Project Summary

The objective is to build a hybrid forecasting model for weekly TSR20 rubber futures data. The model must use an LSTM to generate initial forecasts and an SVR model to correct residual errors. Both models should be optimized using Particle Swarm Optimization (PSO), with Mean Squared Error (MSE) as the optimization objective.

Dataset

The dataset covers weekly TSR20 rubber futures from 2015 to 2025. The target variable is "Price". The file name is:

Weekly Rubber TSR20 Futures Historical Data_2015-2025.csv

Model Requirements

Model structure:

- Use LSTM for primary forecasting.
- Use SVR to model residual errors from the LSTM output.
- Optimize both LSTM and SVR hyperparameters using PSO.
- The final hybrid model (LSTM + SVR) must outperform a baseline LSTM model trained with default parameters (without PSO).

Forecast target:

- Forecast the price 4 weeks ahead (t+4).
- Use a rolling sequence window (lookback) of 12 weeks.

Data split:

- Time-based split:
  - 70% training
  - 15% validation
  - 15% testing
- No random shuffling.

Data Preprocessing

- Convert the Date column to datetime and sort in ascending order.
- Convert 'Vol.' from 'K' string format to numeric values.
- Convert 'Change %' to decimal values.
- Rename columns as: ['Date', 'Price', 'Open', 'High', 'Low', 'Volume', 'Change']

Outlier handling:

- Apply the IQR method.
- Replace outliers with the column median.

Feature Engineering

- Use the ta library where applicable. The goal is to help the model learn important **price patterns**, **market volatility**, **seasonality**, and **short-term memory** effects. Not all features must be used—exclude the ones that degrade model performance.

1. Technical Indicators

Used primarily in the LSTM model to provide trend-following signals and momentum context.

- **Simple Moving Average (MA_5, MA_10, MA_20):**
  Shows short-term to mid-term price trends. Helps model detect direction and smoothing patterns.
- **Exponential Moving Average (EMA_5, EMA_10):**
  Similar to MA but gives more weight to recent prices. Can help the model respond faster to new trends.
- **Bollinger Bands (Upper, Lower, Width):**
  Capture price volatility and whether price is overbought/oversold. Useful to provide context on abnormal price moves.
- **RSI (Relative Strength Index, window=14):**
  Measures momentum. Helps model understand overbought (>70) or oversold (<30) conditions.

2. Volatility Indicators

Used to inform both LSTM and SVR models about current uncertainty or risk in the market.

- **Rolling Std Dev (5 and 10 periods):**
  Measures price variability. High values may signal unstable market conditions, useful for LSTM.
- **High − Low Spread:**
  Daily range of price movement. A good proxy for intraperiod volatility, used in LSTM.
- **Price − Open Difference:**
  Captures bullish or bearish pressure within a week. Useful to help LSTM model intraperiod directional momentum.

3. Temporal Features

These are date-based fields extracted from the timestamp and used by the LSTM to capture **seasonality or periodic behavior**.

- **Month:**
  Monthly seasonality effects. For example, Q1 demand spikes or Q4 dips.
- **Quarter:**
  Broader seasonal groupings (useful in commodities where quarter-level effects matter).

- **Week of Year:**
  Captures granular cyclical patterns (e.g., harvest cycles, supply contracts, etc.).
- All temporal features are treated as additional input features for the LSTM model.

4. Lagged Variables

Used **only for SVR** to help it model residuals based on recent error behavior.

- **Price_t-1 to Price_t-4:**
  These are lagged values of the target (Price), used as part of the input to SVR. They help model autocorrelation in the residuals—especially where LSTM underfits short-term changes.
- These lag features are excluded from LSTM to avoid leaking future information, but are useful for SVR to learn error correction patterns.

5. Residual Lags (Dynamic at Inference)

- Lagged residuals (e.g., residual_t−1 to residual_t−4) are constructed and passed to SVR during training and inference. These are not part of the dataset but generated after LSTM prediction.
- Residual lags are essential to prevent data leakage and to enable the SVR to model persistent under- or over-prediction patterns in LSTM.

After feature engineering, drop all rows containing NaN values.

Model Optimization

LSTM (via PSO):

- Hyperparameters to optimize:
  - units_1: 32 to 128
  - units_2: 32 to 64
  - dropout: 0.1 to 0.5
  - learning_rate: 0.0001 to 0.01
- Use RMSprop optimizer
- Loss function: MSE
- Include EarlyStopping and ReduceLROnPlateau during training for computational efficiency

SVR (via PSO):

- Kernel: RBF
- Hyperparameters to optimize:
  - C: 0.1 to 100
  - gamma: 0.001 to 1.0
  - epsilon: 0.001 to 0.5
- Target values: residuals from LSTM predictions

Evaluation and Visualization

Plots required:

1. Line plots comparing:
    - Actual vs LSTM predictions
    - Actual vs Combined (LSTM + SVR) predictions
2. Residual scatter plots (Train/Val/Test) for:
    - LSTM
    - LSTM + SVR
3. Histogram of test residuals for both LSTM and combined model
4. Training history plots (Loss and MAE per epoch)
5. Comparison bar chart of test set performance metrics for LSTM vs Combined

Metrics to report:

- MSE
- RMSE
- MAE
- R^2
- MAPE
  (MAPE should ignore zero targets to avoid division errors.)

Deliverables

- Python code (compatible with Google Colab and GPU runtime)
- Jupyter Notebook (.ipynb) and/or .py script
- Model training output
- PNG image files for all plots
- Final test set evaluation metrics for both LSTM-only and LSTM+SVR models
- A short summary of findings and performance comparison

The final model must demonstrate measurable performance improvement over the baseline LSTM across multiple evaluation metrics.