PROJECT ABSTRACT

Stock Price Forecasting using LSTM Neural Networks (NIFTY-50 Dataset)

An LSTM neural network that forecasts the next trading day's NIFTY-50 closing level from a 15-day window of daily OHLCV data — trained with a leakage-free chronological pipeline, evaluated with walk-forward RMSE/MAE/MAPE against a persistence baseline, and demonstrated through a live web app where the model trains in the browser on real market data.

01. Title

Stock Price Forecasting using LSTM Neural Networks (NIFTY-50 Dataset)

02. Introduction / Background

Equity index forecasting is a classic time-series problem and a perennial final-year project choice, because it combines real data, deep learning and a genuinely hard evaluation question. Prices carry short-term memory — trend, weekly rhythm and volatility clustering — that feed-forward models discard, which is why recurrent architectures such as the Long Short-Term Memory (LSTM) network are the natural tool: its forget, input and output gates learn what to retain across a look-back window. The NIFTY-50, India's benchmark index of fifty large-cap stocks, provides a clean, publicly available daily series. This project builds a complete forecasting pipeline around it: real NSE data, a sequence-to-one LSTM regressor, and an evaluation protocol strict enough that the reported numbers mean something.

03. Problem Statement

Most student stock-prediction builds fail on methodology, not modelling. Typical defects: shuffling the series before the train/test split (leaking future prices into training), fitting scalers on the full dataset, reporting a single error number with no baseline, and implying the model is tradeable. This project addresses each defect directly: it (1) fetches real NIFTY-50 daily OHLCV from NSE archives, (2) builds 15-day sliding-window samples with strictly chronological splits and train-only scaler fitting, (3) trains a documented LSTM regressor, (4) evaluates with walk-forward retraining using RMSE, MAE and MAPE, and (5) scores a persistence baseline ("tomorrow equals today") on the identical folds — so the model's true edge, or lack of one, is explicit rather than hidden.

04. Objectives

- Fetch and clean real NIFTY-50 daily OHLCV data (reference build: 249 trading days, 27 Sep 2024 – 26 Sep 2025) into a documented CSV.
- Prepare leakage-free sequences: 15-day windows of open/high/low/close/volume targeting the next day's close, with chronological 80/20 splits and train-only min-max scaling.

- Design and train a sequence-to-one LSTM (32 units, dropout 0.2, linear Dense output, ~5,000 parameters) with Adam on MSE and early stopping.
- Evaluate with walk-forward expanding-window retraining, reporting RMSE, MAE and MAPE per fold against the persistence baseline.
- Build a live web demo: price explorer with moving averages, LSTM architecture views, and a pure-JavaScript LSTM that trains in the browser on the real series.
- Document metrics, limitations and the efficient-market reality honestly, with a design target of MAPE around 1% of the index level — never an invented accuracy.
- Deliver the complete student kit: scripts, notebook, trained model, demo, report, PPT and viva Q&A.;

05. Proposed Methodology

The pipeline is implemented in Python (TensorFlow/Keras, pandas, scikit-learn) as a Jupyter notebook plus data scripts. (1) Data: yfinance pulls ^NSEI daily OHLCV from NSE archives; a cleaning step checks for missing sessions and corporate-action adjustments. (2) Sequencing: each sample is a 15×5 window with the next day's close as target; the series is split chronologically 80/20 and scalers are fit on the training portion only. (3) Model: LSTM(32, tanh/sigmoid gates) → Dropout(0.2) → Dense(1, linear), compiled with Adam (learning rate 0.005) and MSE loss, early stopping on a validation slice. (4) Evaluation: walk-forward expanding-window retraining scores RMSE, MAE and MAPE per held-out fold; the persistence baseline runs on identical folds. (5) Demo: a single-file HTML app embeds the series and implements the same LSTM in JavaScript (forward pass, backpropagation through time, Adam, dropout) for live in-browser training.

06. System Architecture / Working

At runtime the forecaster takes the most recent 15 trading days of OHLCV, applies the fitted min-max scalers, and feeds the 15×5 sequence through the LSTM layer; the final hidden state passes through dropout (training only) and the Dense layer to produce the next-day close in index points, which is inverse-scaled for display. The training loop shuffles only within the training window — never across the chronological boundary — and checkpoints on validation MSE. The web demo mirrors this: the Price Explorer view charts the embedded real series with 20/50-day moving averages and OHLC hover; the Architecture view documents the pipeline and cell internals; the Train & Forecast view trains the JavaScript LSTM live (fixed seed for reproducibility), plots train/validation loss, and overlays one-step-ahead forecasts against actual closes with the baseline.

07. Hardware / Software Requirements

- Python 3.10+ with TensorFlow/Keras, pandas, NumPy, scikit-learn, Matplotlib
- yfinance for NSE archive access (or a manually downloaded NSE CSV)
- Jupyter Notebook for the training and evaluation pipeline
- Laptop CPU (training completes in approximately 2–5 minutes — expected)
- Modern web browser for the demo (Chrome/Edge/Firefox; trains in-browser, works offline)

- Git for source versioning

08. Expected Outcomes

- A cleaned, documented NIFTY-50 daily OHLCV dataset with a reproducible fetch script.
- A trained LSTM forecaster (.keras) with fitted scalers and a leakage-free training notebook.
- Walk-forward evaluation tables: RMSE, MAE and MAPE per fold, side-by-side with the persistence baseline.
- A live single-file web demo: price explorer, architecture diagrams, and in-browser LSTM training with loss curves and forecast-vs-actual overlay.
- A report documenting the achieved metrics from the build run (design target: MAPE around 1% of index level — stated as a target, not a measured claim).
- PPT presentation and viva Q&A; covering LSTM gates, leakage pitfalls, baseline reasoning and metric interpretation.

09. Applications

- Teaching time-series methodology: chronological splitting, walk-forward validation and baseline discipline.
- A portfolio-grade deep-learning project demonstrating LSTM design on real financial data.
- A reusable forecasting template — the same pipeline ports to other indices or commodities by changing the data source.
- Classroom demonstration of why naive accuracy claims in finance collapse under proper evaluation.

10. Future Scope

- Multi-horizon forecasting (3-day / 5-day ahead) with sequence-to-sequence decoders.
- Exogenous features: India VIX, USD/INR, sector indices and event calendars.
- Attention-based models (Temporal Fusion Transformer) compared on the same walk-forward protocol.
- A proper backtesting layer with transaction costs — explicitly separated from the forecasting study.
- Intraday extensions using NSE 15-minute candle data.