**Report: Stock Price Prediction Using LSTM**

**1. Objective**

The primary objective of this project is to predict future stock prices using historical stock data. It leverages Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN) capable of capturing sequential dependencies in time-series data.

**2. Approach**

This project follows a structured approach:

1. **Data Preprocessing**:
   * Raw stock data is cleaned, converted to numeric form, and normalized to ensure consistency.
2. **Model Building**:
   * An LSTM-based deep learning model is constructed to predict stock prices.
3. **Training and Evaluation**:
   * The model is trained on historical data and evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE), and R² metrics.
4. **Future Predictions**:
   * The trained model forecasts future stock prices using the last 60 days of historical data.
5. **Visualization**:
   * Results are visualized to compare actual and predicted prices, making trends and accuracy observable.

**3. Tools and Technologies Used**

* **Programming Language**: Python
* **Libraries**:
  + **Data Processing**: pandas, numpy
  + **Machine Learning**: scikit-learn (MinMaxScaler, metrics, model selection)
  + **Deep Learning**: TensorFlow, Keras (LSTM layers)
  + **Visualization**: matplotlib
* **Data Input**: Historical stock price CSV files.

**4. Features and Functionalities**

1. **File Selection**:
   * The user selects one of the predefined stock files for prediction.
   * Input is validated to ensure correctness.
2. **Data Preprocessing**:
   * Converts date strings to datetime objects for chronological ordering.
   * Converts numeric columns from string format with commas (e.g., "1,000") to floats.
3. **Model Training**:
   * Uses the past 60 days of stock prices to predict the next day's price.
   * Splits data into training (80%) and testing (20%) sets.
   * Normalizes data for faster and more stable training.
4. **Model Evaluation**:
   * Metrics like MSE, MAE, and R² evaluate the model's performance.
5. **Future Predictions**:
   * Predicts stock prices for a user-defined number of future days.
6. **Visualization**:
   * Plots actual vs. predicted prices for testing data.
   * Generates a future price forecast graph.

**5. Results**

1. **Training and Evaluation**:
   * The LSTM model was trained on the closing price of stocks.
   * Metrics:
     + **Mean Squared Error (MSE)**: Measures overall prediction error.
     + **Mean Absolute Error (MAE)**: Measures average prediction deviation.
     + **R² Score**: Indicates the percentage of variance explained by the model.
2. **Prediction Performance**:
   * On testing data, the model captures trends well, though exact price prediction is challenging due to market volatility.
3. **Future Forecast**:
   * Predicted future stock prices align with the observed trends but depend on assumptions about market behavior remaining consistent.

**6. Challenges and Limitations**

1. **Data Quality**:
   * The accuracy of predictions depends heavily on the quality and amount of historical data.
   * Missing or noisy data could degrade performance.
2. **Volatility**:
   * Stock markets are highly volatile, with events causing sudden price swings that are hard to predict.
3. **Overfitting**:
   * The model may perform well on historical data but struggle with real-world, unseen scenarios.
4. **Simplistic Feature Set**:
   * The model only considers closing prices. Including additional features (e.g., volume, moving averages, technical indicators) could improve performance.

**7. Improvements and Future Work**

1. **Feature Engineering**:
   * Incorporate features like trading volume, market sentiment, and technical indicators.
2. **Model Enhancements**:
   * Explore advanced architectures like GRU, Bidirectional LSTM, or Transformer-based models.
3. **Regularization**:
   * Add dropout layers to prevent overfitting.
4. **Additional Metrics**:
   * Include financial metrics like Mean Absolute Percentage Error (MAPE) for better interpretability.
5. **Hyperparameter Tuning**:
   * Optimize sequence length, batch size, learning rate, and number of LSTM units.
6. **Live Predictions**:
   * Integrate APIs to fetch live stock data and make real-time predictions.

**8. Conclusion**

This project demonstrates the use of LSTM for stock price prediction. While the model effectively captures historical patterns and trends, challenges like market volatility and limited features highlight areas for improvement. The inclusion of additional features, better data preprocessing, and advanced architectures can further enhance predictive accuracy.