The LSTM successfully models long-term dependencies to forecast future cellular states, whereas the feedforward network fails to capture the necessary temporal structure.

The LSTM achieved an R^2 of 0.5847, meaning it was able to make reasonable long-term predictions. The feedforward network, however, failed completely, getting a negative R^2 (-0.4714), which means it did worse than just guessing the mean.

This large difference shows why sequential models are necessary for forecasting in biology. Feedforward networks simply cannot capture the time-based structure needed for this kind of problem.

3.3 Implications for Computational Biology

These results make it clear that both the type of model and the way the experiment is designed matter a lot. The gene-level task helps find common rules in gene regulation, while the cell-state task shows how predictions can be used in real biological research, like predicting cell fates or medical responses.

The strong results from the LSTM suggest it could be useful in medical research, while the failure of the feedforward model is a warning that the wrong type of model can lead to bad predictions.

4 Conclusion

This study shows that Long Short-Term Memory neural networks are effective for modeling how cells change over time, even with limited data. By testing both gene-level prediction and full cell forecasting, the results show that the choice of method and model are both critical.

Key findings include: (1) LSTM models outperform feedforward networks in both tasks, proving that sequential data processing is necessary for biological time-series; (2) shared patterns in gene behavior can be learned and used to predict unseen genes; (3) LSTM models can predict long-term cell states, where feedforward networks completely fail.

For computational biology, this paper gives a clear example of how sequential models should be used on time-series data. For developmental biology, it shows a useful way to make predictions from limited datasets.

Future work could involve using datasets with more time points, testing models that can capture networks of interacting genes, and developing methods that track both individual gene patterns and whole-system changes at the same time.