Time Series Forecasting of User Engagement in Python Software Applications Based on Usage Logs

Janarthan V[1*], Erandi JDT[2], and Tharaka YMS[3]

[1]Department of Software Engineering, Faculty of Computing, Sabaragamuwa University of Sri Lanka
[1]Department of Data Science, Faculty of Computing, Sabaragamuwa University of Sri Lanka
[3]Department of Software Engineering, Faculty of Computing, Sabaragamuwa University of Sri Lanka
*janarthanvijayakumar@gmail.com

Abstract

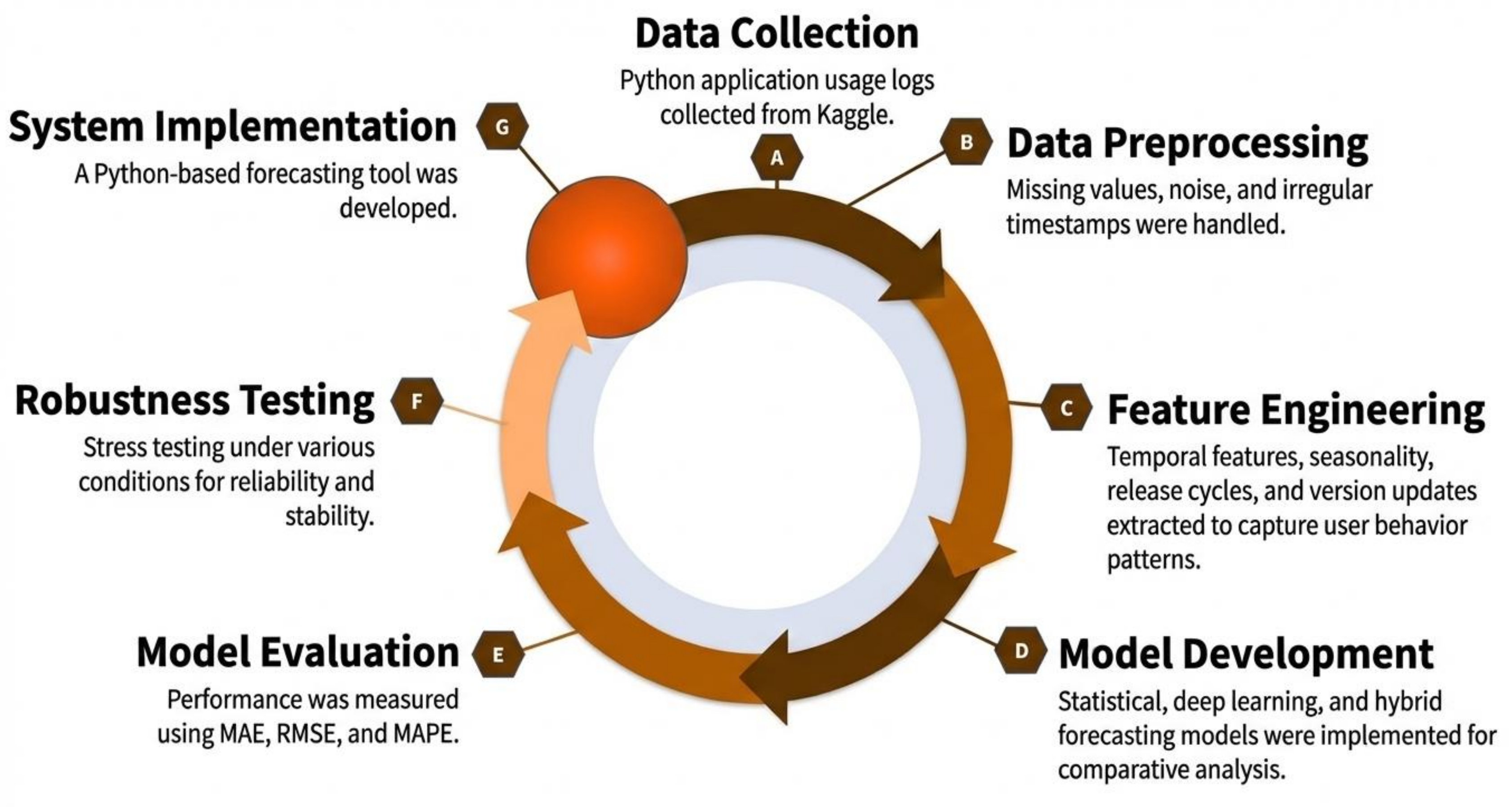
Understanding users’ engagement in Python-based software applications is important for improving usability, maintaining user satisfaction, and strategic future planning. Python usage logs contain important but rarely used variables like timestamps, functional interaction, session and download counts to analyze user engagement. However, existing traditional analysis mainly supports statistical analysis rather than capturing temporal dependencies, irregular usage patterns, and domain specific variables like feature updates or version releases and these studies mainly focused on single applications like Instagram, Facebook and TikTok but not considered as Python-based applications. Therefore, research addresses this gap by developing a time series forecasting framework with use of Python application usage logs. This research study analyzes and compares three types of forecasting model like traditional statistical models (ARIMA, SARIMA and Prophet), deep learning models (LSTM and GRU), and hybrid models (Prophet-LSTM, SARIMAX-GRU and Prophet-ElasticNet), guided by research questions assessing by evaluating their forecasting performance and testing under different use cases, such as sparse logs and bursty activity, to examine their robustness and practical suitability for Python application forecasting. The study started with literature reviewing of limits of existing forecasting models, limits of features and application types, and fewer studies on Python applications. Continuing with accurate methodology including data analysis preprocessing, feature engineering, models development, evaluation (MAE, RMSE, and MAPE) and testing and comparison of models. This study involves a Python native forecasting tool developed with Django, React, REST-API, TensorFlow, Prophet and Python libraries with developed model artifacts. Findings explain that hybrid models got high performance (MAPE are 6.35% and 8.53%) other than both statistical and deep learning models particularly under the sparse logs and bursty conditions. The research concludes that irregular features and hybrid residual prediction methods gave high forecast performance and offered valuable insights for Python developers and product teams.

Keywords : Engagement Forecasting, Prophet-LSTM, Python Software Applications, Time Series, Usage Logs

I. Introduction

- Software teams struggle to understand users' engagement in Python-based applications over time.
- Existing analytics focus on historical insights and social media platforms, not Python usage logs.
- Python logs often show irregular, sparse, and bursty behavior.
- This study compares statistical, deep learning, and hybrid models.
- The goal is to build an accurate and robust forecasting framework.

III. Methodology


Data Collection
A
Python application usage logs collected from Kaggle.

Data Preprocessing
B
Missing values, noise, and irregular timestamps were handled.

Feature Engineering
C
Temporal features, seasonality, release cycles, and version updates extracted to capture user behavior patterns.

Model Development
D
Statistical, deep learning, and hybrid forecasting models were implemented for comparative analysis.

Model Evaluation
E
Performance was measured using MAE, RMSE, and MAPE.

Robustness Testing
F
Stress testing under various conditions for reliability and stability.

System Implementation
G
A Python-based forecasting tool was developed.

Fig. 1. Methodology diagram

The diagram shows the study workflow, including data collection, data cleaning, feature extraction, model building, model testing, and system implementation.

V. Conclusion

This study demonstrates that hybrid forecasting models provide more accurate and reliable predictions for Python application usage logs. By capturing irregular and bursty usage patterns and incorporating contextual features such as version updates, hybrid approaches outperform traditional statistical and deep learning models. The proposed framework offers practical value for software teams to support planning, monitoring, and decision-making.

II. Literature Review

Table 1. Literature review table

Concept	Purpose	Relevance to this Study
ARIMA	Finds seasonality & linear trends	Base statistical forecasting for stable engagement (Zhang, 2003)
LSTM	Nonlinear & long-term dependencies modelling	Handle irregular, bursty, or sparse logs by deep learning modelling (Chen et al., 2023)
Prophet-LSTM	Models separate nonlinear log data from trends	For the complex patterns improve performance (Kong et al., 2021)
Action-graph Behavior modelling	Context actions are used in sequential manner	Justifies the contextual variable integrations and modelling (Liu et al., 2019)
Hybrid TSF Surveys	Hybrid modelling has high performance.	It Supports selection of the hybrid frameworks (Miller et al., 2024)

IV. Results and Discussion

Hybrid models achieved the best forecasting performance, with **MAPE values of 6.35% and 8.53%,** lower than those of statistical and deep learning models. Traditional models showed higher errors, especially with sparse and bursty logs. Adding contextual features such as version releases improved prediction accuracy, confirming that hybrid approaches are more reliable for Python application usage forecasting.

Table 2. Final model performance summary

Model	Prophet-LSTM	SARIMAX-GRU	ARIMA	Prophet-ElasticNet	GRU	LSTM	Prophet	SARIMA
Type	Hybrid	Hybrid	Statistical	Hybrid	Deep	Deep	Statistical	Statistical
MAE	192.66	236.68	226.32	1792.44	349.30	420.08	474.53	686.79
RMSE	281.42	298.63	318.04	3292.89	478.13	623.47	888.11	873.89
MAPE	6.35%	8.53%	12.34%	18.68%	32.96%	34.90%	37.40%	47.84%

References

[1] Chen, Z., Ma, M., Li, T., Wang, H., & Li, C. (2023). Long sequence time-series forecasting with deep learning: A survey. Inf. Fusion, 97, 101819.

[2] Kong, Y. H., Lim, K. Y., & Chin, W. Y. (2021). Time Series Forecasting Using a Hybrid Prophet and Long Short-Term Memory Model. Soft Computing in Data Science, Singapore.

Computing Undergraduate Research Symposium
(ComURS2026)
Faculty of Computing
Sabaragamuwa University of Sri Lanka