Comparative Analysis of SVM and K-SVR in Stock Market Prediction

A project work submitted for the course of

Machine Learning
(CSL 7620)

Indian Institute of Technology, Jodhpur
Department of Computer Science and Engineering

Submitted by:
Akash Banik (M25MAC001)
Ali Asad Quasim (M25MAC002)
Anindita Pramanik (M25CSE006)

Abstract: The project develops accurate stock price prediction models using Support Vector Machines (SVM) with historical Yahoo Finance data. It compares SVR and kernel-based SVR using MAE, RMSE, and R^2 metrics, applying feature engineering, preprocessing, and hyperparameter tuning to optimize prediction accuracy for informed investment decisions.

1-1.1Introduction

Stock market prediction aims to forecast future price trends using analytical and computational methods. Due to market volatility, machine learning models like SVR, LSTM, and neural networks are widely used. Kernel-based SVR captures nonlinear relationships by mapping data into higher-dimensional spaces. Incorporating technical indicators and sentiment analysis enhances accuracy. Although perfect prediction is unattainable, these models offer valuable insights into market dynamics, assisting investors in making data-driven and informed financial decisions.

1.2 Important Factors for Prediction

Stock prediction depends on **fundamental**, **technical**, and **economic** indicators:

- **Fundamental:** Company performance, financial ratios (P/E, ROE, P/B), and balance sheet stability.
- **Technical:** Momentum indicators (SMA, RSI), trend patterns, volatility, and trading volume.
- **Economic:** GDP growth, inflation, and employment levels affecting overall market behavior.
- **Sentiment:** News, social media, and public perception influencing market reactions.

Combining multiple factors allows ML models to extract meaningful patterns for more reliable predictions despite the market's inherent uncertainty.

1.3 Challenges in Stock Market Prediction

Major challenges include:

1. **Market Nonlinearity:** Highly dynamic, chaotic price movements.
2. **Data Quality:** Noisy or limited historical data affecting learning.
3. **Overfitting/Underfitting:** Balancing model complexity with generalization.
4. **Market Efficiency:** Prices often reflect all known information (EMH).
5. **External Events:** Geopolitical or economic shocks causing unpredictability.
6. **Sentiment & Psychology:** Difficult-to-model emotional and behavioral influences.
   These factors make stock prediction complex, emphasizing the need for robust, adaptive ML models.

2 – Technology Used

2.1 Python

2.2 Google Colab

2.3 Models Used

- **Support Vector Machine (SVM) Regressor**
- **Kernel-based Support Vector Regressor (k-SVR)**

3 – The Data: Preprocessing and Feature Engineering

3.1 Data: Training and Testing -The dataset, sourced from Yahoo Finance, includes daily Open, High, Low, Close, and Volume data for **Apple, Microsoft, Google, Reliance, Tata Steel,** and **Meta** from **2021–2025**. Using an **80–20 train-test split**, the data exhibits upward trends with short-term volatility and event-driven spikes, making it ideal for evaluating linear and nonlinear regression models.

3.2 Feature Engineering and Data Analysis

To enhance model learning and prediction accuracy, several preprocessing and feature engineering techniques were used:

- **Simple Moving Average (SMA):** Used to smooth price fluctuations and identify trends. Shorter SMAs (e.g., 10 or 50 days) capture short-term trends, while longer ones (e.g., 200 days) indicate overall direction. SMA crossovers were used as signals for trend reversals.
- **Relative Strength Index (RSI):** A momentum indicator (0–100 scale) measuring the speed and change of price movements to identify **overbought** or **oversold** conditions.

Min-max scaling and time series split were applied for proper model training. Hyperparameter tuning used GridSearch with k-fold cross-validation (k=3 or 5), and model performance was evaluated using the negative mean squared error scoring technique for accurate results.

4 – Model Evaluation and Results

4.1 Evaluation Metrics-Three key metrics were used to assess model performance:

- **R² Score:** Measures similarity in predicted and actual data . Higher R² indicates better model fit.

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{n} \{y_i - \hat{y}_i\}^2}{\sum_{i=1}^{n} \{y_i - \bar{y}_i\}^2}$$

- **Mean Absolute Error (MAE):** Average absolute difference between predicted and actual prices.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |x_i - \hat{x}_i|$$

- **Mean Squared Error (MSE):** Average squared difference between predictions and actual values, penalizing larger errors.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

These metrics together evaluate **accuracy, stability,** and **error sensitivity** of each model.

4.2 Model Evaluation and Results-The models were tested across six stocks — **Apple, Microsoft, Google, Reliance, Tata Steel,** and **Meta**.

- **Polynomial and RBF kernels** in k-SVR gave the best results across all datasets.
- **Linear SVR** worked well for near-linear relationships but failed under high volatility.
- **Sigmoid kernels** performed poorly, producing unstable or unrealistic predictions.

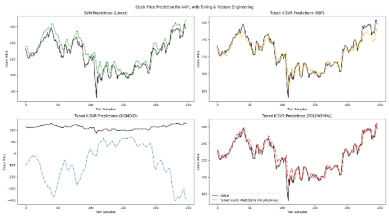
Metric	Tuned K-SVR (Polynomial Kernel)	SVM (Linear Kernel)	Tuned K-SVR (RBF Kernel)	Tuned K-SVR (Sigmoid Kernel)
Mean Absolute Error	8.533	12.195	18.726	214.423
Root Mean Square Error	11.734	15.630	23.438	281.558
R2 Score	0.941	0.872	0.721	-130.834

Fig. Model Evaluation Table pertaining the average evaluation metric

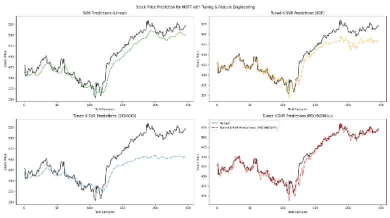
Key findings:

- **Apple, Google, Meta:** Polynomial kernel achieved the lowest MAE & RMSE, highest R².
- **Reliance, Tata Steel:** RBF and Polynomial kernels captured nonlinear dependencies effectively.
- **Microsoft:** Polynomial kernel managed both trend following and volatility modeling best.

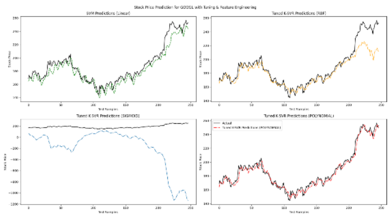
Overall, **Polynomial k-SVR** consistently achieved **R² > 0.9**, showing its strong ability to model both nonlinear and cyclical stock behaviors.



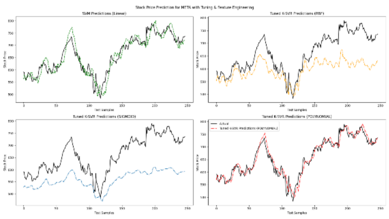
(a) Apple Stocks



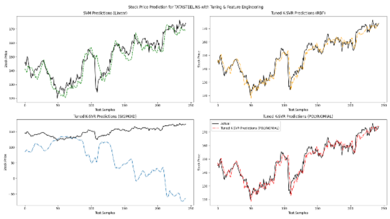
(b) Microsoft Stocks



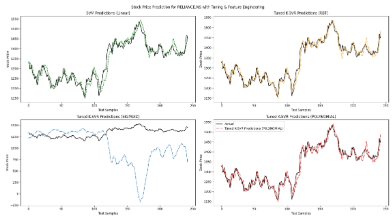
(c) Google Stocks



(d) Meta Stocks



(e) Tata Steel Stocks



(f) Reliance Stocks

Fig. Plots of predicted data using various kernels vs real data

	SVM (Linear Kernel)	Tuned K-SVR (RBF Kernel)	Tuned K-SVR (Sigmoid Kernel)	Tuned K-SVR (Polynomial Kernel)
Mean Absolute Error	6.616276	4.533641	300.423070	4.022545
Root Mean Square Error	8.728015	6.271062	347.751377	5.764101
R2 Score	0.789323	0.891240	-333.443647	0.908114

(a) Apple Stocks

	SVM (Linear Kernel)	Tuned K-SVR (RBF Kernel)	Tuned K-SVR (Sigmoid Kernel)	Tuned K-SVR (Polynomial Kernel)
Mean Absolute Error	14.220027	15.930308	29.234030	7.192258
Root Mean Square Error	16.525687	20.589762	36.796407	10.700711
R2 Score	0.890715	0.830354	0.458186	0.954179

(b) Microsoft Stocks

	SVM (Linear Kernel)	Tuned K-SVR (RBF Kernel)	Tuned K-SVR (Sigmoid Kernel)	Tuned K-SVR (Polynomial Kernel)
Mean Absolute Error	6.935320	7.874298	372.611647	3.787487
Root Mean Square Error	8.882381	12.956595	526.534216	4.918928
R2 Score	0.900964	0.789274	-347.008355	0.969628

(c) Google Stocks

	SVM (Linear Kernel)	Tuned K-SVR (RBF Kernel)	Tuned K-SVR (Sigmoid Kernel)	Tuned K-SVR (Polynomial Kernel)
Mean Absolute Error	20.994262	62.902506	98.957445	14.299351
Root Mean Square Error	26.363494	76.762102	108.229470	18.880794
R2 Score	0.875087	-0.059002	-1.105204	0.935932

(d) Meta Stocks

	SVM (Linear Kernel)	Tuned K-SVR (RBF Kernel)	Tuned K-SVR (Sigmoid Kernel)	Tuned K-SVR (Polynomial Kernel)
Mean Absolute Error	4.020270	2.716633	90.444354	2.665245
Root Mean Square Error	5.338815	3.664731	115.538485	3.626646
R2 Score	0.858428	0.933293	-65.304321	0.934672

(e) Tata Steel Stocks

	SVM (Linear Kernel)	Tuned K-SVR (RBF Kernel)	Tuned K-SVR (Sigmoid Kernel)	Tuned K-SVR (Polynomial Kernel)
Mean Absolute Error	20.388515	17.398832	396.868536	17.233805
Root Mean Square Error	25.945116	22.384298	551.497128	22.514214
R2 Score	0.921201	0.941346	-34.603803	0.940663

(f) Reliance Stocks

Fig. Evaluation metrics of models for various stocks

5 – Conclusion and Future Scope-

The study concludes that **Kernel-based SVR**, especially with the **Polynomial kernel**, offers the **best performance** for stock price prediction among all tested models. Its ability to capture complex, nonlinear relationships leads to more accurate and robust forecasts.

Future enhancements include:

1. Implementing **deep learning architectures** (LSTM, GRU, CNN-LSTM) for better temporal modeling.
2. Integrating **Big Data sources** like news sentiment, global indices, and social media analytics.
3. Applying **uncertainty quantification** to improve real-world decision-making and risk assessment.

In summary, **Polynomial k-SVR** provides a powerful, interpretable, and efficient framework for **stock market forecasting**, aiding investors in **data-driven portfolio management**.

References

1. Stock price prediction using support vector regression on daily and up to the minute prices | Bruno Miranda Henrique, Vinicius Amorim Sobreiro, Herbert Kimura
2. Stock Price Prediction Using Support Vector Machine Approach | Naliniprava Tripathy
3. Relative Strength Index (RSI): What It Is, How It Works, and Formula
4. Simple Moving Average (SMA). A stock's average closing price over a specified period
5. M. Narang, “How to Calculate R squared in Linear Regression,” 16 November 2022.
6. Major Kernel Functions in Support Vector Machine (SVM)
7. Machine Learning for Algorithmic Trading by Stefan Jansen

Appendix

- Link to Google Colab Notebook: M25MAC001_M25MAC002_M25CSE006_ML Project.ipynb
- Yahoo Finance API documentation link