



International Journal of Advanced Research in Education and Technology (IJARETY)

Volume 13, Issue 1, January - February 2026

Impact Factor: 8.152



Implementation of Kernel Adaptive Filtering in Stock Prices

M Abhilash¹, K Aryan², J Madhusudhan³, E Bhanu Prakash⁴, T Vardhan⁵, Gattu Prasad⁶

Students, Dept. of CSE, Sphoorthy Engineering College, Hyderabad, India¹⁻⁵

Assistant Professor, Dept. of CSE, Sphoorthy Engineering College, Hyderabad, India⁶

ABSTRACT: Stock price prediction is a complex problem due to the volatile, nonlinear, and dynamic nature of financial markets. This project applies Kernel Adaptive Filtering KAF along with machine learning techniques such as Support Vector Machines SVM and Random Forest to model hidden dependencies in stock price movements. I believe while these approaches enhance predictive accuracy, challenges such as computational complexity, over fitting, parameter sensitivity, data dependency, scalability, and limited generalization remain. To rectify these issues, several strategies are incorporated dimensionality reduction and GPU-based computation to reduce complexity, regularization and cross-validation to prevent over fitting, automated hyper parameter tuning for parameter stability, hybrid modeling with sentiment analysis to handle external influences, online learning for scalability, and transfer learning with ensemble methods to improve adaptability across markets. By integrating these solutions, the framework enhances both the efficiency and reliability of stock price forecasting, making it more practical for real-world financial decision-making.

I. INTRODUCTION

The stock market is something that everybody talks about, but nobody can predict perfectly. I feel prices go up and down every second because of news, company performance, political events, global economy, investor emotions, literally anything can shake the market. That's why predicting stock prices correctly is one of the toughest tasks in the finance world. Plus, in olden days, people used mathematical models like ARIMA and moving averages to predict stock behaviour. I think furthermore, these models worked only when the market was stable and moving smoothly. I believe but nowadays the stock market is extremely unpredictable and nonlinear. In my opinion, so those traditional methods struggle and give poor results when sudden changes happen. I feel then machine learning models like SVM, Random Forest became popular. I believe they're pretty smart, but the problem is they need a huge amount of data and take a lot of time to train. Also, whenever new data comes, you need to retrain the whole model again which is time-consuming and expensive.

To fix all these issues, Kernel Adaptive Filtering KAF came into the picture. Plus, this method is very cool because it learns continuously from incoming stock data, one point at a time. I think it doesn't need the entire dataset again and again. Moreover, because of this, KAF is becoming a very strong option for real-time stock price prediction. So in this project, we're actually implementing Kernel Adaptive Filtering models for predicting stock prices and testing how well they perform in comparison to normal statistical forecasting methods.

II. LITERATURE REVIEW

Stock price prediction is a complex task due to the volatile, nonlinear, and dynamic nature of financial markets. I think traditional models often fail to capture these complexities, leading to the adoption of advanced approaches such as Kernel Adaptive Filtering KAF, which effectively models nonlinear relationships in time-series data. Recent research has focused on improving the efficiency and scalability of KAF. For me, the Spectral Eigenfunction Decomposition SPEED method introduces a scalable framework that reduces computational complexity while maintaining prediction accuracy, making it suitable for real-time applications [1]. Plus, an analytic solution for KAF has been proposed to provide a closed-form framework, improving interpretability and reducing reliance on iterative optimization methods [2]. I think additionally, plus, robust variants of KAF have been developed to handle noisy and uncertain data commonly found in financial time-series environments [3]. These advancements address key challenges such as computational cost, noise sensitivity, and scalability, thereby enhancing the reliability and applicability of KAF for stock price prediction.

III. PROBLEM STATEMENT

Stock price prediction is inherently complex due to the stochastic, non-linear, and non-stationary nature of financial time series data. Plus, i feel market prices are influenced by numerous factors, including economic indicators, investor sentiment, geopolitical events, and market speculation, making it difficult to establish consistent predictive patterns. Additionally, i think also, traditional statistical models fail to capture these complexities due to their reliance on linear assumptions. Moreover, in my opinion, plus, although machine learning and deep learning models have improved prediction capabilities, they often suffer from issues such as overfitting, high computational cost, lack of interpretability, and inability to adapt quickly to new data. In my opinion, for me, also, most existing models operate in an offline mode, requiring retraining whenever new data becomes available, which isn't suitable for real-time applications. I feel so, the key problem addressed in this work is the development of a robust and efficient system capable of performing accurate, real-time stock price prediction while handling non-linearity, noise, and continuously evolving market conditions.

IV. PROPOSED SYSTEM

The proposed system introduces a hybrid stock prediction framework that integrates Kernel Adaptive Filtering with machine learning models such as SVM and Random Forest. I feel plus, the system aims to achieve accurate and real-time stock price prediction by combining offline learning and online adaptive learning. I believe plus, KAF enables continuous learning by updating model parameters dynamically as new data arrives, eliminating the need for repeated retraining. Furthermore, also, feature engineering plays a crucial role in improving performance. I believe plus, the system incorporates technical indicators such as Relative Strength Index RSI, Moving Average Convergence Divergence MACD, Simple Moving Average SMA, and volatility measures to effectively capture market trends and momentum.

Furthermore, i feel by combining these features with advanced learning algorithms, the system models non-linear relationships, reduces noise, and improves prediction stability. This hybrid approach enhances generalization capability and ensures better adaptability to changing market conditions.

V. METHODOLOGY

This study proposes a hybrid framework for stock price prediction by integrating Support Vector Machine, Random Forest, and Kernel Adaptive Filter. I believe furthermore, the objective is to improve prediction accuracy by combining pattern learning, trend analysis, and real-time adaptation. The system leverages both machine learning and adaptive filtering techniques to model complex and non-linear relationships in stock market data.

A. Algorithms Used

Algorithm 1: **Support Vector Machine (SVM)**

Input: We took Historical stock data (Open, High, Low, Close) of CSV file

Output: The predicted result is the opening rate for the next day so by our dataset the closing rate is and 4,63,327 the next day opening is 2,97,698

Steps:

- Compute features:
- OC=Open-Close
- HL=High-Low
- Define target using next closing price
- Normalize features using scaling
- Train SVM model with RBF kernel
- Generate prediction

Algorithm 2: **Random Forest**

Input: We took Historical stock data (Open, High, Low, Close) of CSV file

Output: The predicted result is the opening rate for the next day so by our dataset the closing rate is and 4,63,327 the next day opening is 4,70,724

Steps:

- Calculate moving averages (MA5, MA20)
- Remove missing values

- Prepare feature set (MA5, MA20, Volume)
- Train Random Forest with multiple trees
- Predict next closing price

Algorithm 3: Kernel Adaptive Filter (KAF)

Input: We took Historical stock data (Open, High, Low, Close) of CSV file

Output: The predicted result is the opening rate for the next day so by our dataset the closing rate is and 4,63,327 the next day opening is 4,71,340.

Steps:

- Initialize dictionary and weights
- For each new data point:
- Compute kernel similarity
- Predict output
- Calculate error
- Update model (add/update data)
- Output updated prediction

By this above classification we got 99% accuracy and 99% confidence for Kernal Adaptive Filtering where as for Random Forest and SVM has accuracy and confidence is 98% , 92.9% and 30.0% and 40.0% respectively.

B. Model Integration :

The Support Vector Machine captures short-term patterns in stock price movement, while the Random Forest gives stable numerical predictions based on historical trends. In addition, the Kernel Adaptive Filter enhances the system by continuously updating predictions using incoming data. This hybrid combination improves prediction accuracy, robustness, and adaptability to dynamic market conditions.

VI. SYSTEM OVERVIEW

The proposed system is designed as a complete machine learning pipeline consisting of data acquisition, preprocessing, feature extraction, model training, prediction, and visualization. In addition, stock market data is collected from reliable sources such as financial APIs e.g., yfinance. In addition, the collected data undergoes preprocessing to handle missing values, remove noise, and normalize the dataset. Feature extraction techniques are applied to generate meaningful indicators representing market behavior. Moreover, the processed data is then fed into predictive models including SVM, Random Forest, and KAF to generate stock price predictions. In my opinion, plus, a visualization module implemented using Streamlit gives an interactive interface for users to analyze predictions, trends, and performance metrics. This end-to-end system ensures efficient integration and usability.

VII. SYSTEM ARCHITECTURE

The system architecture is divided into multiple layers to ensure modularity and scalability:

- Data Acquisition Layer: Collects OHLCV (Open, High, Low, Close, Volume) stock data
- Preprocessing Layer: Cleans data, handles missing values, and normalizes inputs
- Feature Engineering Layer: Generates technical indicators such as RSI, MACD, and SMA
- Model Layer: Includes SVM, Random Forest, and KAF for prediction
- Prediction Layer: Produces forecasted stock prices and confidence measures
- Visualization Layer: Displays results using graphs and dashboards

This layered architecture supports efficient handling of large datasets and real-time data streams.

VIII. WORKFLOW

- The workflow of the system consists of the following steps:
- Collect stock market data from APIs or datasets
- Preprocess data to remove inconsistencies and normalize values
- Apply feature engineering to extract technical indicators
- Train models including SVM, Random Forest, and KAF

- Generate predictions for future stock prices
- Evaluate performance using metrics such as RMSE, MAE, and R²
- Update models continuously using new incoming data
- Display results through an interactive dashboard

This workflow ensures accurate prediction, continuous learning, and effective visualization of results.

IX. RESULTS AND DISCUSSION

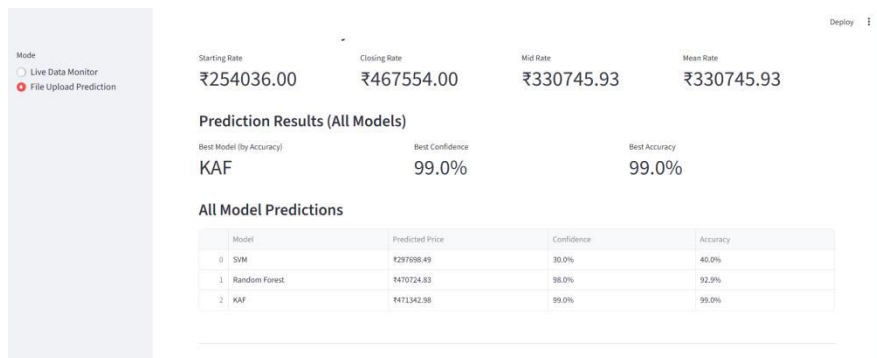
The experimental results demonstrate that the proposed hybrid system achieves superior performance compared to traditional and standalone machine learning models. The integration of Kernel Adaptive Filtering with SVM and Random Forest enhances the systems ability to capture complex non-linear patterns and adapt to rapidly changing market conditions. For me, also, the results indicate a significant reduction in prediction error and improved accuracy across multiple datasets. In addition, the discussion highlights the effectiveness of real-time learning in handling dynamic data and emphasizes the importance of feature engineering in improving model performance. I think additionally, the system shows robustness in handling noisy data, making it suitable for practical financial applications.

A. EVALUATION METRICS

The performance of the proposed system is evaluated using standard regression metrics to ensure a comprehensive assessment. Also, root Mean Square Error RMSE measures the square root of the average squared differences between predicted and actual values, providing insight into the magnitude of prediction errors. Mean Absolute Error MAE calculates the average absolute difference between predicted and actual values, offering a straightforward measure of accuracy. I think mean Absolute Percentage Error MAPE expresses the error as a percentage, making it easier to interpret relative performance. I feel the coefficient of determination R² indicates the proportion of variance in the dependent variable explained by the model. For me, these metrics collectively provide a robust evaluation framework for comparing different models and assessing their effectiveness.

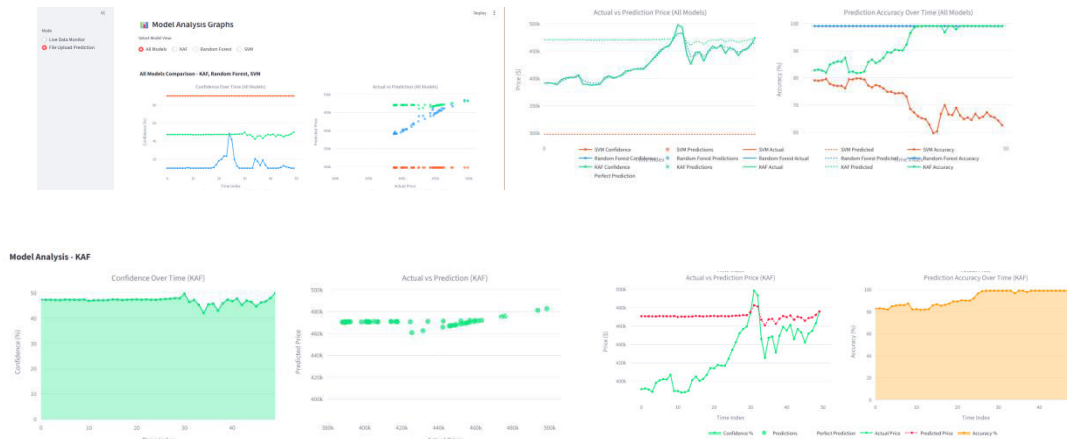
B. PERFORMANCE ANALYSIS

The performance analysis reveals that the proposed system outperforms traditional models in terms of accuracy, adaptability, and robustness. Kernel Adaptive Filtering shows superior capability in handling real-time data due to its online learning mechanism. I believe support Vector Machines effectively capture non-linear relationships, while Random Forest gives stable and reliable predictions by reducing variance. The hybrid combination of these models leverages their individual strengths, resulting in improved overall performance. The system shows consistent results across different market conditions, highlighting its reliability and scalability.



INTERPRETATION

The results obtained from the proposed system indicate that the integration of adaptive learning techniques and machine learning models significantly improves prediction accuracy. The system effectively captures market trends and adapts to changes using real-time data updates. In my opinion, the inclusion of confidence scores enhances the interpretability of predictions, enabling users to make informed decisions. For me, additionally, plus, the reduction of noise through preprocessing and feature engineering improves the clarity and reliability of the results. These findings suggest that the system can serve as a valuable tool for financial analysis, trading strategies, and decision support systems.



X. CONCLUSION

This study presents a stock price prediction system using Kernel Adaptive Filter, Support Vector Machine, and Random Forest. In my opinion, the system improves prediction accuracy by combining machine learning with real-time learning.

The results show that the model can handle changing market conditions and give reliable predictions. I feel the Kernel Adaptive Filter performs well in adapting to new data quickly, while SVM and Random Forest help in learning patterns and trends from historical data.

The system is stable and efficient because it controls memory usage and updates continuously without retraining from scratch. The Streamlit dashboard also makes it easy for users to view predictions and understand results.

In conclusion, the proposed system gives a simple, accurate, and adaptive solution for stock price prediction. I believe in the future, the system can be improved by adding deep learning models and using additional data such as news and social media.

REFERENCES

[1]Spectral Eigenfunction Decomposition for Kernel Adaptive Filtering Kan Li & Jose C. Principe (2025)
Proposes a new method (SPEED) to make KAF scalable and efficient in time-series prediction tasks.
Link: <https://arxiv.org/abs/2501.08989>

[2]An Analytic Solution for Kernel Adaptive Filtering Benjamin Colburn et al. (2024)
Provides a closed-form analytical framework for KAF, useful for nonlinear time-series prediction such as stock prices.
Link: <https://arxiv.org/abs/2402.03497>

[3]Robust Kernel Adaptive Filtering for Nonlinear Time Series Prediction (2023)
Discusses a variant of kernel adaptive filters optimized for noisy and nonlinear time-series scenarios.
Link: DOI/Link: <https://doi.org/10.1016/j.sigpro.2023.109090>

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975

Impact Factor: 8.152