A Systems Engineering Approach

Johan S. Beltrán Merchán, Edison D. Álvarez Varela, Yader I. Quiroga Torres, Julián D. Celis Giraldo

Systems Engineering Department
Universidad Distrital Francisco José de Caldas

1. Introduction: The Systems Problem

Forecasting daily traffic for approximately **145,000 Wikipedia articles** is a large-scale Systems Engineering challenge characterized by:

- **Massive Scale:** Requires high computational efficiency (**Scalability**).
- **Chaos Factors:** High volatility due to viral or unpredictable events.
- **Heterogeneity:** No single model fits all series.

The architecture must be adaptive and robust to minimize the **SMAPE** metric.

2. Goal: Adaptive Forecasting Architecture

Research Question: How can a scalable and maintainable system architecture minimize SMAPE across heterogeneous time series using **Systems Engineering Principles**?

Expected Product: A **Modular Monolith** based on a **Hierarchical Ensemble** that dynamically selects the best forecasting model per article.

Performance Metric

Symmetric Mean Absolute Percentage Error (SMAPE):

$$SMAPE = \frac{100}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{(|A_t| + |F_t|)/2}$$

Where F_t is the forecasted value and A_t is the actual value at time t.

3. Proposed Solution: Architecture and Patterns

A **Modular Monolith** with clear separation of concerns, anchored by two design patterns:

System A: Data Flow Integrity

- **Pattern:** Chain of Responsibility.
- **Function:** Defines a linear 9-module pipeline (*Ingestion* $\rightarrow$ *Feedback*) ensuring traceability and data consistency.

System B: Adaptive Forecasting

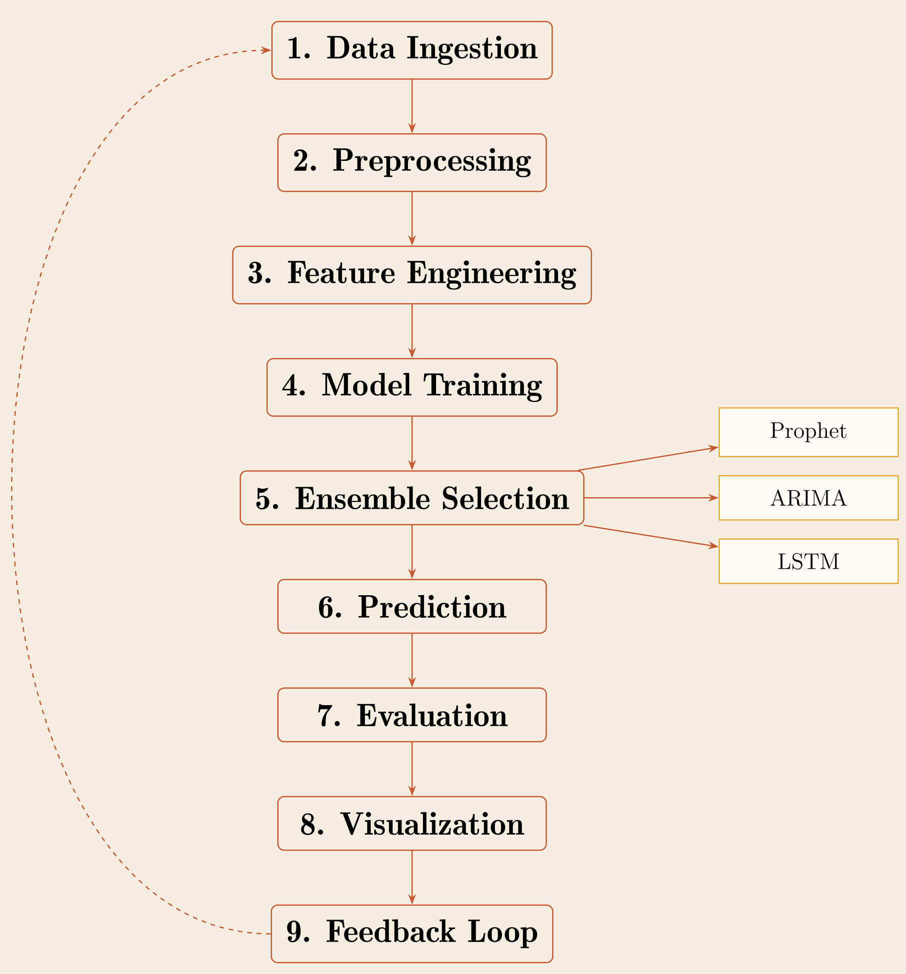
- **Principle:** Equifinality (multiple valid pathways to the goal).
- **Implementation:** Hierarchical Ensemble with the Strategy Pattern.

Hierarchical Ensemble Breakdown

- **Level 1 (Strategies):** Base models (**ARIMA**, **Prophet**, **LSTM**).
- **Level 2 (Meta-Model):** Analyzes metadata (*language*, *volatility*) and dynamically selects the optimal model.

Scalability: Achieved via parallel processing with `Joblib`, distributing models across multiple CPU cores.

Architectural Blueprint (Data Flow)



4. Validation and Testing Philosophy

Rigorous testing ensures robustness and maintainability:

- **Unit Tests:** Ensure deterministic behavior and prevent data leakage.
- **Integration Tests:** Validate the Chain of Responsibility pipeline.
- **Acceptance Testing:** Evaluate overall SMAPE performance.
- **Performance Tests:** Monitor computational efficiency and scalability limits.
- **Quality Assurance:** Continuous integration with automated testing ensures system reliability across updates.

5. Results Projected: Granular Analysis

The **Evaluation Module** produces a **Stratified Post-Prediction Analysis** feeding insights back into the system.

SMAPE Decomposition Categories

Category	Purpose
Traffic Level	Compare performance across page popularity levels.
Language	Identify bias in multilingual datasets.
Volatility Score	Correlate error with series variance.
Temporal Patterns	Assess weekly/seasonal trend accuracy.

Expected Benefits:

- Improved model selection for volatile series
- Reduced computational overhead through intelligent caching
- Enhanced interpretability via stratified metrics

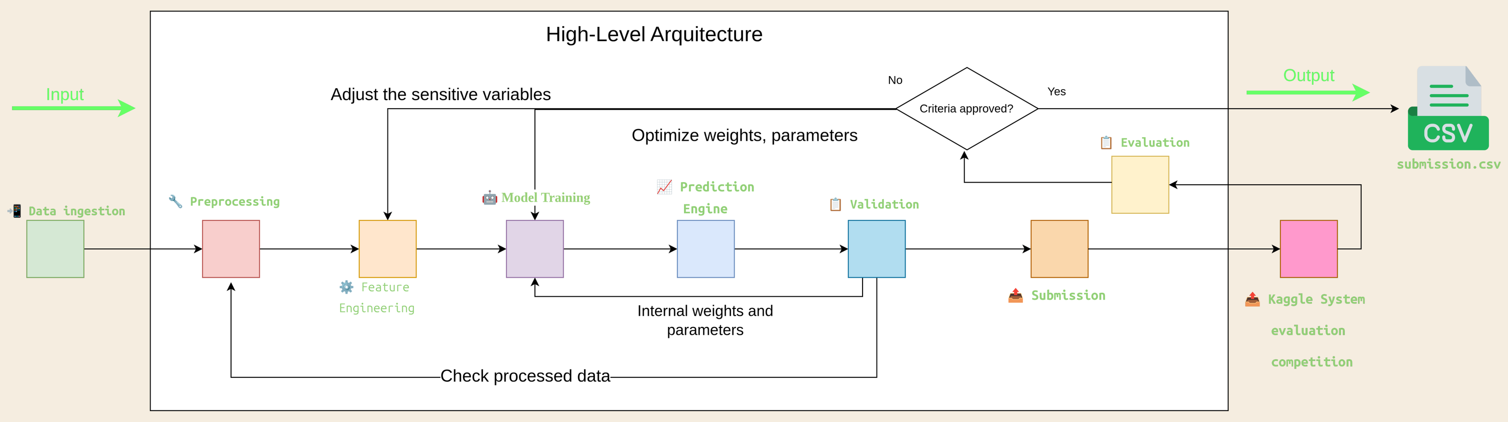
This analysis refines feedback loops for volatile or underperforming subgroups.

6. Conclusion and Future Work

The architecture provides a robust, scalable, and adaptive solution for chaotic web traffic forecasting.

Key Contributions:

- **Chain of Responsibility** ensures data integrity and traceability.
- **Hierarchical Ensemble** ensures adaptive model selection.
- **Systems Engineering principles** enable maintainability and extensibility.



Conceptual Architecture: Inputs, Core Processes, and Quality Loop

Future Work:

- Integrate advanced models (**Transformer Networks**, **XGBoost**, **Random Forests**) as new strategies.
- Implement real-time streaming architecture for live predictions.
- Develop automated hyperparameter optimization framework.
- Deploy containerized microservices for cloud-scale deployment.

Acknowledgments and References

The team thanks Professor Carlos Andrés Sierra Virgüez for his guidance throughout this research project.

References:

1. Shannon, C. E. (1948). "A Mathematical Theory of Communication," *Bell System Technical Journal*, 27(3), 379-423.
2. Trusov, A. (2017). "Kaggle Web Traffic Time Series Forecasting - 1st Place Solution." GitHub Repository. `https://github.com/Arturus/kaggle-web-traffic`
3. Taylor, S. J., & Letham, B. (2018). "Forecasting at Scale," *The American Statistician*, 72(1), 37-45.
4. Hochreiter, S., & Schmidhuber, J. (1997). "Long Short-Term Memory," *Neural Computation*, 9(8), 1735-1780.
5. Makridakis, S., et al. (2018). "Statistical and Machine Learning forecasting methods," *PLoS ONE*, 13(3).

Contact: jsbeltranm@udistrital.edu.co
jdcelisg@udistrital.edu.co
yiquirogat@udistrital.edu.co
edalvarezv@udistrital.edu.co