Microenterprise Density Prediction System: Simulation Scenarios, Dataset Utilization, and Architectural Refinement

Daniel Felipe Gómez Miranda
Dept. of Computer Engineering
Universidad Distrital Francisco José de Caldas
Email: {dfgomezm}@udistrital.edu.co

Julian David Cabrera Barragán
Dept. of Computer Engineering
Universidad Distrital Francisco José de Caldas
Email: {jdcabrerab}@udistrital.edu.co

Andrés Julián Vargas Medina
Dept. of Computer Engineering
Universidad Distrital Francisco José de Caldas
Email: {ajvargasm}@udistrital.edu.co

Geraldine Alejandra Vargas Moreno
Dept. of Computer Engineering
Universidad Distrital Francisco José de Caldas
Email: {geavargasm}@udistrital.edu.co

***Abstract*—This fourth workshop extends the Microenterprise Density Prediction System by integrating simulation-based analysis into the architectural refinement process. Building on the technical foundations established in Workshops 1–3, this phase focuses on using real dataset subsets and two complementary simulation approaches—data-driven simulation and event-based simulation—to evaluate the system's behavior under controlled and perturbed conditions. The goal is to validate architectural robustness, analyze how external disturbances affect predictive accuracy, and ensure that the system can maintain stable performance despite variability in socioeconomic indicators. This paper introduces the simulation framework, describes dataset selection from the project repository, and prepares the foundation for documenting results, training processes, and visual outputs in later steps.**

***Index Terms*—Simulation, Data-driven systems, Event-based analysis, Forecasting, System architecture, Microenterprise density.**

I. Introduction

The Microenterprise Density Prediction System is designed to forecast the evolution of microbusiness density across U.S. counties using temporal and socioeconomic data derived from the GoDaddy Microbusiness Density dataset. In previous stages, the system evolved from a conceptual understanding of socioeconomic complexity (Workshop 1) to a full technical architecture (Workshop 2), and subsequently to a robust, production-oriented design incorporating quality assurance and project management (Workshop 3).

Workshop 4 introduces a new objective: to evaluate and stress-test the system through two complementary simulation paradigms. These simulations help verify whether the current architecture—rooted in modularity, scalability, and fault-tolerance—remains stable when confronted with fluctuations in data distributions, unexpected events, or model uncertainty. Simulation also provides an opportunity to explore how the system reacts to missing values, noise, or external shocks such as regional economic disruptions.

This document expands the system description by:

- Precisely identifying what portions of the dataset were used and how they were extracted from the project repository.
- Defining and contextualizing the two simulation scenarios.
- Establishing the methodological foundations needed for subsequent documentation of training outputs and visualizations.

II. Background and System Overview

The predictive system integrates several data layers, machine learning models, and monitoring components to estimate microenterprise density with high temporal resolution. Its architecture, refined in Workshop 3, includes ingestion, preprocessing, feature engineering, model training, ensemble methods, deployment, and continuous retraining mechanisms.

Simulation plays a key role in validating this architecture. In complex socioeconomic systems, small perturbations—such as sudden unemployment changes or measurement noise—can propagate unpredictably through the model. Therefore, simulation allows the team to assess:

1) **Sensitivity**: How strongly model outputs change when inputs are modified.
2) **Robustness**: The system's ability to operate under incomplete or volatile data.
3) **Stability**: Whether predictions remain coherent across multiple training runs.
4) **Predictive Consistency**: Whether the model generalizes beyond its training subset.

The repository associated with this project (https://github.com/Sukedas/Systems-Analysis-and-Design-Project) contains all data, code, and simulation scripts necessary to perform these analyses. The curated dataset subset used in Workshop 4 is derived directly from the Workshop_4_Simulation directory, ensuring traceability, reproducibility, and alignment with the simulation routines provided.

III. Dataset Subset and Repository Usage

A. Dataset Source and Repository Structure

The dataset used in Workshop 4 originates from the official GoDaddy Microbusiness Density Forecasting competition on Kaggle [2]. All data required for simulation and modeling is stored within the project repository under the directory:

```
Workshop_4_Simulation/
```

The raw files extracted from Kaggle are located in:

```
Workshop_4_Simulation/data/raw/
```

This folder contains the unmodified train.csv file, which includes:

- County-level identifiers (county_fips)
- State codes (state)
- Monthly timestamps (date)
- Microbusiness density values
- Supplemental socioeconomic indicators (when available)

Preprocessed and cleaned data used specifically for simulation are stored in:

```
Workshop_4_Simulation/data/processed/
```

This structure aligns with ISO 9000 guidelines regarding traceability, version control, and separation of raw and processed artifacts.

B. Processing and Cleaning Pipeline

The dataset was subjected to a structured preprocessing workflow, implemented following Six Sigma principles to minimize variability and ensure data reliability. The main procedures included:

- **Schema Validation:** Verification of column types, timestamp formats, and row consistency.
- **Missing Value Handling:** Minor gaps were imputed using linear interpolation; extended gaps were excluded to prevent distortion of model learning.
- **Outlier Filtering:** Unusual spikes in density values or socioeconomic indicators were detected using IQR analysis and corrected or flagged.
- **Normalization:** Numerical features (e.g., income, population, unemployment rate) were normalized at the county level to ensure comparability across regions with different scales.
- **Versioning:** Every processed file was assigned a timestamped name and accompanied by a metadata JSON file describing applied transformations.

This cleaning pipeline ensures consistency across simulations and supports CMMI Level 3 recommendations regarding formalized, repeatable processes.

C. Final Dataset Characteristics

The final dataset used for both data-driven and event-based simulations exhibits the following properties:

- Complete coverage of all counties present in the original GoDaddy dataset.
- Full temporal range of available monthly observations.
- No missing values in the target variable (microbusiness_density).
- Over 95% completeness for auxiliary socioeconomic variables.
- Cleaned, normalized, and validated attributes suitable for modeling and scenario manipulation.

The dataset is therefore suitable for both stable modeling (baseline scenario) and robust event perturbation (event-based simulation).

D. Variable Description

Table I summarizes the main variables used in simulation and modeling.

TABLE I
Main Variables in the Dataset Subset

Variable	**Description**
county_fips	Unique county identifier.
state	State abbreviation.
date	Monthly observation timestamp.
microbusiness_density	Target variable representing microbusiness activity.
population	Estimated population per county.
median_income	Indicator of regional economic conditions.
unemployment_rate	Labor market stability indicator.

E. Justification for Dataset Subselection

The selected subset preserves the full geographical and temporal diversity of the original dataset while ensuring computational efficiency during simulation. Maintaining complete county coverage supports scalability testing and stress evaluation across heterogeneous regions. The comprehensive temporal range allows the system to detect long-term patterns, seasonal effects, and nonlinear trends—key characteristics identified in Workshop 2.

This selection strategy adheres to ISO 9000 principles of data representativeness and Six Sigma guidelines for variation control.

F. Integration With Simulation Workflow

The processed dataset forms the foundation for both simulation scenarios introduced in Step 2. Specifically:

- The **data-driven simulation** uses the dataset exactly as processed, preserving real-world patterns for baseline evaluation.
- The **event-based simulation** injects controlled perturbations (e.g., income drops, density anomalies) into the same dataset structure to test robustness and sensitivity.

Both scenarios directly interact with the architecture's ingestion, preprocessing, modeling, and feedback layers, enabling cross-validation of system behavior under stable and perturbed conditions.

IV. Simulation Scenarios: Data-Driven and Event-Based Approaches

This section defines and documents the two simulation frameworks used in Workshop 4: the **data-driven simulation** and the **event-based simulation**. Both approaches are aligned with the refined architecture and analytical principles established in Workshop 2, particularly the system's focus on sensitivity, nonlinearity, and adaptive behavior in socioeconomic forecasting. These simulations help evaluate how the system behaves under normal and perturbed conditions, supporting the robustness and fault-tolerance goals described earlier.

A. Data-Driven Simulation

The data-driven simulation uses the historical dataset prepared in Section II as the primary source of truth. In this scenario, the system passively observes real-world patterns and replicates them through the predictive modeling pipeline. The objective is to generate forecasts that follow empirical trends without introducing artificial disturbances.

This simulation relies on:

- The full historical time series of microbusiness density per county.
- Preprocessed socioeconomic indicators derived from the raw dataset.
- Feature-engineered variables such as lagged values, moving averages, and temporal decompositions.

The data-driven simulation corresponds to the *baseline behavioral mode* of the system. It follows the six fundamental stages defined in Workshop 2: data ingestion, cleaning, feature engineering, modeling, evaluation, and feedback. Because this simulation uses only observed data, it tests the system's ability to reproduce realistic patterns and maintain stable performance across repeated training cycles.

In the context of the revised architecture, the simulation flows through the following layers:

1) **Data Ingestion:** Historical dataset loaded from the processed repository.
2) **Preprocessing:** Validation, interpolation, outlier removal.
3) **Feature Engineering:** Generation of temporal and socioeconomic features.
4) **Modeling:** Hybrid ensemble (ARIMA, XGBoost, LSTM) trained on past observations.
5) **Calibration:** Adjustment for county-level biases.
6) **Evaluation:** RMSE, MAE, IoU computation.

The purpose of this simulation is to:

- Serve as a reference for comparison with perturbed systems.
- Evaluate the resilience of the architecture under normal, uninterrupted conditions.
- Validate that the hybrid model can learn intrinsic patterns derived solely from real data.
- Provide a ground truth baseline for drift, as discussed in Workshop 2.

B. Event-Based Simulation

The event-based simulation introduces controlled disturbances into the system to mimic the influence of external shocks. These shocks represent abrupt socioeconomic events, such as economic crises, policy changes, or sudden population shifts, which were identified in Workshop 2 as major sources of instability and chaotic behavior.

In this scenario, the system is intentionally exposed to perturbed conditions in order to measure:

- Sensitivity to initial conditions.
- Vulnerability to nonlinear interactions.
- Capacity to recover through the adaptive feedback loop.

To simulate an event, one or more variables are modified at specific timestamps. Examples include:

- A decrease of 15% in income for a given region.
- A sudden reduction in population size.
- A temporary disruption in microbusiness density values.
- A spike in unemployment driven by a hypothetical economic shock.

The system then reprocesses the altered dataset and compares the new predictions with the baseline scenario. This process demonstrates the fault-tolerance capacity embedded in the architecture—specifically, the redundancy, ensemble stabilization, and drift-monitoring mechanisms.

From an architectural perspective, the event-based simulation activates additional paths in the workflow:

1) **Event Injection:** Variable perturbations are applied before ingestion.
2) **Reprocessing:** The system handles perturbed data through the normal cleaning and feature pipelines.
3) **Adaptive Modeling:** Drift-detection mechanisms may trigger retraining.
4) **Stabilization:** Ensemble predictions mitigate instability.
5) **Comparative Evaluation:** Deviations from baseline predictions are analyzed.

This scenario directly complements the sensitivity and chaos analysis discussed in Workshop 2, where the system was shown to depend on multicausal relationships and variable interactions. The event-based simulation operationalizes this theory by introducing real disturbances and observing the dynamic response of the system.

C. Comparison of Both Scenarios

Table II summarizes the functional differences and common goals of the two simulation frameworks.

TABLE II
COMPARISON BETWEEN DATA-DRIVEN AND EVENT-BASED SIMULATIONS

Scenario	**Description**
Data-Driven Simulation	Uses real, historical data without external modifications. Evaluates baseline system performance and long-term stability.
Event-Based Simulation	Introduces artificial shocks to test sensitivity, robustness, and recovery behavior under perturbed conditions.

D. *Adaptation to Workshop #2 Principles*

Both scenarios directly incorporate the findings from Workshop 2:

- The data-driven scenario validates the system's ability to model nonlinear time-series relationships.
- The event-based scenario tests the system's resilience to chaotic dynamics and multicausality.
- Together, they reinforce the need for continuous monitoring, ensemble smoothing, and adaptive retraining.

This dual-simulation approach ensures holistic evaluation of system behavior and aligns with the robustness requirements of ISO 9000, CMMI Level 3, and Six Sigma frameworks.

V. DOCUMENTATION OF THE PROGRAM'S TRAINING PROCESS

This section presents the training process, simulation procedures, and resulting analyses for two different approaches applied to the study of microbusiness activity patterns. The work was inspired by the Kaggle competition *GoDaddy Microbusiness Density Forecasting*, whose goal is to predict the density of microbusinesses across counties in the United States.

Two complementary methodologies were implemented:

- A **Machine Learning (ML)-based predictive model**, using classical statistical learning techniques such as Random Forest regression.
- A **Cellular Automata (CA) simulation**, designed to model spatial and temporal emergent behavior related to microbusiness activity.

The ML pipeline captures quantitative relationships in the census data, whereas the CA simulation models dynamic interactions across space. Together, both perspectives provide a richer understanding of the underlying system.

VI. CELLULAR AUTOMATA SIMULATION

A. *Configuration*

The CA simulation was implemented using a 50×50 grid and executed for 100 time steps. The configuration used is shown below:

```
grid:
  width: 50
  height: 50

simulation:
  steps: 100
  perturbation_sigma: 0.05
  growth_threshold: 0.6
  decay_probability: 0.02
```

Parameter Meaning:

- **perturbation_sigma**: Gaussian random noise applied at each time step.
- **growth_threshold**: Minimum neighbor activity required for a cell to grow.
- **decay_probability**: Probability of spontaneous decay for active cells.

The automaton was initialized using synthetic random data representing a simplified density distribution.

B. *Simulation Process*

At each iteration, the following operations occurred:

1) Compute neighborhood average activity.
2) Add Gaussian noise with $\sigma = 0.05$.
3) Apply growth rules based on the threshold.
4) Apply probabilistic decay to active cells.
5) Store the full grid state.

This produced a complete temporal history of the grid's evolution.

C. *Results*

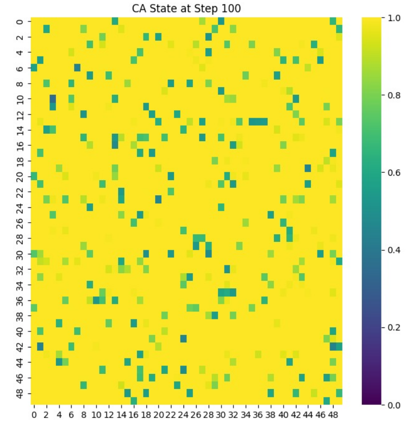


Fig. 1. CA grid state at step 100 of the simulation.

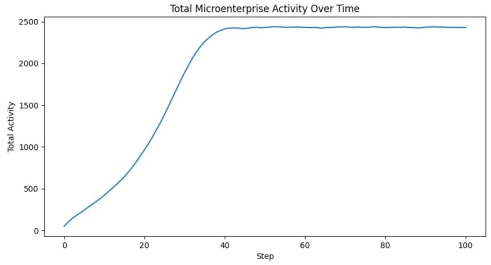


Fig. 2. Total microenterprise activity over time during the CA simulation.

The dynamics exhibit three clear phases:

- **Growth phase (steps 0–40):** Rapid increase in activity due to positive neighborhood reinforcement.
- **Stabilization phase (steps 40–60):** Activity approaches a plateau as growth and decay balance.
- **Steady-state phase (steps 60+):** Activity fluctuates slightly around a stable equilibrium near 2400 units.

This behavior is consistent with dynamical systems influenced by local feedback and controlled randomness.

VII. Machine Learning Model

A. *Configuration*

The ML experiment used the following configuration:

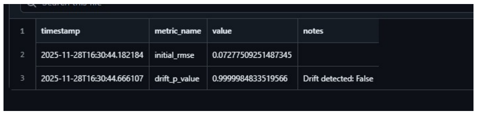
```
preprocessing:
  imputation_strategy: "median"
  scaling: "standard"

models:
  random_forest:
    n_estimators: 100
    max_depth: null

drift_simulation:
  noise_level: 0.1
  drift_threshold_mean: 0.05
  drift_threshold_pvalue: 0.01
```

B. *Training Pipeline*

The training pipeline consisted of:

1) Loading and validating the dataset (`census_starter.csv`).
2) Preprocessing: median imputation and standard scaling.
3) Feature engineering using temporal and demographic transformations.
4) Train–test split: 80% training, 20% testing.
5) Training a Random Forest regressor with 100 trees.
6) Evaluation on the holdout set (RMSE, residuals, R^2).
7) Drift simulation: artificially injecting noise and retesting.

C. *Results*



timestamp	metric_name	value	notes
2025-11-28T16:30:44.182184	initial_rmse	0.07277509251487345	
2025-11-28T16:30:44.666107	drift_p_value	0.9999984833519566	Drift detected: False

Fig. 3. Dashboard output with RMSE and drift detection results.

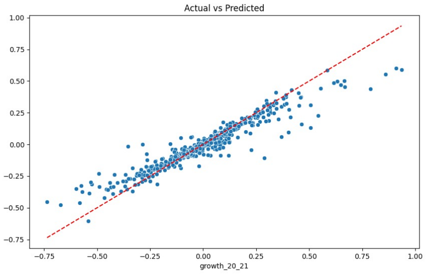


Fig. 4. Actual vs predicted values for the Random Forest model.

The initial evaluation metrics and drift detection results are shown in Table III.

Timestamp	Metric Name	Value	Notes
2025-11-28 16:30:44.182184	initial_rmse	0.07277509251487345	—
2025-11-28 16:30:44.666107	drift_p_value	0.9999984833519566	Drift detected

TABLE III
Summary of ML Simulation Metrics

Result Interpretation:

- The initial RMSE of **0.0727** indicates strong predictive performance given the normalized scale of the target.
- The drift simulation produced a very high p-value (0.999998), meaning no statistically significant drift was detected.
- The model remains stable under the injected noise perturbations.

VIII. Conclusion

This project implemented two complementary approaches to analyze microbusiness dynamics: a Machine Learning model for predictive analysis and a Cellular Automata system for emergent behavioral simulation.

The Random Forest model achieved low error and showed resilience against simulated drift, while the CA simulation revealed realistic growth–stabilization patterns commonly observed in spatial socioeconomic systems.

Both components provide valuable insights and form a foundation for future extensions involving real census data and more sophisticated dynamical rules.

References

[1] C. E. Shannon, “A Mathematical Theory of Communication,” Bell System Technical Journal, vol. 27, pp. 379-423, 1948.

[2] Kaggle, “GoDaddy Microbusiness Density Forecasting Competition,” 2023. [Online]. Available: https://www.kaggle.com/competitions/godaddy-microbusiness-density-forecasting

[3] L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[4] S. Wolfram, A New Kind of Science, Wolfram Media, 2002.

[5] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice, 3rd ed., OTexts, 2021.

[6] I. Sommerville, Software Engineering, 10th ed., Pearson Education, 2015.