System design implementation for COVID-19 Global Forecasting System Kaggle Competition

Santiago Vargas Gomez - 2024202139
Dept. of Systems Engineering
Universidad Distrital Francisco José de Caldas

David Esteban Sanchez Torres - 20221020093
Dept. of Systems Engineering
Universidad Distrital Francisco José de Caldas

Dilan Guisseppe Triana Jimenez - 20221020100
Dept. of Systems Engineering
Universidad Distrital Francisco José de Caldas

Daniel Alejandro Castro - 20242020271
Dept. of Systems Engineering
Universidad Distrital Francisco José de Caldas

***Abstract*—Through this paper, the reader will comprehend the conceptual and technical design of a data-driven forecasting system developed for the analysis of a Kaggle competition about COVID-19 Global Forecasting. The development of the system follows systems engineering and systems design principles. The study integrates a modular architecture, feedback control, and sensitivity analysis to ensure reliability and adaptability under uncertain conditions caused by inconsistencies in the dataset. The analysis highlights the system's sensitivity to noisy and incomplete data, model variability, and chaotic behaviors arising from policy shifts and feedback effects. To mitigate these challenges, the design incorporates validation processes, monitoring routines, and retraining mechanisms that promote stability and continuous improvement. Via the results, it is demonstrated how conceptual modeling and technical implementation work together to build a robust, traceable, and adaptive forecasting system.**

***Index Terms*—Systems Design, Systems Engineering, Sensitivity Analysis, Chaos Theory, Forecasting, COVID-19, Machine Learning, Noise, Parameters, Feedback.**

I. Introduction

Recent works have demonstrated the importance of ensemble and probabilistic approaches for improving the accuracy of pandemic forecasting models. Cramer et al. [1] evaluated multiple individual and ensemble forecasts of COVID-19 mortality in the United States, providing a comprehensive benchmark for model comparison and uncertainty quantification. Similarly, Shinde et al. [2] presented a broad review of forecasting models, identifying the key challenges of data variability, model overfitting, and sensitivity to input noise. These studies highlight the necessity of system architectures capable of handling uncertainty and feedback, which directly influenced the design of this work.

The COVID-19 Global Forecasting competition, hosted on Kaggle, aimed to predict confirmed cases and fatalities worldwide using daily time-series data. The challenge represents a complex, dynamic, and nonlinear system due to data quality and its sensitivity, which is related not only to the reported cases but also to external variability factors intrinsic to the nature of the virus and its propagation, such as maximum environmental temperature, relative humidity, and wind speed reporting frequency. Furthermore, the consideration of human interventions has kept certain challenges that remain unaddressed afloat. These challenges can be divided into medic and non-medic factors that can affect the final forecast result due to the significant influence they have on the analyzed data; The tracking of victims before and after the infection, the quality of health services, and how they reported the information are truly difficult tasks to document and incorporate into the provided datasets due to the specificity of the tasks and their high-time dependence.

As can be seen, one of the central aspects of this study is the analysis and control of sensitivity and chaotic behavior within the forecasting process. Through this research, we can take into account key factors, such as those mentioned earlier, to comprehend how they contribute to instability by including incomplete or noisy data, overfitting, and hyperparameter sensitivity. The aforementioned aspects led us to search for proposals related to control and mitigation mechanisms capable of managing sensitivity and chaos, based on continuous monitoring, feedback loops, and adaptive retraining. These techniques align with systems control theory, ensuring that the forecasting process remains resilient under uncertain conditions. The overall objective is to demonstrate how systems engineering principles can be effectively applied to a real-world predictive analytics problem, yielding a framework that is both technically sound and conceptually coherent. The results highlight the importance of structured design, feedback-based adaptation, and iterative refinement as essential components of reliable and sustainable data-driven systems.

From a systems engineering standpoint, the methodology used combines a modeling focus for defining system boundaries, interrelations, and feedback loops—with technical implementation that involves data pipelines, predictive models, and adaptive control. This forecasting task can be understood as the modeling of a complex, dynamic, and nonlinear system. Data irregularities, changing testing policies, and government interventions introduced significant noise and variability, making the system highly sensitive to initial conditions. Small changes in the data—such as delayed updates from the information source, missing records, or reporting corrections are prone to produce substantial shifts in the predicted outcomes. These characteristics align with the concept of sensitive dependence on initial conditions described in chaos theory, emphasizing the need for control mechanisms and adaptive

feedback to maintain stability and reliability.

As mentioned before, the present work applies systems engineering and systems design principles to the development of a modular forecasting system. The proposed approach integrates both conceptual and technical design phases, enabling a clear separation between the high-level architecture of the system and its computational implementation. Conceptually, the system is represented as a pipeline of interconnected modules, each of which describes how the data is treated in order to adjust the model's specifications. The pipeline is distributed in the following way: data ingestion, preprocessing, modeling, evaluation, and feedback. Each module fulfills a specific function within the overall architecture, following the principles of modularity, scalability, and traceability. Technically, the system is implemented using Python and its data science ecosystem, including libraries such as Pandas, NumPy, Scikit-learn, and XGBoost, which collectively support efficient data manipulation, modeling, and validation.

II. Methods and Materials

A. Design choices

The design of the model for the COVID-19 Forecasting System was guided by key systems engineering principles like modularity, scalability, stability, and traceability. Each principle is applied to ensure that the system could handle uncertainty, data variability, and model drift in a controlled and transparent way. The architecture was divided into five interconnected modules—data ingestion, preprocessing, modeling, evaluation, and feedback—each responsible for a specific stage of the data transformation process. This modular structure allows individual components to be updated or replaced without compromising the overall system. Scalability was achieved by selecting lightweight and parallelizable machine learning algorithms, such as Random Forest and XGBoost, which perform efficiently on large datasets. Stability was enhanced through ensemble learning, combining the advantages of both algorithms, XGBoost for high predictive accuracy and Random Forest for robustness against overfitting. Traceability is implemented through structured data management, GitHub version control, and experiment logging. These practices will allow every configuration, parameter set, and model output to be reproducibly traced across system iterations. Finally, a feedback-based control mechanism was embedded to detect performance degradation or data drift, ensuring that retraining is triggered automatically when necessary. This adaptive design establishes a balance between predictive power and system reliability.

B. System Architecture

The high-level architecture follows the modular pipeline described above. The TikZ diagram below is a compact representation of the data flow and the feedback control loop. The modularity functionality is described by:

- **Data Ingestion:** This module manages the import of input datasets (`train.csv` and `test.csv`) from Kaggle or local sources. It verifies file structure, consistency, and accessibility, providing the foundation for the entire data pipeline.
- **Preprocessing:** Responsible for cleaning and transforming the data. Cleans missing or inconsistent data, standardizes dates, and creates derived temporal and spatial features (e.g., lag variables and rolling averages).
- **Modeling:** The analytical core of the system. In this stage, algorithms such as *Random Forest Regressor*, *Gradient Boosting*, or *XGBoost* are trained using preprocessed data. The models learn temporal and spatial patterns that enable the prediction of confirmed cases and fatalities.
- **Evaluation:** The evaluation module validates predictions against known data and competition metrics. Submissions are generated in the `submission.csv` format and uploaded to Kaggle for automatic scoring, ensuring consistency and comparability.
- **Feedback:** This module collects the evaluation results and identifies potential improvements in data handling, feature engineering, or model configuration. It closes the learning cycle by sending refined parameters and insights back to earlier modules.

This workflow transforms raw input data into validated predictions in a structured, iterative manner. The feedback loop acts as the system's self-correcting mechanism—detecting performance drift and reinitiating model retraining to ensure long-term stability. In systems design terms, this feedback control aligns with cybernetic principles: the model's outputs are continuously monitored, compared to expectations, and used to correct system behavior through adaptive learning. Figure 1 represents this structure, showing the linear data flow and the feedback mechanism that ensures continuous model improvement.

C. Algorithm pseudocode

Algorithm 1 COVID-19 Forecasting and Feedback Control Loop

Require: train.csv, test.csv, retraining threshold δ
Ensure: submission.csv with predicted ConfirmedCases and Fatalities

```
Load train and test datasets
Preprocess data: clean, normalize, and engineer features
Train model (Random Forest or XGBoost)
Generate predictions for test data
Evaluate results using RMSLE metric
if error ¿ δ then
    Retrain model with updated data
end if
Output submission.csv and update via Kaggle API
```

III. Results & Discussion

The design process led us to a modular and adaptive architecture that demonstrates clear alignment with systems engineering principles such as modularity, feedback control,

and maintainability. The system integrates five functional stages—data ingestion, preprocessing, feature engineering, modeling, and feedback validation—forming a closed-loop structure capable of continuous learning and self-correction. The final architecture successfully balanced complexity management and sensitivity control, two major challenges identified during the conceptual design. Table 1 summarizes the main design outcomes:

TABLE I
KEY DESIGN COMPONENTS AND THEIR CONTRIBUTION TO SYSTEM ROBUSTNESS AND MAINTAINABILITY

Design Element	**Objective**	**Contribution to System Robustness**
Modular pipeline architecture	Separate functions by stage	Reduces interdependence and simplifies updates
Ensemble modeling structure	Combine different learners	Balances sensitivity and accuracy
Feedback control loop	Automate model retraining	Maintains performance under data drift
Data standardization and normalization	Ensure uniform input quality	Minimizes noise and input variability
Version control (GitHub)	Track configurations and iterations	Guarantees reproducibility and traceability

To validate the integrity and consistency of the architecture, a system-level traceability diagram was constructed (Fig. 2). This diagram ensured that every subsystem and design module was directly linked to a functional requirement—such as data reliability, model interpretability, or feedback response.

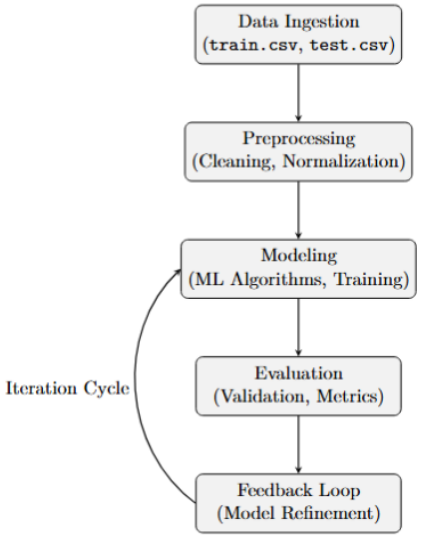

Fig. 1. Data Flow Between Modules of the COVID-19 Forecasting System.

With this validation, it is confirmed that the data flow remains consistent from ingestion to feedback. There were no dependency conflicts detected between modules, and changes in one stage could be performed independently without affecting downstream processes. This demonstrates the implementation of a low coupling and high cohesion design, which is a desirable property in systems engineering for adaptability and maintenance of any kind of system.

During the conceptual analysis, the system's sensitivity was identified as a potential weakness—particularly due to noisy or incomplete COVID-19 datasets. The final design addressed this by embedding monitoring and feedback mechanisms that stabilize outputs even when inputs fluctuate.

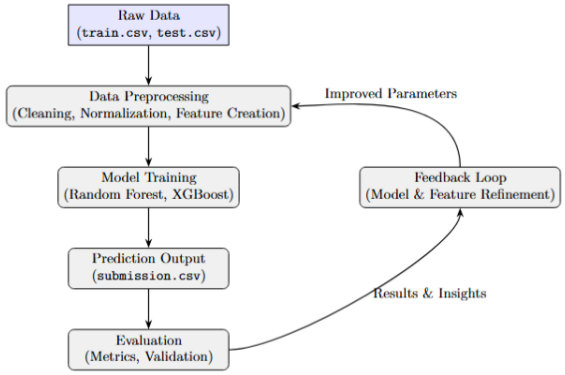

Fig. 2. System stability chart illustrating controlled sensitivity: the architecture mitigates abrupt changes in outputs under noisy or incomplete input data.

The design process highlights how systemic thinking can enhance the reliability of data-driven architectures. Unlike traditional machine learning pipelines that operate in isolation, this system integrates feedback, monitoring, and validation as structural components rather than afterthoughts. This approach transforms the forecasting model into a dynamic system capable of responding to uncertainty and evolving data conditions.

The integration of control and sensitivity principles also demonstrates that engineering design frameworks can effectively guide the development of intelligent forecasting systems. By systematically analyzing points of complexity—such as incomplete data, hyperparameter instability, and feedback latency—the architecture evolved into a resilient platform that maintains consistency under real-world variability.

Comparing the initial conceptual model and the implemented architecture reveals an improvement in clarity, modular interaction, and process feedback. The refinement stages transformed an abstract design into a practical implementation framework ready for reproducible experimentation.

IV. CONCLUSIONS

The design proposed for the COVID-19 Forecasting System successfully integrated systems engineering principles with data-driven modeling to design a modular, adaptive, and traceable architecture. The project's main achievement lies in transforming a static forecasting model into a dynamic system that incorporates feedback control, sensitivity management, and reproducibility. By structuring the system into independent modules—data ingestion, preprocessing, modeling, evaluation, and feedback—the design ensured stability and maintainability under real-world uncertainty and variations. The results

highlight the importance of systemic thinking for enhancing the abstract thinking in order to implement the robustness and transparency of analytical architectures.

Despite these advances, some limitations still remain. The current design depends on traditional machine learning models, which may struggle to capture long-term temporal dependencies or highly nonlinear behaviors in pandemic data. Additionally, sensitivity analysis and feedback are semi-automated, requiring human supervision for model retraining decisions. Future work will focus on extending the framework with deep learning architectures such as LSTM networks, integrating real-time interpretability tools (e.g., SHAP, LIME), and deploying continuous monitoring systems for fully automated feedback adaptation. These improvements will strengthen the system's resilience, scalability, and applicability to broader forecasting domains.

REFERENCES

[1] E. Y. Cramer, E. L. Ray, V. K. Lopez, ..., and N. G. Reich, "Evaluation of individual and ensemble probabilistic forecasts of covid-19 mortality in the united states," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 119, no. 15, p. e2113561119, 2022. [Online]. Available: https://www.pnas.org/doi/10.1073/pnas.2113561119

[2] G. R. Shinde, A. B. Kalamkar, P. N. Mahalle, N. Dey, J. Chaki, and A. E. Hassanien, "Forecasting models for coronavirus disease (covid-19): A survey of the state-of-the-art," *SN Computer Science*, vol. 1, no. 4, p. 197, 2020. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC7289234/