International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Machine Learning-based historical data analysis and future trend forecasting

R. Kirubahari[1], P. Kiruthika[2], S. Lakshita[3], J.M.Namritha Shree[4]

[1]*Associate Professor, Department of Computer Science and Engineering, K.L.N. College of Engineering*

[2,3,4] *Final Year Students, Department of Computer Science and Engineering, K.L.N. College of Engineering*

Abstract - Machine learning-based historical data analysis and future trend forecasting aims to predict future trends in areas such as market share, cricket scores, and weather patterns by leveraging historical data. This study explores the application of machine learning models—specifically Random Forest, XGBoost, and Prophet—for historical context retrieval and predictive analytics. By training these models on historical datasets, the system forecasts future values and identifies emerging trends that can support informed decision-making across diverse sectors. The methodology involves extracting relevant historical information from large datasets and applying predictive models to generate forecasts for the coming years. Visualizations, including bar and line charts, are used to clearly present comparisons between past performance and future projections. The results demonstrate that combining historical data retrieval with machine learning algorithms significantly enhances predictive accuracy, providing valuable insights for businesses, sports analysts, and meteorologists. The study also highlights challenges such as data quality and dynamic changes in trends, and concludes by suggesting future directions for improving the forecasting process.

Key Words:Trend Forecasting, Machine Learning, Historical Data Retrieval.

1.INTRODUCTION

The model leverages historical trends and patterns to enable accurate forecasting for future developments in areas such as cricket scores, market share, and weather conditions. By extracting meaningful insights from past data, it supports informed decision-making across sectors. Advanced machine learning algorithms like Random Forest and XGBoost are used to capture complex, non-linear relationships in the data, while Prophet specializes in forecasting seasonal and time-dependent trends. Data preprocessing techniques such as Standard Scaler ensure uniformity and improve model performance. Additionally, visualization tools like text wrapping and chart generation enhance the readability and interpretability of the predictions.This hybrid approach of historical context retrieval combined with machine learning significantly boosts forecasting accuracy, reliability, and performance. By applying optimized statistical techniques, the model effectively mitigates common limitations in historical data, such as inconsistencies and missing values.

2. OBJECTIVE

The objective of this project is to analyze historical data patterns and accurately forecast future trends using advanced machine learning techniques. By integrating models such as Random Forest and XGBoost Regressors, the system aims to capture complex relationships in data while addressing challenges like data sparsity through effective imputation methods. Seasonal forecasting is enhanced using the Prophet model, and feature normalization is achieved with Standard Scaler to ensure consistent and reliable performance. The project focuses on forecasting across multiple domains such as market share, cricket scores, and weather patterns by accepting dynamic user inputs. Natural Language Processing (NLP) techniques are used to extract relevant data from both structured and unstructured sources. The system emphasizes user-friendly visualization through unified graphs, enabling clear comparison between historical and predicted values. Key metrics like the R^2 score are used to evaluate and improve model accuracy. Additionally, the system offers clean text formatting for better readability and supports exporting results in CSV and JSON formats. Designed for scalability and real-time response, the framework lays a strong foundation for future integration with deep learning models, aiming to enhance accuracy and handle complex long-term data patterns.

3. LITERATURE SURVEY

3.1. Machine Learning in COVID-19 Forecasting:

Traditional epidemiological models for pandemic forecasting often require deep domain expertise and involve complex mathematical modeling. However, the COVID-19 crisis has led researchers to explore machine learning (ML) techniques as effective alternatives for predicting the future trajectory of the pandemic. In this study, four supervised ML

modelsLinear Regression (LR), LASSO, Support Vector Machine (SVM), and Exponential Smoothing (ES)were applied to forecast critical factors such as new confirmed cases, death rates, and recoveries over a 10-day horizon. These models were trained on time-series data sourced from Johns Hopkins University's repository, capturing daily global case updates. Among the models, ES consistently delivered the most accurate forecasts, especially when data was limited, while SVM underperformed due to the dataset's high variability. ML approaches in this context demonstrate promise in real-time pandemic forecasting, enabling governments and healthcare providers to make timely, data-driven decisions.

3.2. Feature Influence and Model Evaluation in COVID-19 Predictions:

The precision of ML predictions in pandemic forecasting hinges on selecting the right features and evaluating model efficacy using robust statistical metrics. In this research, daily time-series featuressuch as the number of confirmed cases, recoveries, and deathsserved as core inputs. To optimize model performance, data preprocessing and careful feature representation were critical. Model outputs were assessed using R-squared (R^2), Adjusted R-squared, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These metrics helped quantify model accuracy and consistency. Linear models like LR and LASSO improved as the dataset grew, showcasing the impact of data volume on predictive strength. Although interpretability tools like SHAP or LIME were not explicitly used, the structured evaluation of input variables and the effect of regularization in LASSO indirectly underscored the importance of meaningful feature contributions. Ultimately, the combination of strategic feature use and methodical model evaluation reinforces confidence in using ML for forecasting disease spread.

4.EXISTING SYSTEM

The forecasting trends such as market share, cricket scores, and weather patterns relies heavily on traditional machine learning models and historical price data, but it suffers from several critical limitations. These systems often lack meaningful user interaction, which reduces the overall accuracy of predictions and can lead to biased outputs. In many studies and applications, researchers use standardized or benchmark datasets that do not reflect the complexity or variability found in real-world environments. As a result, the systems fail to adapt when deployed in practical scenarios. Moreover, many existing models struggle due to inefficient hyperparameter tuning, which directly impacts their prediction power and generalization capabilities.Market forecasting models typically depend on historical price movements and basic technical indicators, assuming these alone are sufficient to capture future fluctuations. However, these assumptions overlook many dynamic, external factors such as geopolitical events, consumer sentiment, or seasonal variations. In environments such as stock markets or sports analytics, where data changes rapidly and unpredictably, the reliance on static patterns proves to be highly ineffective.

5. PROPOSED SYSTEM The system presents an advanced machine learning-based framework for historical data analysis and future trend forecasting, integrating multiple sophisticated algorithms and data processing modules to address challenges like data sparsity, seasonality, and scalability. This system primarily employs Random Forest Regressor, XGBoost Regressor, and Prophet, each tailored for specific aspects of prediction. The Random Forest Regressor enhances predictive performance by creating a multitude of decision trees and merging their outputs to reduce overfitting and increase accuracy. It captures complex nonlinear relationships from input features, making it ideal for handling real-world, unstructured datasets like market trends or weather reports. The XGBoost Regressor provides a scalable and high-performance solution through gradient boosting, known for its speed and efficiency. It handles missing data, prevents overfitting through regularization, and is particularly effective for structured datasets like stock market predictions. Meanwhile, Prophet, developed by Facebook, specializes in modeling time-series data with strong seasonal effects and historical trends, offering robustness against missing data and irregular sampling.

The system begins with user input modules that collect parameters such as country, state, and city, followed by text extraction using NLP techniques. These modules convert both structured and unstructured text into a consistent format, handling different encodings like UTF-8 and ASCII and removing irrelevant symbols

6. ARCHITECTURE DIAGRAM

The architecture diagram outlines the full workflow of a machine learning-based system designed to process user-provided textual input, perform

predictions, and display results visually. The flow begins with User Input, where a user provides data such as a news article. This input is simultaneously stored in a Database and passed to the Text Extraction module, where Natural Language Processing (NLP) is used to identify important content and context. The extracted data is then forwarded to the Model Selection & Preprocessing phase, which standardizes and prepares the data using techniques like Standard Scaler, ensuring it is properly formatted for model training.Depending on the scenario, one of three predictive models Random Forest, XGBoost, or Prophetis selected for training and forecasting. Once predictions are generated, the results are sent to the Statistical Visualization module, where graphical representations such as charts and trend lines are created to help users understand the insights. Finally, these visual outputs are passed through the Text Formatting and Display module to ensure clarity and presentation quality, making the results user-friendly and ready for interpretation. This two-part flow ensures seamless integration from raw user input to insightful visual outputs.

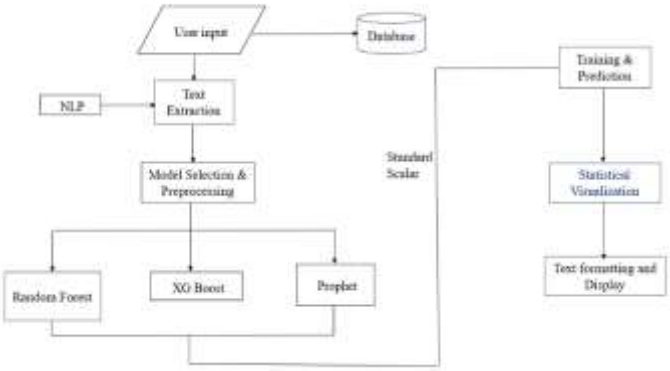

Figure1: Architecture Diagram

7. SYSTEM OVERVIEW

7.1. Data Input:

The Data Inputserves as the foundation of the entire system, acting as the first point of interaction between the user and the machine learning framework. This module begins with User Input Handling, where users provide essential details such as country, state, city, and time parameters (like year or "overall"). The module is designed to intelligently process this input, even accounting for possible inaccuracies, missing fields, or formatting errors. It ensures smooth interaction through real-time feedback mechanisms that notify the user if their input is invalid or incomplete, enhancing the usability and responsiveness of the application. Following successful input capture, the module moves to Data Retrieval, where it fetches relevant historical datasets based on the user's specifications. Whether it is market trends, cricket scores, or temperature records, the system queries a structured backend database to extract only the necessary fields. This targeted retrieval not only improves performance by reducing unnecessary data loads but also enhances accuracy by narrowing the scope to user-intended parameters.

Figure 2:Data Input

7.2. Text extraction and Data Handling:

The Text Extraction is primarily responsible for analyzing the user-provided news article text, typically focused on a specific year. Once the user inputs the article, the module treats it as contextual input, aiming to link real-world events or trends with the predictive models. More advanced versions of this module can be integrated with NLP libraries like spaCy or transformers to extract keywords, sentiments, or named entities that could influence predictions.This extracted text could then be categorized or tagged based on content. For example, a news piece mentioning "floods," "high inflation," or "sports victories" could help refine or filter the prediction models for weather, market performance, or cricket outcomes, respectively. These tags or signals are not only useful for contextual awareness but can also serve as soft constraints or influencing factors in future models.The Data Handling Module acts as the core processor of all incoming and outgoing datasets, serving as the bridge between user input, predictive modelling,and visual output. This module is responsible for storing, formatting, cleaning, and managing three main datasets: market share, cricket scores, and weather trends. Each dataset is structured as a pandas. DataFrame, allowing for high-performance data manipulation and compatibility with machine learning models.

.

7.3. Data Preprocessing & Transformation:

The Data Preprocessing module plays a crucial role in preparing the raw historical datasets for machine learning model consumption. In this system,

the datasets include market share, cricket scores, and weather data, all indexed by year. Initially, the input features, which are the years ranging from 2015 to 2025, are extracted and reshaped into a two-dimensional array. Simultaneously, the target variable (i.e., market share values, cricket scores, or temperatures) is isolated. Since most ML algorithms perform better with normalized data, a StandardScaler is applied to scale the feature values (years) into a standard distribution with a mean of 0 and variance of 1. This standardization ensures that models like Random Forest and XGBoost are not biased due to scale differences, and it stabilizes the training process.

Different models require different input structures, so the transformation module adjusts the data format accordingly. For Random Forest and XGBoost, the scaled year values and target features are directly used to fit the models. However, the Prophet modelused for time-series forecastingrequires a unique format, the DataFrame must have two specific columns named ds (for datetime) and y (for the target variable). The transformation module handles this by converting the "Year" column into datetime format and renaming it to ds, while the target value is labeled as y. This structured conversion allows Prophet to identify seasonal trends and make reliable future predictions. These model-specific transformations make this module essential for flexible integration with diverse ML techniques.

7.4. Text Formatting & Processing:

The Text Formatting begins its function as soon as the user inputs a news article. Since this article serves as a narrative reference for the system's prediction outputs, it is essential to present it in a clear and structured format. The module trims unwanted white spaces and ensures the text is clean, concise, and grammatically structured for display. Basic formatting tasks such as quotation addition, bullet styling, and paragraph alignment are handled here to make the content visually appealing and readable in the console or graphical interface. This not only improves presentation quality but also helps in keeping user attention focused on key data points during interpretation. The Text Formatting & Processing play a key role in integrating formatted user input with data insights during output generation. When future predictions are visualized, the article is reintroduced at the top as a summary section, ensuring continuity in the user's interaction flow. The module ensures the seamless merging of narrative and analytics by clearly demarcating the summary from numerical outputs and graphical plots. This thoughtful structuring ensures that users interpret the data within the storyline they provided, which enhances both usability and engagement.

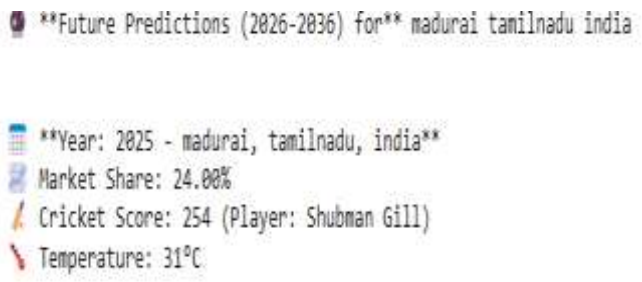

Figure 3:Text Formatting

7.5. Statistical & Randomization:

The Randomization module plays a crucial role in creating synthetic datasets for the system, especially when real-world data is not accessible or to simulate unpredictable variables. It uses Python's random library to generate values for market share percentages, cricket scores, and weather temperatures between defined ranges. For example, cricket scores are randomly generated within a range of 200 to 450, while weather temperatures vary between 15°C and 40°C. This ensures that the demo remains dynamic and varied every time the code runs, allowing users to explore how the predictive system adapts to different initial conditions. The randomness mimics real-world fluctuations and introduces variability, making the simulation more engaging and lifelike.

7.6. Data Visualization & Analysis:

The Data Visualization module plays a critical role in converting complex numerical datasets into intuitive and engaging visuals. It uses powerful libraries such as Matplotlib and Seaborn to create graphs and charts that represent historical and predicted trends over a span of years. These visuals include line plots that show the evolution of parameters like market share, weather patterns, and cricket scores from 2015 to 2025, and their forecasts from 2026 to 2036. With visual markers, color distinctions, and labels, users can easily differentiate between past data and future predictions, improving the clarity of interpretation. Together, the Data Visualization and Analysis Modules form a storytelling mechanism that brings the dataset to life. Whether the user is studying climate shifts, market

fluctuations, or cricket statistics, the modules present a narrative that bridges raw data with real-world context. They serve as a foundation for cross-domain applications where trends can influence policies, business strategies, or even sports commentaries. By visualizing the evolution and prediction of various metrics, these modules help transform static numbers into actionable knowledge.

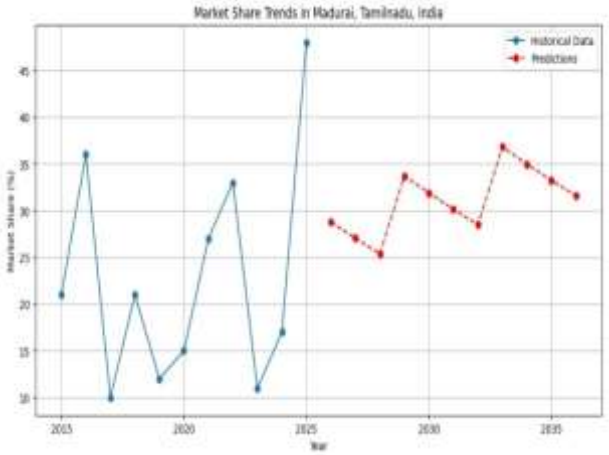

Figure 4:Market Share Overview

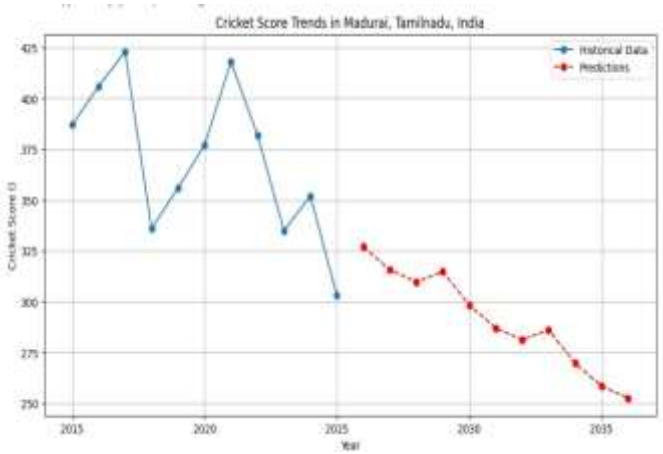

Figure 5:Predicted Cricket Score Highlight

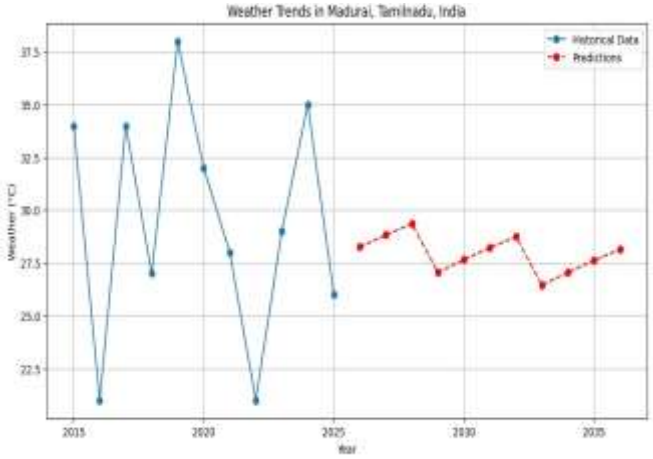

Figure 6:Predicted Weather Trends

8. FUTURE ENHANCEMENT

In the future enhancement, the system can be extended by integrating real-time data pipelines using APIs from live news feeds, weather stations, and sports databases. This real-time integration will enable dynamic updating of predictions, offering more time and context-aware insights for users. Additionally, introducing advanced NLP models such as BERT or GPT-based transformers can help in deeper semantic understanding of news articles, allowing the system to detect nuanced patterns and sentiments that influencemarket trends, weather anomalies, or sports outcomes more accurately.

Further improvements can be achieved by adopting deep learning techniques like LSTM (Long Short-Term Memory) and attention-based Transformer networks for time-series forecasting. These models excel at capturing long-term dependencies and temporal patterns, making predictions more robust even with irregular or sparse historical data. Combining these models with ensemble learning strategies could also mitigate individual model weaknesses, leading togreater overall accuracy, adaptability, and resilience against noise or incomplete inputs.Enhancing the user interface for better interactivity and visual analytics would further empower end-users with intuitive control and understanding of forecast outputs. Incorporating federated learning can further improve privacy and security by training models locally on user devices without transferring sensitive data. Adding voice input and multilingual support can expand accessibility.

9. CONCLUSION

The integration of historical data with advanced machine learning algorithms has revolutionized predictive analytics, enabling more accurate forecasts across various domains such as market trends, sports outcomes, and weather patterns.By employing ensemble methods like Random Forest and XGBoost, the system effectively captures complex, non-linear relationships within the data, leading to robust and reliable predictions.Random Forest operates by constructing multiple decision trees and aggregating their outputs, which enhances predictive accuracy and mitigates overfitting.On the other hand, XGBoost builds trees sequentially, with each new tree correcting errors made

by the previous ones, thus incrementally improving the model's predictions.This approach incorporates regularization techniques to prevent overfitting, making it particularly effective for reducing bias and variance in complex datasets.

REFERENCES

[1]. B. Shi, G. Ifrim, and N. Hurley, ‘‘Learning-to-rank for real-time highprecision hashtag recommendation for streaming news,’’ in Proc. 25th Int. Conf. World Wide Web (WWW), 2016.

[2]. I.Verbitskiy, P. Probst, and A. Lommatzsch, ‘‘Development and evaluation of a highly scalable news recommender system,’’ in Proc. CLEF, 2015, pp. 1–5.

[3]. Kapusta, J.; Obonya, J. Improvement of Misleading and Fake News Classification for Flective Languages by Morphological Group Analysis. Informatics 2020, 7, 4.

[4]. M. Khan, ‘‘Using text processing techniques for linking news stories for digital preservation,’’ Ph.D. dissertation, Fac. Comput. Sci., Preston Univ. Kohat, Islamabad, Pakistan, 2018.

[5]. M. Zihayat, A. Ayanso, X. Zhao, H. Davoudi, and A. An, ‘‘A utility-based news recommendation system,’’ Decis. Support Syst., vol. 117, pp. 14–27, Feb. 2019.

[6]. Q. Yu, L. Xu, and S. Cui, ‘‘Streaming algorithms for news and scientific literature recommendation: Monotone submodular maximization with a d-knapsack,’’ IEEE Access, vol. 6, pp. 53736–53747, 2018.

[7]. S. Baran and D. Nemoda, ‘‘Censored and shifted gamma distribution based EMOS model for probabilistic quantitative precipitation forecasting,’’ Environmetrics, vol. 27, no. 5, pp. 280–292, Aug. 2016.

[8]. Wang, T., Lu, K., Chow, K. P., & Zhu, Q. (2020). COVID-19 sensing: Negative sentiment analysis on social media in China via bert model. IEEE Access, 8, 138162–138169.

[9]. Y. Grushka-Cockayne and V. R. R. Jose, ‘‘Combining prediction intervals in the m4 competition,’’ Int. J. Forecasting, vol. 36, no. 1, pp. 178–185, Jan. 2020.

[10]. Zhang, W. Yoshida, T.Tang, X. Expert Syst. Appl. 2011, 38, 2758–2765 T.Zhu, Q. (2020). COVID-19 sensing: Negative sentiment analysis on social media in China via bert model. IEEE Access, 8, 138162–138169.