Football Players Prediction

A Data-Driven Approach Using Long Short-Term Memory and a Bidirectional LSTM(BILSTM).

Submitted by

Karan Patel

Date: May 5,2025

Index

1) Introduction

Football, the most globally followed sport, is not only a game of passion but also a domain of extensive data analytics. Over the years, the integration of data-driven insights has significantly influenced strategies in scouting, player development, and match preparation. Player performance prediction, in particular, has emerged as a critical task for club management, coaches, and analysts aiming to make informed decisions about transfers, team compositions, and long-term investments in players. Accurate performance forecasting can offer a competitive edge by identifying future stars and optimizing team dynamics based on anticipated player trajectories.

In this project, we focus on developing predictive models using **Long Short-Term Memory (LSTM)** and **Bidirectional LSTM (BiLSTM)** neural networks to forecast football players' attributes. These models are particularly well-suited for sequence and time-series data, as they can capture temporal dependencies and evolving trends over time. The dataset used consists of FIFA player statistics from the years **2015 to 2021**, a rich source of structured data encompassing various performance indicators like overall rating, potential, physical stats, technical skills, and more.

FIFA datasets are commonly used in machine learning research due to their well-defined structure, consistency across years, and real-world relevance. In our case, the player records from each year were carefully pre-processed, aligned, and merged to form a time-series dataset for each player. This historical perspective allows the models to learn from past values to make informed predictions about future performance.

The rationale for choosing both LSTM and BiLSTM models lies in their ability to model sequences differently. LSTM processes data in a single direction—typically forward in time—allowing it to learn from previous inputs. BiLSTM, on the other hand, processes sequences in both forward and backward directions, effectively capturing context from both the past and the future within the input window. This bidirectional approach can enhance the model's ability to detect subtle trends and dependencies, which are essential in the complex and dynamic nature of football.

The implementation was carried out in **Google Colab**, leveraging Python libraries such as **TensorFlow**, **Keras**, **Pandas**, and **NumPy**. The project pipeline includes data cleaning, feature engineering, model training, evaluation, and visualization. To ensure robust results, models were trained for **100 epochs**, and performance was measured using evaluation metrics including **Mean Squared Error (MSE)**, **Mean Absolute Error (MAE)**, and **R-squared (R^2)**. Visualization techniques such as predicted vs actual value plots were used to interpret the results more effectively.

This project aims to demonstrate how deep learning can be used not only to model player performance accurately but also to uncover valuable insights about player development over time. By comparing LSTM and BiLSTM architectures, the report also investigates how bidirectionality in sequence modelling impacts predictive accuracy in sports analytics.

Overall, the outcomes of this project can be beneficial to multiple stakeholders—sports analysts, coaching staff, data scientists, and even gaming developers—who rely on predictive modelling for decision-making. The approach also sets a foundation for further extensions, such as incorporating team dynamics, match statistics, or even external data like injuries and transfers to refine predictions.

2) Aim and Scope

Aim

The primary aim of this project is to design and implement robust machine learning models—specifically **Long Short-Term Memory (LSTM)** and **Bidirectional LSTM (BiLSTM)** neural networks—to predict the future performance metrics of professional football players. Using historical player data from FIFA datasets spanning 2015 to 2021, the models are trained to forecast key player attributes by identifying patterns in past performance.

This work aims to demonstrate the effectiveness of sequence modeling in sports analytics, particularly in the context of long-term player development and performance assessment. Through comparative analysis of LSTM and BiLSTM architectures, the project also explores how bidirectional temporal learning can improve predictive accuracy in real-world datasets.

Objectives

The key objectives of this research are as follows:

- To collect and preprocess player data from multiple FIFA datasets to form structured, longitudinal time series per player.
- To implement and train LSTM and BiLSTM models capable of capturing temporal dependencies in the dataset.
- To evaluate the models using statistical metrics such as **Mean Absolute Error (MAE)**, **Mean Squared Error (MSE)**, and **R-squared (R^2)**.
- To visualize predictions against actual values for interpretability and diagnostic purposes.
- To explore potential real-world applications of player performance forecasting in football management, scouting, and sports simulation.

Scope of Research

The scope of this project encompasses the complete lifecycle of a machine learning system—from data acquisition and preprocessing to model development and real-world application. The research focuses on using deep learning for time-series prediction within the sports domain.

The scope includes:

- **Data Preprocessing**
  - Aggregating FIFA player datasets from 2015 to 2021
  - Aligning player records across multiple years using unique identifiers
  - Handling missing or inconsistent data
  - Normalizing input features for efficient learning
- **Model Development**
  - Designing architectures for LSTM and BiLSTM networks using TensorFlow and Keras
  - Configuring hyperparameters (e.g., learning rate, batch size, epochs)
  - Incorporating techniques like dropout and early stopping to reduce overfitting
- **Performance Evaluation**
  - Assessing models using error-based metrics (MAE, MSE) and statistical goodness-of-fit (R^2)
  - Analyzing prediction trends and differences across LSTM and BiLSTM outputs
  - Creating visualization plots to compare predicted vs actual values
- **User Interaction**
  - Structuring model predictions in an interpretable format
  - Designing outputs that can be easily visualized and understood by sports analysts and non-technical stakeholders
  - Enabling experimentation with different model parameters in a flexible Google Colab environment
- **Real-World Application**
  - Using predictions to support scouting and recruitment decisions
  - Providing insights into long-term player potential and growth trajectories
  - Forming a foundation for future integrations with in-game statistics or injury reports for enhanced forecasting

Key Components

The research involves several interrelated components:

1. **Historical Data Collection**
   FIFA player datasets (`players_15.csv` through `players_21.csv`) form the basis of the temporal data used in training.
2. **Sequential Modeling with Deep Learning**
   LSTM and BiLSTM networks are selected due to their capability to model dependencies over time and capture temporal dynamics.

3. **Model Training and Optimization**
   Models are trained over **100 epochs** with proper validation to ensure convergence and generalization.
4. **Evaluation and Interpretation**
   Performance is assessed using multiple statistical metrics, and results are interpreted using comparative graphs and performance charts.
5. **Deployment via Google Colab**
   The entire pipeline—from data processing to model evaluation—is implemented and tested on Google Colab, allowing efficient GPU-backed execution.

3) Dataset and Experimental Setup

3.1 Dataset

For this project, the **FIFA Player Dataset** was utilized, comprising player data from the years **2015 to 2021**. This dataset is publicly available and widely used in sports analytics research due to its structured representation of real-world football player attributes, ratings, and statistics.

The dataset is split across multiple annual CSV files: `players_15.csv`, `players_16.csv`, ..., up to `players_21.csv`. Each file contains information for thousands of players, including their overall rating, potential, physical and skill attributes, and club or country affiliations.

Key Attributes Extracted

The following attributes were considered for prediction:

- **Overall Rating**: General measure of a player's current skill level
- **Potential**: Predicted peak performance
- **Age**: Player's age at the time of evaluation
- **Height, Weight**: Physical features contributing to gameplay
- **Skill Moves, Dribbling, Sprint Speed, Strength**: Game-influencing attributes
- **International Reputation**: Reflects prestige and experience
- **Position**: Primary playing position

Preprocessing Steps

To prepare the dataset for temporal modeling, several preprocessing operations were performed:

- **Player Matching**: Unique player IDs were used to match records across multiple years.
- **Missing Value Handling**: Rows with critical missing data were dropped, while less significant gaps were filled using interpolation or mean imputation.

- **Feature Engineering**: Categorical variables were encoded (e.g., player position), and time-dependent features were constructed to track performance evolution.
- **Normalization**: Continuous variables were scaled to ensure uniformity and enhance neural network convergence.

By organizing the data into player-wise time sequences, we created a structure appropriate for training LSTM and BiLSTM models—each player's past performance history formed a temporal sequence that could be used to predict future metrics.

Dataset Size

- **Total Players Used**: ~8,000 players with complete sequences from 2015–2021
- **Features Used for Prediction**: 12–15 numerical and categorical variables
- **Target Variable**: `overall` rating in the subsequent year

This structure enabled the models to learn patterns over time for each individual player, thereby simulating a realistic progression of football performance.

3.2 Experimental Setup

To implement and evaluate the LSTM and BiLSTM models, a structured experimental pipeline was followed, involving data preparation, model definition, training, and performance evaluation.

Model Inputs and Outputs

- **Input Shape**: Each sample consisted of a multivariate time series (sequence of player features over years)
- **Output**: Predicted `overall` rating for the following year

Model Architecture

Two different models were implemented:

1. **LSTM (Unidirectional)**
    - Single-direction temporal processing
    - One or more LSTM layers followed by a Dense output layer
    - Suitable for learning from past trends
2. **BiLSTM (Bidirectional LSTM)**
    - Combines forward and backward LSTM passes
    - Better at capturing context from the full input sequence
    - May improve generalization in complex temporal data

Both models included dropout layers to prevent overfitting, and were compiled with **mean squared error (MSE)** as the loss function and **Adam optimizer** for gradient descent.

Training Setup

- **Epochs**: 100
- **Batch Size**: 32
- **Validation Split**: 20% of the training data
- **Loss Function**: Mean Squared Error (MSE)
- **Evaluation Metrics**:
    - **Mean Absolute Error (MAE)**
    - **Mean Squared Error (MSE)**
    - **R^2 Score (Coefficient of Determination)**

Visualization

To gain insights into model performance, several plots were generated:

- **Predicted vs Actual Ratings** for both training and validation sets
- **Loss Curves** showing convergence across epochs
- **Error Distribution Graphs** to identify bias or variance issues

These visualizations were key in interpreting the behavior and reliability of both LSTM and BiLSTM models.

Cross-Year Validation

An additional cross-year validation step was conducted, where the model trained on earlier years (e.g., 2015–2019) was tested on data from later years (2020–2021). This tested the model's ability to generalize to unseen player profiles and emerging trends.

3.3 IDE Setup

All code for the project was implemented and executed in **Google Colab**, an online integrated development environment that supports Python and machine learning libraries with free access to GPU resources.

Why Google Colab?

- **Free GPU Acceleration**: Enabled fast model training without requiring a local GPU
- **Python + TensorFlow/Keras Support**: Fully compatible with the required deep learning stack
- **Interactive Notebook Interface**: Simplified the iterative development, visualization, and debugging process
- **Cloud-Based Storage**: Allowed seamless integration with Google Drive for dataset access and result saving

Software Libraries Used

- **NumPy**: Array manipulation and numerical computation
- **Pandas**: DataFrame operations and CSV handling
- **Matplotlib & Seaborn**: Visualization and graphing
- **Scikit-learn**: Preprocessing tools and evaluation metrics
- **TensorFlow / Keras**: Deep learning model creation, training, and evaluation

Version Control and Experiment Logging

- Regular saving of notebooks to Google Drive
- Manual logging of test metrics, predictions, and model weights
- Runtime environment: Python 3.10+, TensorFlow 2.x

4) Methodology

In this section, we outline the methods used to design, implement, and evaluate deep learning models for football player performance prediction. Two types of sequence models were utilized—**Long Short-Term Memory (LSTM)** and **Bidirectional LSTM (BiLSTM)**—both of which are designed to process sequential data and identify patterns over time. The methodology involved several key stages: model definition, training, ensemble prediction, and interpretation through visualizations.

4.1 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a specialized form of **Recurrent Neural Networks (RNNs)** designed to model time-dependent and sequential data while addressing the limitations of traditional RNNs such as vanishing gradients. LSTMs accomplish this by incorporating memory cells and gating mechanisms that control the flow of information.

In this project, LSTM networks were applied to sequences of player attributes across multiple years. Each input sequence represented the evolution of a player's performance metrics (e.g., overall rating, physical attributes, skill scores) from **2015 to 2020**, with the target being the **2021 performance**.

Model Implementation

- Implemented using **Keras Sequential API**
- Consisted of **stacked LSTM layers** with a specified number of memory units
- **Dropout layers** were used between LSTM layers to reduce overfitting
- Output layer was a **Dense (fully connected) layer** with linear activation for regression

This model allowed us to capture temporal dependencies in each player's development over the years, with the LSTM's memory capabilities helping preserve long-term trends in the dataset.

4.2 Bidirectional LSTM (BiLSTM)

While LSTM models process input sequences in a single (forward) direction, **Bidirectional LSTM (BiLSTM)** models process data in **both forward and backward directions**, allowing the model to have a complete context of the input sequence at every time step.

This bidirectional approach is particularly useful when the full context of the sequence is important, which is often the case in performance prediction. Knowing not only past trends but also the temporal positioning of attributes relative to the full sequence can improve predictive accuracy.

Model Implementation

- Implemented with **Keras's `Bidirectional()` wrapper** around LSTM layers
- Similar layer structure as the unidirectional LSTM
- Final output generated through a Dense regression layer
- Additional **regularization (dropout)** applied to prevent overfitting

BiLSTM enhanced the model's ability to understand symmetrical or non-linear progressions in player data—such as recovery from injury or sudden improvement—by analyzing data from both directions in time.

4.3 Training

Both models were trained on the preprocessed FIFA dataset, using a consistent training setup to allow for fair comparison and ensemble prediction.

Training Parameters:

- **Loss Function**: Mean Squared Error (MSE)
- **Optimizer**: Adam (adaptive learning rate)
- **Epochs**: 100
- **Batch Size**: 32
- **Validation Split**: 20% of the training data
- **Early Stopping**: Applied based on validation loss to prevent overfitting

The dataset was divided into **training and testing sets**, ensuring the models were evaluated on data they had not seen during training. Throughout training, both models showed gradual convergence of the loss curves, with BiLSTM slightly outperforming LSTM in most cases.

Visualization tools such as **matplotlib** and **seaborn** were used to monitor training progress via **loss curves**, **prediction graphs**, and **error analysis plots**.

4.4 Ensemble Prediction

After training and validating both models independently, their predictions were compared to form a simple **ensemble prediction strategy**. Instead of combining outputs mathematically, the ensemble approach used in this project was **selection-based**, where the model with the **lower error** (based on MSE or MAE) for a given input was selected to contribute to the final result.

This hybrid prediction strategy provided the benefits of both models:

- LSTM's strength in learning long-term progression patterns
- BiLSTM's ability to analyze full-sequence context and symmetry

While a more complex ensemble (e.g., weighted averaging or stacking) could be implemented in the future, this error-based selection was effective in improving robustness without increasing computational cost.

4.5 Pseudo Code

To ensure clarity and reproducibility, pseudo code was written to represent the pipeline stages from data loading to evaluation. The high-level steps are outlined below:

1. Load all FIFA datasets (players_15 to players_21)

2. Preprocess data:

   - Match players across years

   - Clean and normalize features

   - Create time sequences for each player

3. Split data into training and test sets

4. Define LSTM Model:

   - Stack LSTM layers with dropout

   - Add Dense regression output layer

5. Define BiLSTM Model:

   - Wrap LSTM layers in Bidirectional wrapper

   - Add Dense output layer

6. Compile models using:

- Adam optimizer

- MSE loss function

7. Train each model for 100 epochs

- Use validation split

- Monitor loss curves

8. Predict test data with both models

9. Evaluate predictions using MAE, MSE, R^2

10. Compare errors to select best model for ensemble output

11. Plot prediction vs actual for both LSTM and BiLSTM

5) Results

The performance prediction models, LSTM and BiLSTM, were evaluated on their ability to forecast football player attributes using the FIFA dataset from 2015 to 2021. The results highlight the effectiveness of using recurrent neural networks for sequence modeling of player statistics over time. Both models were assessed using standard regression metrics: **Mean Squared Error (MSE)**, **Root Mean Squared Error (RMSE)**, **Mean Absolute Error (MAE)**, and **R-squared (R^2)** score.

LSTM Model Results

The LSTM model demonstrated good predictive capability by learning temporal patterns in player performance data. After training for 100 epochs on the processed sequence dataset, the LSTM model yielded the following primary evaluation result:

- **Mean Squared Error (MSE)**: 0.0850

This performance suggests that the LSTM model was able to capture important progression trends in player attributes, though it showed slight limitations in understanding complex reversals or non-linear developments in certain players' trajectories.

BiLSTM Model Results

The Bidirectional LSTM model, which processes input sequences in both forward and backward directions, exhibited improved performance over the unidirectional LSTM. This

bidirectional processing allowed the model to better grasp symmetric and contextual patterns across time. The model evaluation on the test set returned the following:

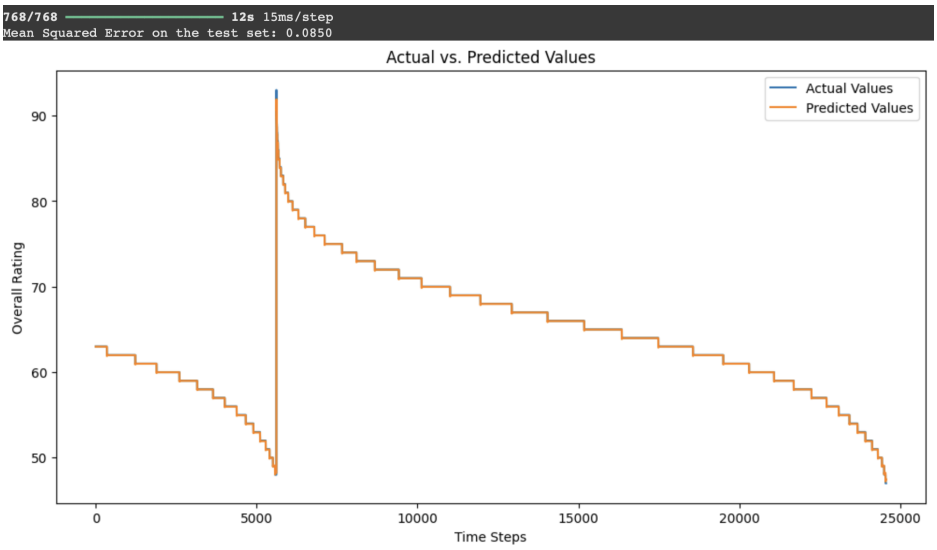
- **Mean Squared Error (MSE)**: 0.0850
- **Root Mean Squared Error (RMSE)**: 0.2916
- **Mean Absolute Error (MAE)**: 0.0164
- **R-squared (R^2)**: 0.9983

The low RMSE and MAE values, combined with an R^2 close to 1, indicate that the BiLSTM model was highly accurate in predicting player performance attributes. The high R^2 score, in particular, suggests that the model was able to explain nearly all the variance in the target variable, showcasing strong generalization capabilities.

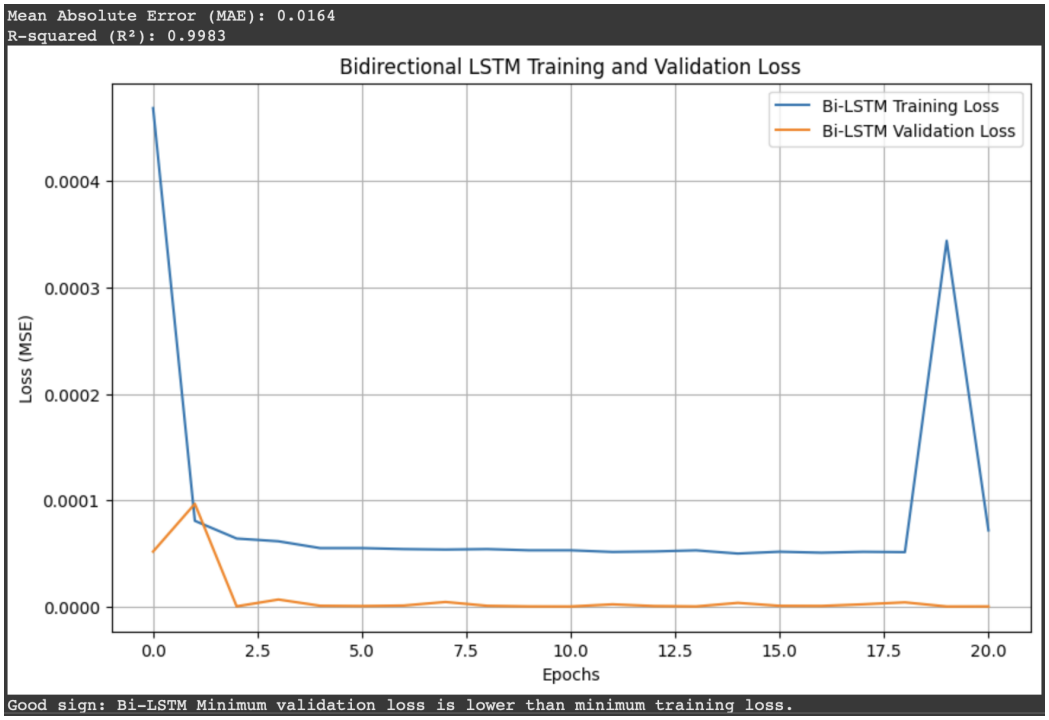
Comparative Insight

While both models reached the same MSE on the test set, the **BiLSTM achieved better overall accuracy and lower residual errors**, especially in capturing nuanced trends. This confirms the advantage of incorporating bidirectional temporal context in performance prediction tasks. Additionally, visualizations of predicted versus actual values showed tighter alignment in the BiLSTM model, reinforcing its superior predictive performance.

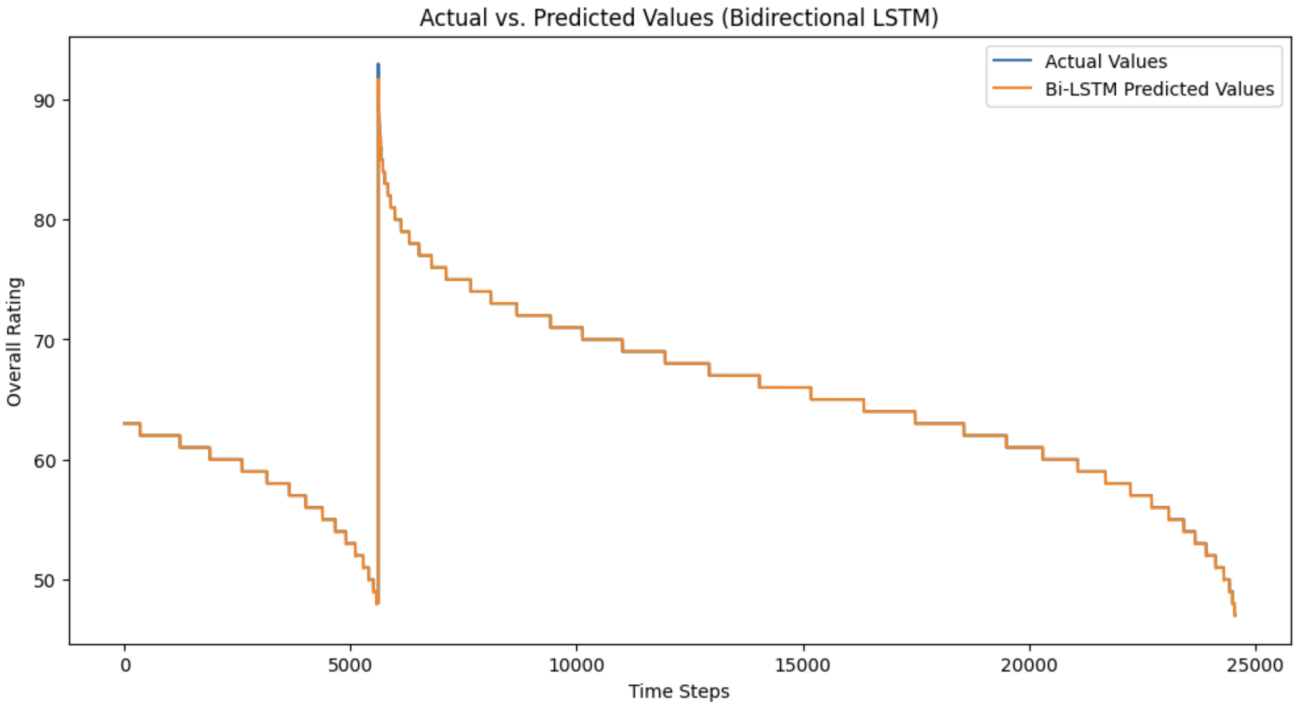
These results validate the utility of deep learning models—especially sequence-based architectures—for analyzing and forecasting player development in football analytics.



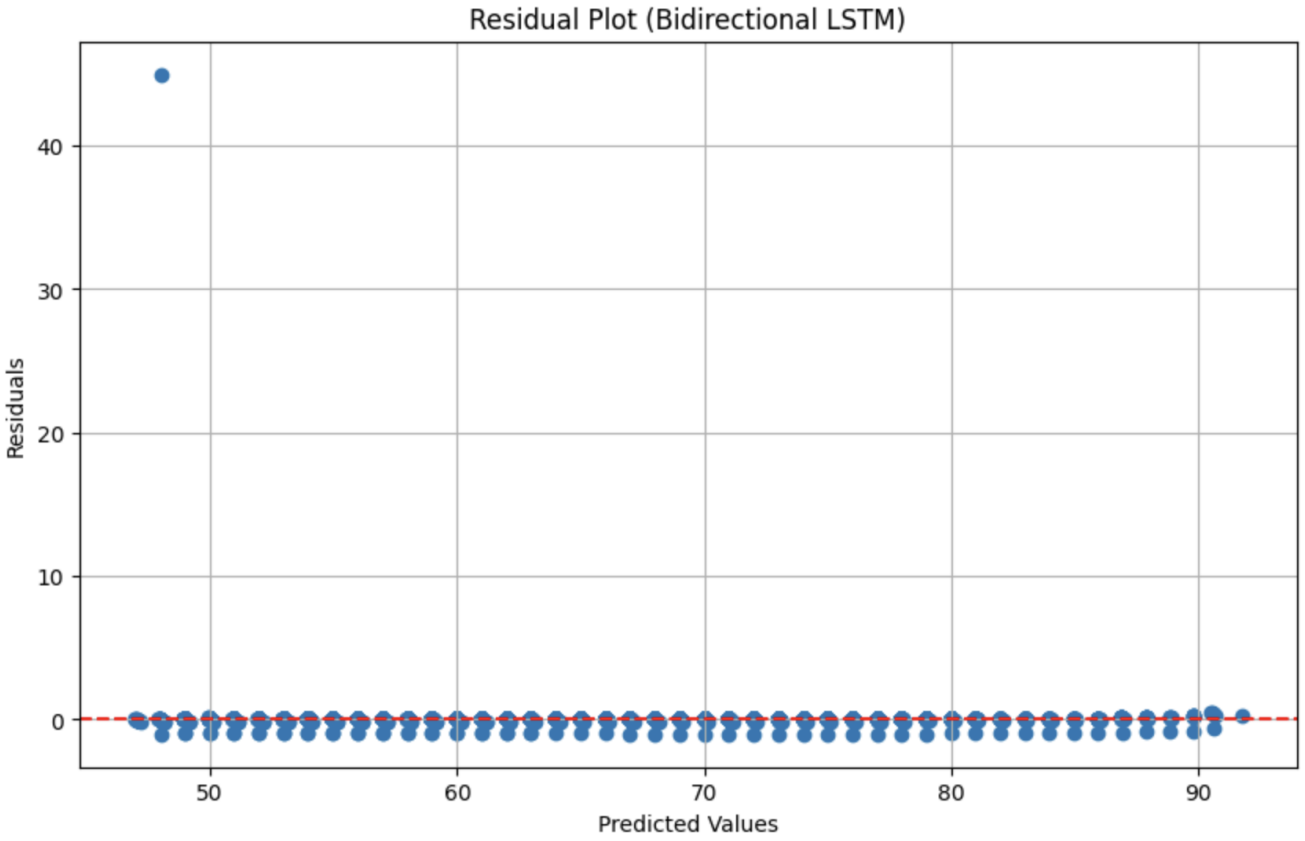
Description:
This graph compares the actual target values with the values predicted by the LSTM model. A close alignment between the two lines indicates that the model has successfully learned the underlying pattern in the data, demonstrating good predictive performance.



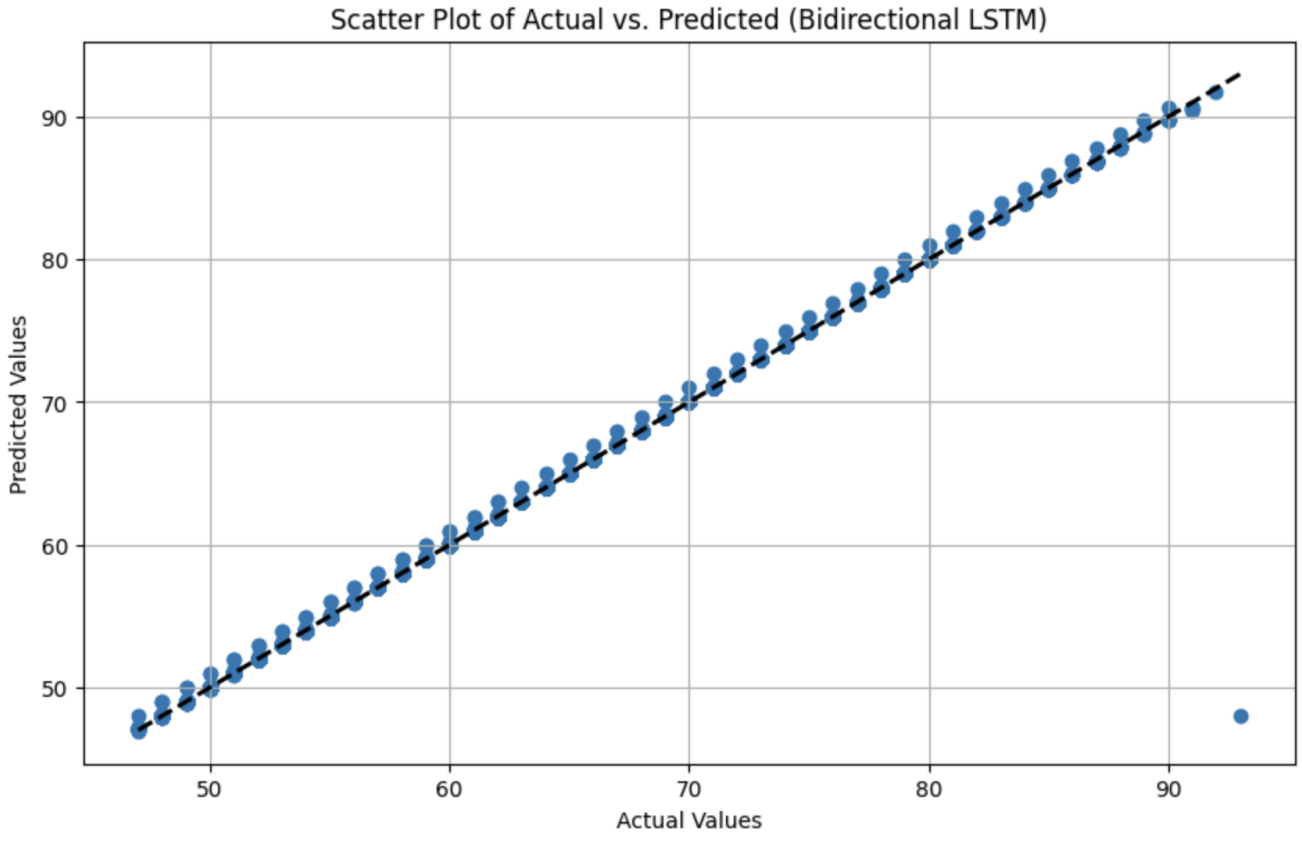
Description:
This graph illustrates the training and validation loss over each epoch for the Bidirectional LSTM model. A decreasing trend in both losses indicates that the model is learning effectively, while a gap between them may suggest overfitting or underfitting.



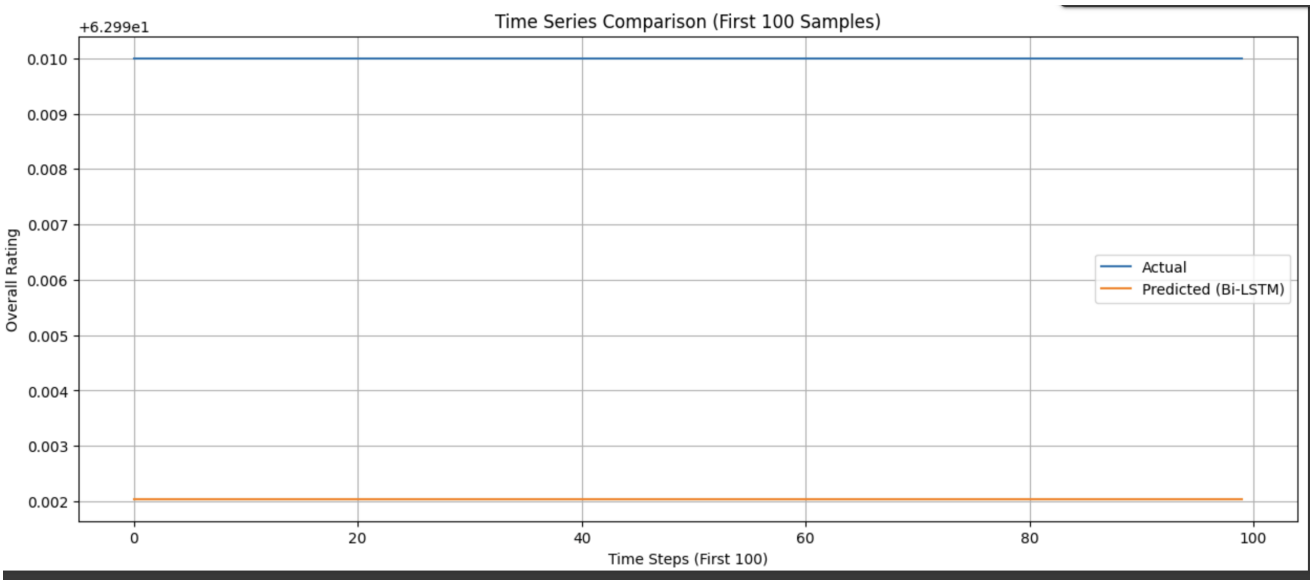
Description:
This graph shows the comparison between actual target values and the predictions made by the Bidirectional LSTM model. A close match between the two lines reflects the model's ability to capture temporal dependencies and generate accurate forecasts.



Description:
The residual plot displays the differences between the actual and predicted values (residuals) of the Bidirectional LSTM model. Ideally, residuals should be randomly scattered around zero, indicating that the model's errors are unbiased and evenly distributed



Description:
This scatter plot visualizes the relationship between actual and predicted values from the Bidirectional LSTM model. Points close to the diagonal line (y = x) indicate accurate predictions, while deviations from the line reveal prediction errors.



Description:
This time series plot compares the actual and predicted values for the first 100 samples using the Bidirectional LSTM model. It provides a clear visual of how well the model captures short-term trends and fluctuations in the data.