Predicting Student Performance from Game Play Using a Systemic Perspective

Javier Alejandro Penagos Hernández, Henry Ricaurte Mora
Dept. of Computer Engineering
Universidad Distrital Francisco José de Caldas
Email: {japenagosh, hricaurtem}@udistrital.edu.co

Germán Darío Aya Fuentes
Dept. of Computer Engineering
Universidad Distrital Francisco José de Caldas
Email: gdayaf@udistrital.edu.co

***Abstract*—Game-based learning represents a dynamic and interactive approach to education. However, little attention has been given to collecting and analyzing data that can help assess student performance within these environments. In this paper, we analyze a Kaggle competition from a systems thinking perspective, aiming to predict student performance. The analysis begins by viewing the problem as a dynamic and complex system, leading to the design of a robust architecture for data organization and processing. We implemented and evaluated tree-based models including Random Forest and XGBoost, achieving average accuracies of 0.750 and 0.747 respectively across 18 questions in the educational gaming environment.**

I. Introduction

Game-based learning (GBL) has emerged as a powerful educational methodology that leverages interactive digital environments to promote engagement, critical thinking, and problem-solving skills among students [1]. Compared to traditional methods, GBL offers personalized and immersive experiences that enhance motivation and knowledge retention [2].

Despite its growing implementation in educational contexts, limited research has focused on the systematic collection and analysis of gameplay data to assess student performance. Such data are critical for understanding how learners interact with educational content and for guiding improvements in the design of these systems [3].

The Kaggle competition "Predict Student Performance from Game Play" [3] provides a valuable dataset that captures user behavior during interactive educational sessions. This paper addresses the problem from a systems-thinking perspective—treating the learning environment as a dynamic and complex system in which user interactions, system configurations, and contextual variables are interdependent.

Systems thinking facilitates the identification of emergent patterns and causal relationships that are not apparent through reductionist approaches. By conceptualizing the problem at a system level, we designed a modular data architecture and implemented predictive models that incorporate behavioral and contextual features.

Specifically, this study aims to predict whether a student will answer a question correctly based on their interaction history. To achieve this, we apply machine learning techniques—such as Random Forest and XGBoost—that have demonstrated high performance in structured, nonlinear data domains [4]–[7].

II. Method and Materials

A systems thinking approach was adopted to address the problem. The competition was first conceptualized as a system, in which key components, interrelationships, sensitivity factors, and implications of chaos theory were identified and analyzed.

A. System-Level Perspective

In the initial stage of the analysis, the game was treated as a system in which user interactions—specifically mouse events—serve as the primary inputs. These interactions are divided into two types: *click* and *hover*, both recorded with Cartesian coordinates (x, y) to capture the user's spatial behavior.

Subsequently, the main system components and their relationships were examined. Among these components are:

- The player's configuration settings (e.g., fullscreen mode, music, visual quality),
- The current level of the game,
- The question being answered.

There is a direct relationship between the number of correct answers and the player's level: the more correct responses, the higher the level. These elements are linked to a unique `user_id` that stores global configuration data.

Each time a user starts a session, a new `session_id` is generated. This, along with user interaction data, is managed by the `event handler` and stored in the `event registry`. The registry forwards the data to the `data analytics` module, which processes it and generates the system output: a prediction of the student's performance.

B. Sensitivity and Chaos Considerations

The system is sensitive to user input behavior, as interactions significantly influence performance predictions. Factors such as the type of input, session duration, and game configuration directly impact the output.

Moreover, the system is subject to random and unpredictable factors, aligned with chaos theory principles. While certain behavioral patterns may recur, user interactions within a given `session_id` are not guaranteed to be identical in another. External influences—such as the student's focus or prior knowledge—can further affect the accuracy of their responses and, therefore, the system's output.

C. System Overview Diagram

After identifying the system components, their interrelationships, and the influencing factors, a diagram was created to visually represent the system's structure and dynamics.

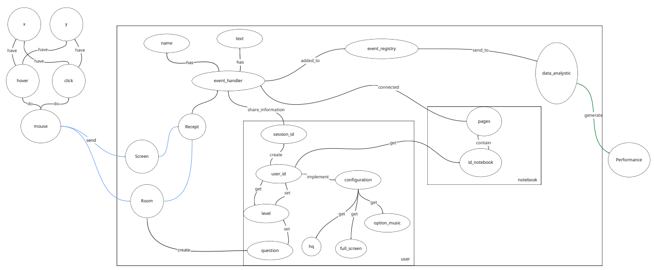

Fig. 1. System Diagram

D. Architecture of the Proposed System

The proposed architecture is structured into five interconnected modules, following principles of scalability and maintainability identified in the requirements phase:

1) Data Ingestion Module: Designed to capture events from the game environment. To accommodate the large volume of gameplay logs, data ingestion was performed with explicit type assignment to minimize memory usage. Numeric fields were downcast from default 64-bit types to smaller representations (e.g., 32-bit or 16-bit integers and 32-bit floats) whenever the observed value ranges allowed.

2) Preprocessing Module: Implements the normalization of coordinates, elimination of anomalous data, and extraction of temporal, spatial, and contextual features. During data preparation, erroneous or duplicate clicks occurring within a 90 ms window are filtered out, and spatial coordinates are normalized to a reference resolution.

3) Training Module: Where tree-based algorithms are configured. Feature extraction encompasses temporal metrics (inter-event intervals and response speed), spatial descriptors (movement trajectories), and contextual indicators (difficulty level and retry counts).

4) Prediction Module: Structures the real-time inference process with minimal latency to support near real-time data ingestion and processing while maintaining high availability and robustness.

5) Feedback Module: Designed to capture the results and constantly update the model, improving its precision over time through internal validation routines that ensure data integrity.

E. Dataset Characteristics and Feature Engineering

Three important types of features were identified: temporal, spatial, and contextual. These characteristics form the basis of the proposed predictive model, allowing the capture of different dimensions of student behavior during interaction with the educational environment.

A Python dictionary named `FEATURE_SETS` was defined to streamline feature engineering. Each entry comprises:

- A list of categorical columns for which frequency or uniqueness aggregations are computed (e.g., interface settings, event types).
- A list of numerical columns for which statistical measures (mean, standard deviation) are calculated (e.g., event durations, cursor coordinates).
- A brief descriptor indicating the thematic focus of the feature set (for instance, "configuration features," "text interaction features," or "screen behavior features").

F. Algorithm Selection

Three main algorithms were selected based on established criteria:

- **XGBoost**: As the main algorithm due to its balance between performance and efficiency
- **Random Forest**: Implemented through TensorFlow Decision Forests for its robustness to noise and interpretability

The selection prioritized tree-based classifiers for their resilience to outliers and partially chaotic user behavior, which aligns with the systems thinking approach adopted in this study.

G. Technical Stack and Implementation

The following tools were used to build and train the models:

- Python as main language
- NumPy and Pandas for data manipulation, cleaning, and processing
- XGBoost for creating tree-based models, highly efficient for structured data
- Matplotlib and Seaborn for visualization and exploratory data analysis
- TensorFlow Decision Forests for experimenting with advanced tree models within the TensorFlow environment

III. Results

To validate the proposed architecture under real conditions, simulations were conducted using the official Kaggle competition dataset. Both XGBoost and Random Forest models were trained on different sets of engineered features grouped by level and session identifiers.

The evaluation focused on 18 questions, with multiple distinct feature combinations tested per model to capture diverse behavioral patterns. Cross-validation techniques were employed, with accuracy chosen as the primary evaluation metric due to its interpretability in the educational context.

A. Overall Performance Summary

Table I presents the average accuracy achieved by each model across different feature sets. Random Forest achieved the highest average accuracy of 0.750 with Feature Set 1, while XGBoost achieved 0.747 with the same feature set. The minimal difference between models indicates that both are viable candidates depending on the deployment scenario.

TABLE I
AVERAGE ACCURACY BY MODEL AND FEATURE SET

Model and Feature Set	Average Accuracy
Random Forest (Set 1)	0.750
Random Forest (Set 2)	0.746
Random Forest (Set 3)	0.739
XGBoost (Set 1)	0.747
XGBoost (Set 2)	0.731
XGBoost (Set 3)	0.744

B. Performance Analysis by Question Complexity

The detailed analysis revealed significant performance variations across different questions, as shown in Table II for the best-performing configuration (Random Forest with Feature Set 1).

TABLE II
ACCURACY PER QUESTION - RANDOM FOREST (FEATURE SET 1)

Question	Accuracy	Question	Accuracy
1	0.736	10	0.525
2	0.983	11	0.616
3	0.933	12	0.887
4	0.831	13	0.701
5	0.559	14	0.718
6	0.718	15	0.627
7	0.684	16	0.768
8	0.644	17	0.644
9	0.712	18	0.955

C. Key Observations

Several important patterns emerged from the experimental results:

High Accuracy on Early Questions: Both models showed high accuracy on early-stage questions like Question 2 (accuracy 0.98), suggesting that decision trees perform well in capturing basic behavioral patterns at the start of the interaction.

Performance Variability in Mid- and Late-Stage Questions: Questions like 5, 10, and 15 showed lower accuracy (0.5-0.6), likely due to increased complexity and variability in user behavior as tasks became more complex or required strategic thinking.

Feature Set Impact: Feature Set 1 generally provided optimal performance for both models, while different feature engineering strategies showed varying effectiveness depending on the specific algorithm used.

Model Robustness: Random Forest demonstrated greater stability in the presence of noisy or less structured data, confirming its suitability for unpredictable educational environments where user interactions can be erratic.

D. Validation of the Proposed Architecture

The experimental results validate the proposed architecture against the established requirements:

- Successfully fulfills the functional requirements for capturing and processing interaction data with demonstrated accuracy above 0.7 on average
- Satisfies response time constraints through the selection of computationally efficient algorithms
- Implements a modular design that facilitates future scalability and maintenance
- Demonstrates robustness to noise and variability in user behavior patterns

The systems thinking approach proved effective in identifying key components and their interrelationships, leading to a comprehensive solution that addresses both technical and educational objectives.

IV. CONCLUSIONS

This work approached the problem from a perspective that differs from the conventional approach in data science by applying a systems thinking framework. This holistic perspective enabled us to conceptualize the phenomenon under study—student performance in game-based learning environments—as a complex, dynamic system composed of multiple interrelated components.

By gaining a deeper understanding of the system's internal dynamics, we identified key functional and non-functional requirements early in the process. This clarity facilitated the design of a scalable and modular data architecture aligned with the project's primary goal: predicting student performance based on behavioral patterns in educational games.

The experimental validation demonstrated the effectiveness of the proposed approach, with Random Forest achieving an average accuracy of 0.750 and XGBoost achieving 0.747 across 18 questions. The results show that tree-based models are well-suited for capturing the complex behavioral patterns in educational gaming environments, with particular strength in early-stage interactions and consistent performance across varying user behaviors.

The performance variations observed across different questions provide valuable insights into the complexity of student behavior in educational games. High accuracy on initial questions suggests that basic behavioral patterns are reliably captured, while the lower performance on mid-stage questions indicates areas where additional feature engineering or alternative modeling approaches might be beneficial.

The systems thinking approach not only enhanced the understanding of the problem space but also provided a robust methodological framework to guide the development of the predictive system, ensuring coherence between the problem analysis, system design, and project objectives. The modular architecture developed supports future enhancements and scalability, making it suitable for deployment in real educational environments.

Future work should focus on addressing the performance variations in complex questions through advanced feature engineering, exploring ensemble methods, and investigating the integration of sequential modeling approaches to better capture the temporal dynamics of student learning processes.

REFERENCIAS

[1] G. Team, "¿Qué es Game-based learning? - Gamelearn," Game Strategies, Jul. 23, 2014. https://gamestrategies.io/es/blog/que-es-game-based-learning/ (accessed May 16, 2025).

[2] J. Nisbet, "Game-Based learning: Pros, cons & implementation tips for educators," Prodigygame.com, 2022. https://www.prodigygame.com/main-en/blog/game-based-learning

[3] "Predict Student Performance from Game Play," @kaggle, 2025. https://www.kaggle.com/competitions/predict-student-performance-from-game-play

[4] "Gini Index and Entropy — 2 Ways to Measure Impurity in Data," Data Science Dojo, Feb. 07, 2025. https://datasciencedojo.com/blog/gini-index-and-entropy/ (accessed May 16, 2025).

[5] A. Saud and S. Shakya, "Analysis of Depth of Entropy and GINI Index Based Decision Trees for Predicting Diabetes," Indian Journal of Computer Science, vol. 6, no. 6, pp. 19–28, Jan. 2022, doi: 10.17010/ijcs/2021/v6/i6/167641.

[6] "XGBoost Documentation — xgboost 3.0.1 documentation," Readthedocs.io, 2025. https://xgboost.readthedocs.io/en/release_3.0.0/index.html

[7] "Titanic Tutorial," kaggle.com. https://www.kaggle.com/code/alexisbcook/titanic-tutorial