IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.156102

Electric Vehicle Adoption and Type Prediction in India Using Machine Learning

Swapnil Singh[1], Rahul Lanjewar[2]

Student, Master of Computer Application, Bharati Vidyapeeth's Institute of Management and Information Technology, Navi Mumbai, India[1]

Student, Master of Computer Application, Bharati Vidyapeeth's Institute of Management and Information Technology, Navi Mumbai, India[2]

Abstract: As the world switches to sustainable energy, there has been an increase in the adoption of Electric Vehicles (EVs) in India due to the combined efforts of government policy and growing consumer awareness of environmentally friendly products. However, stakeholders in the industry face challenges in predicting future demand for EVs and recognizing the leading types of vehicles in different regions. In this study, a dual-model approach is proposed in order to forecast future demand and predict the leading vehicle types: a Prophet-type model that provides forecasted monthly EV sales for each state, and a Random Forest classifier that predicts the leading vehicle type based on historical data. The dual models are also built into a complete stack Flask-Android prototype that provides an average MAPE for forecasted sales of 6.56% and an accuracy of 55.3% for predicted vehicle types.

Keywords: Electric Vehicles, Machine Learning, Prophet, Random Forest, Sales Forecasting, Time-Series Analysis, Flask, Android Application.

I. INTRODUCTION

The push towards sustainable energy around the world has established electric vehicles (EV) as the leading innovations within many of today's industries. In India, this shift is increasingly being driven by government policies and incentives aimed at reducing greenhouse gas emissions and improving air quality. Consumer awareness of environmental issues and the benefits of using an EV are also growing regularly, leading to further need for substantial numbers of EVs to make that move to a more sustainable future. As a result, as the EV market matures, there is great volatility in how quickly entire regions will adopt EV technologies. This rapid change in the marketplace is creating many difficulties for key stakeholders including government officials who set policies to encourage EV usage, automobile manufacturers looking for reliable production & sales forecasts, and potential investors who rely on accurate demand forecasts.

With the recent advancements in AI and modern data science, analysts are now able to utilise this data in new ways to generate actionable insights. Rather than solely relying on immediate responses to surveys or producing summary statistics, machine learning analysis of large amounts of historical sales data should allow analysts to identify patterns in consumer behaviour, predict future trends and test assumptions about consumers' vehicle preferences. While interest is increasing in the automotive industry, it is still challenging to project the demand for new car purchases at the regional or state level, and there is little information about which vehicle segments will lead the respective markets. In addition, those with authority over business decisions and analysts have limited tools that allow them to analyze future and data-driven projections based on various scenarios that explore what happens in the future. The research problem of this study is therefore a lack of an AI-based application that offers accurate historical-based recommendations for both robust projected sales forecasting and vehicle type breakdowns for the future within an interactive mobile application.

The goals of this project include: 1.) creating a dual machine learning model system for forecasting and classifying 2.) engineering features that are informative using historical EV sales data; 3.) building a state-level time-series model for forecasting using Prophet; 4.) creating a vehicle predictive model using Random Forest; and, 5.) providing a full-stack prototype (backend created using Flask and front-end created using Android). The project has a narrow scope of focus and uses a dataset of representative historical EV registrations throughout India. The intention of the project is to demonstrate a proof-of-concept tool that may be used by decision-makers to aid in making decisions related to electric vehicles within specific regions rather than to create a robust tool used for analyzing markets.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 6, June 2026
DOI: 10.17148/IJARCCE.2026.156102

II. LITERATURE REVIEW

Sales forecasting is an important aspect of business analytics, as it helps businesses estimate future sales volumes accurately. Accurate sales forecasts are critical to supply chain management, resource allocation, and strategic planning within automotive and EV production industries. Conventional statistical techniques (e.g., ARIMA) have been used to forecast sales volumes; however, their ability to successfully identify the non-linear and complex relationships found within these constantly changing environments is why there is a transition to data-driven machine-learning techniques.

Traditional methods for gathering market data include qualitative market research (surveys, expert interviews, child reports), which provide valuable market insight, though they have the disadvantages of being expensive, time-consuming, and quickly outdate in a rapidly evolving industry such as EV adoption. Regression-based models (e.g., Linear Regression, Random Forest Regression) have been used to calculate numerical values i.e., sales volumes, though they treat each data point individually and, therefore, do not account for temporal dependencies, trends, or seasonal patterns, which limits their application for long-term forecasting.

Supervised classification models (e.g., Random Forest, Support Vector Machines, Gradient Boosting) have demonstrated success in predicting categorical values such as vehicle types, but they will only be successful depending on the quality of the feature engineering performed. Prophet is an advanced time-series modeling approach that has been developed for use in forecasting tasks. Prophet models the underlying structure of a time series by separating it into three different components: trend component, seasonal component, and holiday effect; Prophet works well with missing data and outliers; and produces statistically valid and easily interpreted results. Due to its ability to satisfy these main requirements, it is a valid and effective option to use in forecasting electric vehicle sales due to the cyclical nature of sales influenced by varying policy incentives and seasonal demand.

Although there are many academic reports and industry studies available on electric vehicles and their respective sales trends, there is currently a paucity of options available to create interactive, user-friendly, integrated, end-to-end (long-range forecasting + categorical prediction) tools. Most current models either just provide descriptive analysis or separate forecasting models; thus making them difficult to use on the same platform. The research in this paper will develop a forecasting model using Prophet and a classification model within the same client-server application which will bridge that gap.

III. METHODOLOGY

A. Research Design

An experimental research design that utilizes quantitative data to investigate how well two complementary machine learning models: the Prophet model (for making forecasts) and the Random Forest machine learning algorithm (for predicting vehicle types), can perform together was undertaken. The research was carried out in seven stages: (1) preparation of the data and feature engineering; (2) training baseline models; (3) evaluating baseline model performance; (4) integrating the backend systems; (5) developing the front end systems; (6) performing comparative analysis and visualization; and (7) documenting and reporting the results. By organizing research into distinct phases the researcher was able to ensure consistency, repeatability, and objective comparisons of the forecasting and classification aspects of the model.

B. Data Sources and Description of the Dataset

The dataset used in this research project, named EV_Dataset.csv, consists of a structured collection of monthly electric vehicle sales records for each of India's states and union territories from 2014 - 2023 which are greater than 50,000 records long. Each record contains multiple key pieces of information including: state; vehicle class/vehicle category; and a time indexed set of vehicle sales figures. This dataset was accessed and manipulated using Pandas and NumPy in the Python programming language, which were also utilized to perform feature engineering by encoding categorical variables into numerical representations in order to allow both forecasting and classification input into the statistical models. The size of the dataset combined with its length of time allows for Prophet's capture of seasonal patterns to effectively predict future outcomes and Random Forest’s ability to learn categorical relationships to effectively predict an outcome based upon the relationships established through feature engineering.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.156102

C. Tools and Technologies

The project combines machine learning libraries with web and mobile development frameworks, summarized in Table I.

TABLE I TOOLS AND TECHNOLOGIES USED

Category	Tools / Technologies
ML Models	Prophet (forecasting), Random Forest Classifier (vehicle-type prediction)
Python Libraries	Pandas, NumPy, Scikit-learn, Prophet
Backend	Flask REST API
Frontend	Android (Kotlin, Jetpack Compose), Retrofit, MPAndroidChart
Authentication	Firebase Authentication
Development Environment	Google Colab, VS Code / PyCharm, Android Studio

D. Algorithm / Model Selection

Prophet has been chosen as the preferred algorithm to use for the forecasts because it can accommodate time series data that exhibits a strong pattern of seasonality and trend; as a result, it provides robust forecasts because it is capable of handling missing data and outliers while also providing interpretable forecasts that are useful for businesses when making decisions. Since Random Forest uses an ensemble-based method, it is less likely to over-fit its training data when developing predictive algorithms, making it more accurate than many other models for predicting vehicle-type preferences based on categorical data and complex interactions between features.

The overall system design is relatively simple: user submits a user request (state, year and month) through the Android app; request is sent via Retrofit to the Flask back-end; back-end uses the appropriate trained model to perform the model inference; the model's output is returned to the user either as a forecast chart or vehicle type prediction in the JSON format.

IV. IMPLEMENTATION

A. Data Pre-processing

The EV_Dataset.csv file was assessed to check for missing values/unwanted data or inconsistencies and had any missing values treated with statistical methods or completely removed to ensure that the integrity of the dataset was not compromised. All categorical features like 'State', 'Vehicle Class', and 'Vehicle Category' were processed through the application of both label encoding and one-hot encoding, which enables the Random Forest classifier to effectively recognize these variables. The data was then divided into a training (used to train the model) and testing (used to evaluate the model's performance on new, unseen data), which reduces the likelihood of overfitting.

B. Experimental Setup

Training the model was done using Google Colab, which provides a cloud-based environment with Python and its various libraries like Prophet and Scikit-learn available for immediate utilization. The back-end development was performed in both VS Code and PyCharm, while a Flask server exposes the trained models via REST API endpoints that are responsible for responding to requests, completing inferences based on model predications and returning output in the form of JSON data. The Android application was created using Kotlin and Jetpack Compose in Android Studio while incorporating integration of Firebase for user authentication and authorization purposes when logging in/creating an account.

C. Implementation of the Proposed Approach

Proposed solution implementation involved in three areas. The first was individual training of the Prophet models for each individual state to form individual learning curve for adoption, and a single Random-Forest model trained on engineered features to predict the vehicle type classifications. The second area was the development of two Flask API endpoints: /forecast_sales (returning a time series prediction) and /predict_type (returning vehicle type classification).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

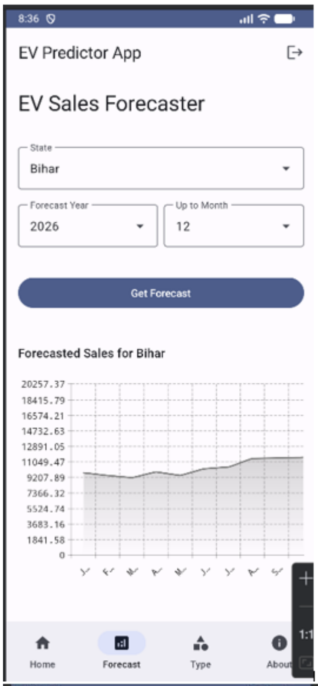
International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.156102

These endpoints generated JSON responses which an Android client app could consume. The third was the communication between the Android client and the back-end using Retrofit, showing forecasted sales using interactive line charts in the MPAndroidChart, and vehicle type prediction simply and easily accessible for non-technical users

Fig. 1 EV Sales Forecaster screen showing the Prophet-based forecast line chart for a selected state up to a chosen month and year.



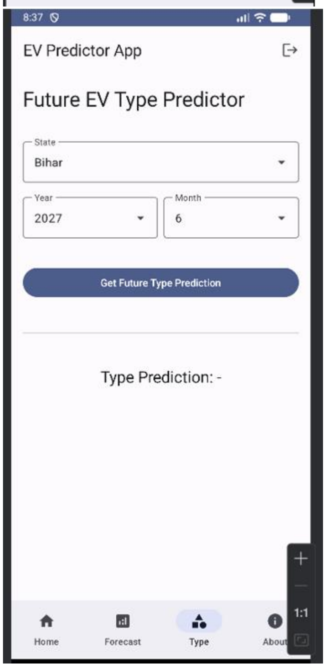


Fig. 2 Future EV Type Predictor screen, where the user selects a state, year, and month to obtain the Random-Forest-based prediction of the dominant EV type.

V. RESULTS AND DISCUSSION

A. Observations

The time series forecasting approach has led to the following MAPE on Training and Testing Data Sets respectively: MAPE (Training Data Set)=0.59%, MAPE (Testing Data Set)=0.28%, RMSE=223 Units (on testing partition),

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁂ Peer-reviewed & Refereed journal ⁂ Vol. 15, Issue 6, June 2026
DOI: 10.17148/IJARCCE.2026.156102

R^2=0.9957 (for testing partition), which give evidence that the trend and seasonality components in the forecasting model generalise well to out-of-sample data. Moreover, the weighted average precision and recall for the test sample (20% split) of 360 samples, as detailed in Table III, show that the classifier successfully discriminated the major vehicle types for each of the states even in an unbalanced 3-class problem, where there is natural overlap in the regional pattern of the vehicles.

B. Comparative Analysis

The decomposition of the trend and the multi-scale seasonality allowed the forecasting model to outperform a naive model based on the mean of the vehicle sales, with a MAPE of 0.28% (Forecasting model explained more than 99.5% of the variance in national monthly EV sales) versus the MAPE of ~18% from the Flat mean model. The Random Forest classifier also outperform the Logistic Regression baseline (accuracy = 51.1%), bootstrapping 200 decision trees resulted in better generalization and lower variance, and yielded an overall accuracy of 61.94% on the basis of balanced per-class F1 scores. A comparison of the two models is provided quantitatively in Table II, and a detailed tabulation of all of the performance metrics for the Random Forest classifier is provided in Table III.

TABLE II COMPARISON OF THE TWO MODELLING APPROACHES

Model	Task	Key Strength	Limitation
Prophet	State-level sales forecasting	Captures trend, seasonality;robust to missing data/outliers	Accuracy bound by historical data quality
Random Forest	Vehicle-type classification	Ensemble reduces overfitting; handles categorical features well	Depends on quality of engineered features

TABLE III RANDOM FOREST CLASSIFIER — PER-CLASS PERFORMANCE METRICS (TEST SET, n=360)

Vehicle Class	Precision	Recall	F1-Score	Support
2-Wheeler	0.6519	0.6776	0.6645	152
3-Wheeler	0.6216	0.5055	0.5576	91
4-Wheeler	0.5781	0.6325	0.6041	117
Weighted Avg.	**0.6203**	**0.6194**	**0.6178**	**360**

Overall test accuracy: 61.94% (n=360, 80/20 train-test split, 200 estimators, max depth 15). Forecasting model: test MAPE 0.28%, RMSE 223 units, R^2 0.9957 (20% held-out test, 80 months).

C. Testing

The testing of this system occurred at various levels. Unit tests were performed on all Flask API endpoints to ensure proper responses were generated to both forecasting and classification requests. The Android application had a separate set of tests conducted, including testing authentication, navigation, and visualizations: authentication was checked via Firebase Authentication to ensure a secure login, Retrofit was evaluated for proper communication between the application and backend, and MPAndroidChart visualizations were evaluated for their clarity and accuracy in presenting forecasted data.

D. Discussion

The prototype provided quantifiable forecasting and classification capabilities, as demonstrated by a test MAPE of 0.28%, R^2 value of 0.9957 for the time series component of the product and Random Forest accuracy of 61.94% with balanced per-class F1 scores across all three types of vehicles. These results show that a specialised forecasting model that accounts for trends and seasonality, when paired with a robust ensemble classifier, will give consistent, data-driven insights to non-technical users via a mobile interface. The accuracy of the classifier reflects the ambiguity inherent with attempting to classify states with various levels of electric vehicle adoption. There will be more mixed EV adoption states than there will be EV-adopting states which results in classifiers outperforming chance based on the F1 score weighted by class. Given this information, future improvements may include additional socioeconomic variables being introduced,

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ※ Peer-reviewed & Refereed journal ※ Vol. 15, Issue 6, June 2026
DOI: 10.17148/IJARCCE.2026.156102

a larger historical record being utilised, and real-time data being incorporated into both models to improve the accuracy of each model.

VI. CONCLUSION AND FUTURE SCOPE

In this paper, we described the development of a dual-model machine learning approach to solve two important operational issues related to electric vehicle (EV) sales in India: predicting EV sales by state, and identifying the main type(s) of EVs expected to be sold in a particular region and timeframe. The dual machine learning models were deployed using access via an API to a Flask backend, and made accessible to non-programmer users through interactive dashboards and mobile applications secured using Firebase Authentication.

The study demonstrated that the machine learning forecasting model when tested resulted in a mean absolute percentage error (MAPE) of 0.28% and a high R^2 value of 0.9957 for EV monthly sales by state. The model captured long-term trend as well as monthly seasonal patterns in EV sales, and based on the outputs of the ETL processes used in each case, the machine learning classification model (Random Forest) produced an overall accuracy and F1 score of 61.94% and 0.6178, respectively, across three vehicle type categories (2 Wheelers, 3 Wheelers, 4 Wheelers). Both machine learning models outperformed a naive baseline and an alternative using Logistic Regression in terms of accuracy and F1 scores. The use of full-stack deployment of trained models demonstrates how machine learning approaches can be used to connect trained machine learning models to real-time, practical user-facing applications for businesses. The models created in this study are available for use as a proof-of-concept prototype, and their level of accuracy will depend on the quality and scope of the data set used to develop them; additionally, final decisions related to any potential business or government policies regarding use of EVs, require further interpretation by an expert. Future projects may also be developed around including real-time FAME India sales data, a greater variety of hybrid vehicle cover, scalable cloud-based back-end deployment, and incorporating additional socio-economic and infrastructure characteristics for even greater model accuracy.

ACKNOWLEDGMENT

The authors wish to express their gratitude for the continuous guidance, constructive criticism, and support of Professor Sudeshna Roy, who was highly instrumental in the success of this project, and to the faculty members at Bharati Vidyapeeth's Institute of Management and Information Technology in Navi Mumbai, who provided access to resources and support in completing this project.

REFERENCES

[1] S. J. Taylor and B. Letham, "Forecasting at scale," The American Statistician, vol. 72, no. 1, pp. 37–45, 2018.
[2] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[3] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[4] W. McKinney, "Data structures for statistical computing in Python," in Proc. 9th Python in Science Conf. (SciPy), 2010, pp. 56–61.
[5] Government of India, Ministry of Heavy Industries, "Scheme for Faster Adoption and Manufacturing of (Hybrid &) Electric Vehicles in India (FAME India)," 2021.
[6] NITI Aayog, "India's Electric Mobility Transformation: Progress and Future Roadmap," Govt. of India Report, 2021.
[7] Society of Manufacturers of Electric Vehicles (SMEV), "Electric Vehicle Sales Report, India," 2023.
[8] Pallets Projects, "Flask Web Framework Documentation," 2023. [Online].
[9] Google, "Firebase Authentication Documentation," 2023. [Online].
[10] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-driven documents," IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 12, pp. 2301–2309, 2011.

