FINAL PROJECT REPORT

(Stock Price Prediction)

Dasari Sai Harsh
2022144

Aryan Singla
2022112

PCA STACKED LSTM:-

Long Short-Term Memory (LSTM) networks have emerged as a powerful tool for predicting stock prices due to their ability to effectively capture and learn from sequential data patterns over time. Stock price data is inherently sequential, where each data point depends on previous observations. Traditional feedforward neural networks often struggle to handle such temporal dependencies effectively. However, LSTM networks address this challenge by incorporating specialized memory cells and gating mechanisms that allow them to maintain a memory of past information and capture dependencies over longer time horizons.

One of the key strengths of LSTM networks lies in their ability to learn long-term dependencies in sequential data. They achieve this by employing gates, such as input, forget, and output gates, which regulate the flow of information through the network. This enables LSTM networks to capture complex patterns and non-linear relationships present in stock price data, which may be challenging to model using traditional statistical methods.

Predicting stock prices involves considering a myriad of financial and technical indicators, each offering unique insights into market behavior. While there isn't a definitive best predictor due to the dynamic nature of markets, combining various indicators enhances predictive accuracy. Financial indicators such as earnings reports and macroeconomic trends provide fundamental insights, while technical indicators derived from historical price and volume data offer insights into market sentiment and momentum.

To streamline the predictive modeling process and improve model performance, techniques like Principal Component Analysis (PCA) are employed. PCA helps identify the most relevant features by reducing the dimensionality of the data and capturing the most significant variation among the indicators. By focusing on the most informative features, PCA mitigates the risk of overfitting and noise in predictive models.

A quick search revealed about 77 of the most commonly used technical indicators(pic attached for reference) all of which were computed for each of the 3 stocks, after which data was “cleaned”.This involved handling large values,dropping rows with null values of the time series values, removing rows and columns whose number of null values exceeded the threshold. The rows and columns with less than the threshold number of null values were filled using knn(imputation) and data was normalized. PCA was then applied on these indicators and 6 principal components were chosen such that cumulative variance sum was about 90 percent of the total variance of the resultant matrix.

category	Index and their identifier
Trading time series indicators	Opening price, high price, low price, close price, pre_close price, change, volume, amount
Trend index	BBI, DMA, DMI, EXPMA, MA, MACD, MTM, RICEOSC, SAR, TRIX,
Counter trend index	B3612, BISA, CCI, DPO, KDJ, SLOWKD, ROC, RSI, RPS, SI, SRDM, VROC, VRSI, WR
Energy index	ARBR, CR, PSY, VR, WAD
Volume price index	MFI, OBV, PVT, SOBV, WVAD
Pressure support index	BBIBOLL, BOLL, CDP, ENV, MIKE
Trading volume	Quantityrelativeratio, VMA, VMACD, VOSC, TAPI, VSTD
Overbought and oversold indicators	ADTM
Swing index	MI, RC, SRMI
Relative Strength Index	DPTB, JDQS, JDRS, ZDZB
Volatility index	ATR, MASS, STD, VHF, CVLT

After PCA, you have a transformed dataset (x_pca) with reduced dimensions, where each component is a linear combination of your original features. This transformed dataset is then used as input for the LSTM model.

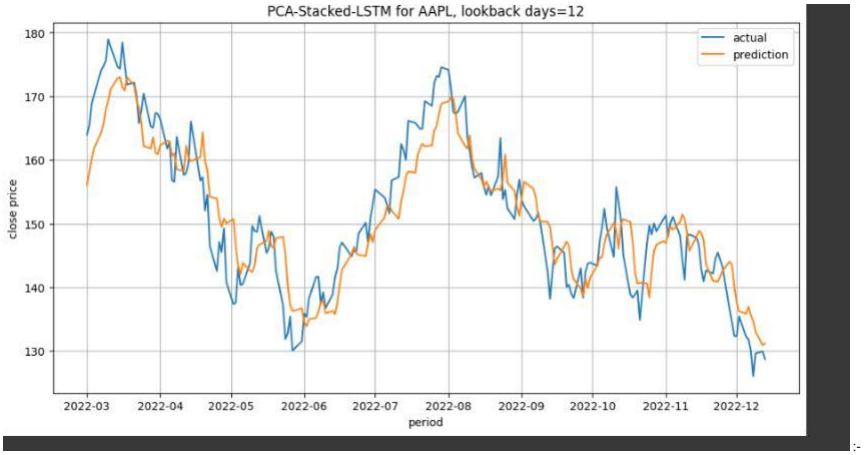
1. **Data Preparation:** The transformed data (x_pca) is prepared for the LSTM model. This involves creating sequences of a specified number of days (num_days). Each sequence consists of num_days consecutive days of data, and the target (y_seq) is the 'Next_day' value of the day following each sequence. The sequences (X_seq) and targets (y_seq) are then converted to numpy arrays.
2. **Data Scaling:** The input sequences and targets are scaled using MinMaxScaler to ensure that the LSTM model, which is sensitive to the scale of input data, can train more effectively.
3. **Train-Test Split:** The data is split into training and testing sets using TimeSeriesSplit. This is a special type of cross-validation technique for time-series data where the train set is always before the test set in time to prevent future leakage.
4. **Model Building:** An LSTM model is built using the Keras library. The model consists of two LSTM layers followed by two Dense layers. The first LSTM layer returns sequences to ensure that the second LSTM layer receives sequences as input. The final Dense layer has one unit, which corresponds to the prediction of the 'Next_day' value.
5. **Model Training:** The LSTM model is trained on the training data using the Adam optimizer and the Mean Squared Error loss function. The EarlyStopping callback is used to stop training when the validation loss stops improving, which helps prevent overfitting.
6. **Prediction:** The trained LSTM model is used to predict the 'Next_day' values of the test data.

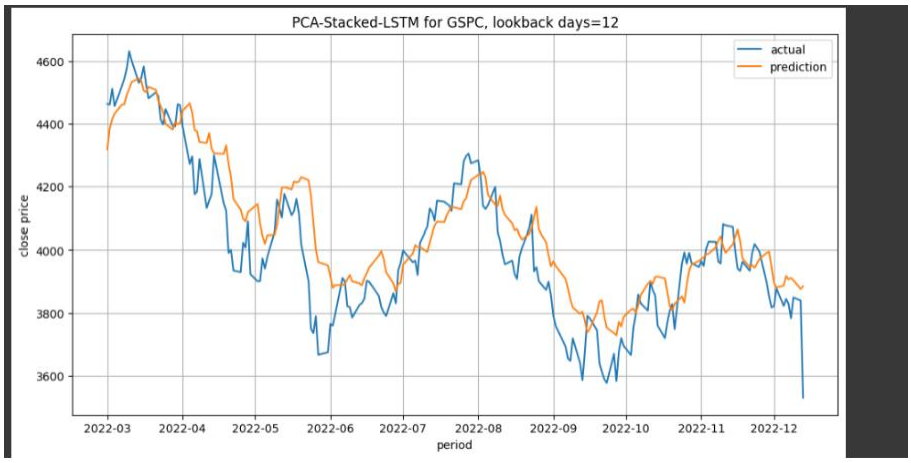
7. **Evaluation:** The model's predictions are compared with the actual 'Next_day' values of the test data. This comparison is done both with the standardized values (used for training the model) and with the original values (obtained by inverse transforming the standardized values). The comparisons are visualized using line plots.

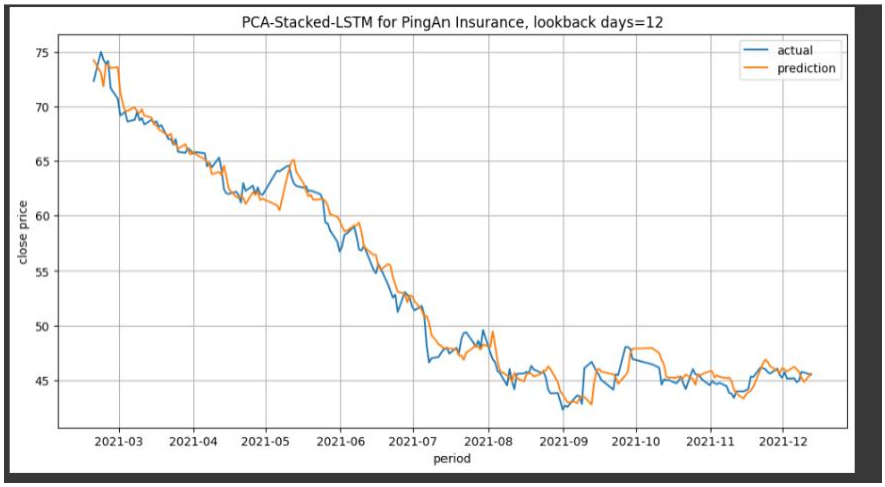
All of the hyperparameters were obtained by testing with various values and finding which values works best on a range of stocks with different trends,and also taking note from various research papers which have been cited below.

The LSTM model uses Principal Component Analysis (PCA) transformed data as input. It consists of two LSTM layers with 128 and 64 units respectively, followed by two Dense layers. The LSTM layers are designed to capture long-term dependencies in the sequence data, making them suitable for time-series forecasting. The model is trained using the Adam optimizer and Mean Squared Error loss function. To prevent overfitting, early stopping is used. This technique monitors the validation loss and stops training if it doesn't improve for a specified number of epochs (in this case, 10). This ensures the model generalizes well and doesn't overfit the training data. After training, the model is used to predict future stock prices.

The following are the results for various stocks







ANALYSIS OF RESULTS:-

Image 1 (AAPL):

The model's predictions generally follow the overall trend of the actual data, capturing major fluctuations and peaks. However, there are some deviations, where the predictions either over or underestimate the actual values at certain points in time.

Image 2 (GSPC):

Similar to the AAPL dataset, the model's predictions capture the overall trend and major fluctuations in the GSPC index. However, there are instances where the predictions deviate from the actual values, particularly during periods of rapid changes or significant peaks and troughs.

Image 3 (PingAn Insurance):

In this case, the model's predictions exhibit a smoother pattern compared to the actual data, which shows more volatility and sharper fluctuations. The predictions tend to underestimate the actual values during periods of rapid increases and overestimate during periods of rapid decreases.

Overall, these results demonstrate the capabilities and limitations of the PCA-Stacked-LSTM model in forecasting time series data. While the model can capture general trends and patterns, there are instances where it struggles to accurately predict extreme fluctuations or rapid changes in the data.

RELEVANT LITERATURE:-

1.) Study on indicators
2.) PCA_LSTM
3.) PCA_LSTM2
4.) technical indicators and their formulas borrowed from here

ENSEMBLE LEARNING USING SVR AND RANDOM FORESTS:-

As mentioned in the first deadline, we wanted to include the random forest in the first deadline as a part of a ensemble learning method to improve a models accuracy.It's interesting to note that while the research paper highlighted the use of Extra Trees as a base regressor, in our implementation, the Random Forest model seemed to yield better results. This could be attributed to the inherent differences between the two models. While both are ensemble methods that use decision trees, the way they split nodes differs. Random Forest chooses the optimum split among a random subset of features, while Extra Trees chooses it randomly. This additional randomization in Extra Trees can help reduce model variance, but it might not always lead to better performance.

1. **Fetching Stock Data**: The stock data for Apple Inc. (ticker symbol 'AAPL') is fetched from Yahoo Finance for the period from August 1, 2018, to January 31, 2019.
2. **Calculating Technical Indicators:** Various technical indicators are calculated from the stock data. These include Simple Moving Average (SMA), Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD), Commodity Channel Index (CCI), Stochastic Oscillator (%K and %D), Rate of Change (ROC), Relative Strength Index (RSI), Price Volume Trend (PVT), and others.
3. **Preprocessing the Data:** The target variable 'Close' is separated from the features. The features are then standardized using StandardScaler from sklearn.
4. **Defining the Base and Meta Regressors:** The base regressors (SVR and Extra Trees Regressor) and the meta regressor (SVR) are defined. These are used to create a StackingCVRegressor, which is a stacked ensemble model.
5. **Time Series Cross-Validation:** Time series cross-validation is performed for training periods of 3, 4, and 5 months. For each training period, the model is trained on the training data and predictions are made on the test data. The R2 score and Root Mean Squared Error (RMSE) are calculated for each split. This process is repeated for both the stacked ensemble model and the standard SVR.
6. **Calculating Percentage Difference**: The percentage difference in R2 and RMSE between the stacked ensemble model and the standard SVR is calculated.
7. **Plotting the Results**: A bar plot is created to visualize the percentage difference in RMSE between the stacked ensemble model and the standard SVR for each training period.

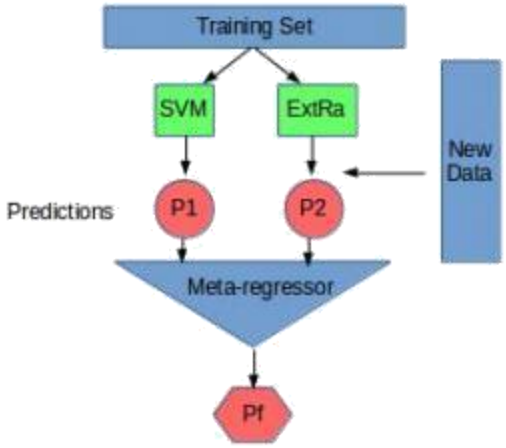


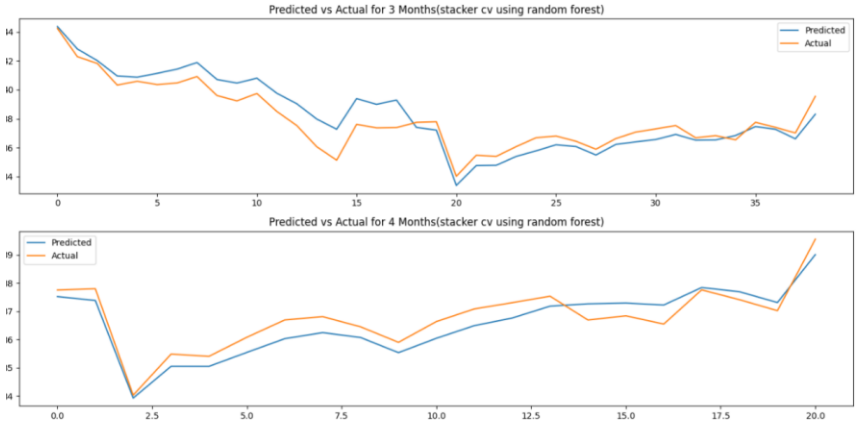
Fig. 1. Stacked Regressor architecture

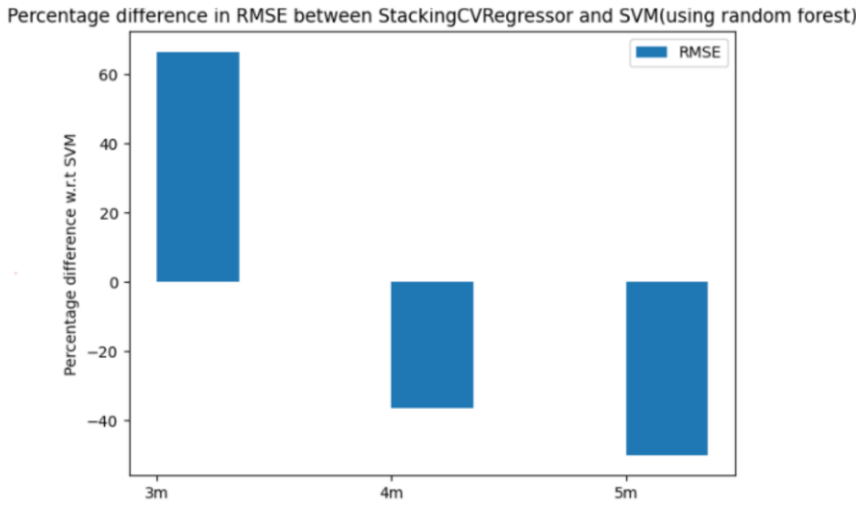
SVR and Extra Trees are combined in a stacked ensemble model for their complementary strengths. SVR, a linear regression model, and Extra Trees, a tree-based model, capture diverse patterns in data. Their different error types can compensate for each other's mistakes, enhancing the model's accuracy. Both are robust to overfitting, SVR through regularization and Extra Trees through randomization and averaging. They handle various data types and complexities, making them ideal base models for a meta-regressor in stock price prediction tasks. However, model choice should always be data and problem-specific, and validated through rigorous testing.Support Vector Regression (SVR) and Extra Trees are fundamentally different in their learning mechanisms, which allows them to compensate for each other's mistakes when used together in an ensemble model.SVR is a type of regression analysis that uses the concept of a hyperplane in a high-dimensional space to predict continuous values. It tries to find the best line (in 2D) or hyperplane (in higher dimensions) that fits the data points. If SVR makes a mistake, it's likely because the data doesn't fit well to a line or hyperplane, or the chosen kernel function doesn't capture the underlying pattern of the data.Extra Trees is a tree-based model that uses a multitude of decision trees to make predictions. Each decision tree in the ensemble is built from the original sample and then adjusted with random splits of the nodes. If Extra Trees makes a mistake, it's likely because the data doesn't fit well to the decision boundaries formed by the trees, or the random splits didn't capture the underlying pattern of the data.

The time series used for testing look similar to this:-

Fig. 2. Visualising the Time Series validation process. n_splits = 5

6 months worth of data is collected.In the first iteration, data is trained on first 3 months and predicts the next 3. In the second, training is done on 4 months worth of data and tested on last 2 months. SImilarly for the 5 months period. Then the results of the stacked regressor are compared with the standard SVM trained on the same parameters. The following are the results:-





Based on the images provided, the following observations can be made:
Image 1 and 3 show the predicted and actual values for 3 and 4 months, using the stacker CV model with extra trees and random forest methods, respectively. The plots indicate that the predicted values generally follow the trend of the actual values, but there are some deviations, especially in certain periods.
Image 2 and 4 display the percentage difference in RMSE (Root Mean Squared Error) between the StackingCVRegressor and SVM (Support Vector Machine) models, using extra trees and random forest methods, respectively. The bars represent the RMSE values for different time periods (3m, 4m, and 5m).
For the extra trees method (Image 2):

The StackingCVRegressor shows a significantly higher RMSE (around 100% difference) compared to SVM for the 3-month period.
The difference in RMSE is lower (around 75%) for the 4-month period.
For the 5-month period, the StackingCVRegressor has a slightly lower RMSE than SVM (around -10% difference).

For the random forest method (Image 4):

The StackingCVRegressor has a higher RMSE (around 20% difference) than SVM for the 3-month period.
For the 4-month period, the RMSE difference is negligible (around 0%).
The StackingCVRegressor shows a significantly lower RMSE (around -40% difference) compared to SVM for the 5-month period.

These results indicate that the performance of the StackingCVRegressor relative to SVM varies depending on the time period and the method (extra trees or random forest) used. The extra trees method seems to favor SVM for shorter periods, while the random forest method favors StackingCVRegressor for longer periods.

RELEVANT LITERATURE:-

3.) Overview on ensemble learning for stock prediction
4.) Stacked regressor
5.) Comparison between stacked and normal regressors