**Project Proposal: Forecasting Next-Day S&P 500 Closing Price Using Deep Learning Models**

1. **Technical Objective & Data**

**Select a target variable from the provided daily time-series (e.g., S&P 500, 10-Year Treasury Yield). WHY (See 4 below)?**

The primary objective of this project is to forecast the next day closing price of the S&P 500 Index using three deep learning architectures: Feedforward Neural Network (FNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM).

The S&P 500 Index represents the performance of the 500 largest publicly traded companies in the United States and is considered one of the most accurate gauges of the overall U.S. equity market. It is highly liquid, broadly diversified, and closely watched by investors, analysts, policymakers, and academics. For our purposes, it offers several compelling advantages:

* Economic significance: As a leading market benchmark, the S&P 500 captures investor sentiment and macroeconomic trends, making it an ideal target for predictive modeling.
* Data richness: The S&P 500 has a long and uninterrupted historical time series, with readily available daily data (Open, High, Low, Close, Volume), enabling robust model training and validation.
* Practical relevance: Accurate short-term forecasts of the S&P 500's closing price can support portfolio allocation decisions, hedging strategies, and algorithmic trading, making our analysis highly applicable to real-world financial scenarios.

Given its central role in global markets, any insights or forecast improvements derived from modeling the S&P 500 can be generalized or extended to other market indices and investment instruments.

**What are your methodological and empirical analysis goals to forecast the next day value using FNN, CNN, and LSTM models?**

Our goal is to forecast the next-day closing price of the S&P 500 index using three different deep learning architectures: Feedforward Neural Network (FNN), Convolutional Neural Network (CNN), and Long Short-Term Memory Network (LSTM), each with unique strengths in time-series modeling.

**Feedforward Neural Network (FNN) – Baseline Approach**

Methodological goal: Construct a tabular supervised learning model using time-lagged input features.

Empirical goal: Use FNN to establish a benchmark for accuracy and model complexity. The performance of CNN and LSTM will be evaluated relative to this baseline.

**Convolutional Neural Network (CNN) – Pattern Recognition**

Methodological goal: Apply 1D convolutions across a window of sequential price data to capture localized temporal patterns such as mini-trends or reversals.

Empirical goal: Determine whether CNNs can outperform FNNs by recognizing short-term market signals more effectively, without requiring the full sequential memory of an LSTM.

**Long Short-Term Memory (LSTM) – Sequential Dependency Modeling**

Methodological goal: Leverage the LSTM’s architecture to model long-range dependencies and memory in financial time series.

Empirical goal: Evaluate if LSTM provides better forecasts around structural market shifts, such as economic crises or interest rate policy changes, where prior context is crucial.

**Cross-Model Goals**

Measure and compare out-of-sample predictive accuracy using RMSE, MAE, and MAPE.

Conduct hyperparameter tuning to assess how architecture-specific parameters affect performance.

**2. Plan for Model Implementations**

We will implement three models, each aligned with its structural strengths:

FNN (Feedforward Neural Network): Used as a baseline model.

We will transform the time series into a supervised learning format using time-lag features (e.g., previous 5 or 10 days of closing prices as input). The data will be tabular and static, without sequential awareness.

CNN (1D Convolutional Neural Network):

Will apply 1D convolutions over sequences of daily features (e.g., past 10 days) to capture local temporal patterns, such as short bursts of momentum or reversals. Expected to be effective at identifying patterns that repeat over short time windows.

LSTM (Long Short-Term Memory Network):

Designed to model long-term dependencies in sequential data, ideal for capturing trends or memory of structural events. We will feed in the past n-days’ worth of data to predict the next day’s closing price.

**3. Plan for Hyperparameter Tuning**

We will implement a systematic tuning process for each model using either Grid Search or Random Search, possibly supported by Keras Tuner or Optuna.

Hyperparameters to be optimized:

* Number of hidden layers and units
* Learning rate
* Batch size
* Dropout rate
* Sequence length (for CNN and LSTM)
* Number and size of convolution filters (for CNN)

Evaluation Metrics:

* Root Mean Square Error (RMSE)
* Mean Absolute Error (MAE)
* Mean Absolute Percentage Error (MAPE)

We will evaluate models on:

* Validation set to tune parameters.
* Test set to assess out-of-sample performance.

Deliverables:

* Loss curves (training vs. validation)
* Predicted vs. actual closing prices.
* Tables comparing model performance.
* Plots of model residuals and errors

**4. Economic/Business Context**

The S&P 500 closing price is not only a barometer of U.S. economic health but also a critical input for a variety of real-world applications:

Investment decisions: Traders and asset managers use forecasts to inform buy/sell signals.

Risk management: Banks and financial institutions rely on forecasts to hedge positions.

Policy analysis: Economists monitor index movements to evaluate the impact of fiscal or monetary policies.

Additionally, this time series contains structural breaks and volatility spikes tied to historical events (e.g., 2008 Financial Crisis, COVID-19 crash in 2020, Fed policy shifts). These events will likely affect model performance, particularly:

CNN may struggle with regime shifts due to localized nature.

LSTM might better adapt due to its ability to retain information across time.

We will identify and annotate such key events to study:

* How sensitive each model is to major market disruptions.
* Whether performance degrades near these points or if certain models adapt more smoothly.