

Systems Analysis and Simulation for Problematic Internet Use

1st Bettsy Liliana Garces Buritica

code: 20231020222

blgarcesb@udistrital.edu.co

2nd Marta Isabel Sanchez Caita

code: 20222020118

maisanchezc@udistrital.edu.co

3rd Luis Fernando Rojas Rada

code: 20222020242

lfrojasr@udistrital.edu.co

Abstract—This work presents the design and implementation of a system for analysing and simulating problematic internet use (PIU) in children and adolescents, using the Child Mind Institute – Problematic Internet Use (CMI-PIU) Kaggle dataset. The project was developed in the context of the Systems Analysis & Design course and integrates four elements: data preparation, a data-driven predictive model (Scenario 1), an event-driven simulation based on a cellular automaton (Scenario 2), and a final web application that exposes the model to end users. Scenario 1 uses a tree-based classifier to predict the Severity Impairment Index (sii), while Scenario 2 simulates how sii might evolve in a synthetic population under different behavioural and social conditions. The results illustrate how classical machine learning and simple simulation techniques can be combined to support system thinking about PIU.

I. INTRODUCTION

The increasing availability of internet-enabled devices has raised concerns about problematic internet use (PIU) in children and adolescents. PIU is associated with sleep problems, attention issues, and other behavioural difficulties. In this project we study PIU from a systems perspective, combining data analysis, robust system design and simulation, following the structure of the course workshops.

We use the CMI-PIU Kaggle dataset, which provides clinical and behavioural measurements together with the Severity Impairment Index (sii), an ordinal label from 0 to 3 indicating the severity of impairment due to internet use. Our goal is twofold: (i) to build a predictive model for sii that can generate valid Kaggle submissions, and (ii) to design a simulation module that explores how sii might change over time in a population of adolescents.

II. DATASET AND PROBLEM DEFINITION

The dataset consists of tabular files (`train.csv`, `test.csv`, `data_dictionary.csv`) and additional time series data. For this project we focus on the tabular part. The `train.csv` file contains around 4000 records from approximately 3800 unique participants. The target variable is `sii`, which takes values in $\{0, 1, 2, 3\}$.

The data dictionary groups variables into several instruments: demographics, physical and fitness measurements, sleep disturbance scales, internet usage variables and the Internet Addiction Test (PCIAT). In some versions of the dataset, `sii` can be deterministically derived from the total PCIAT score using published thresholds; in our case `sii` is

already given and is used as the main label for supervised learning.

The Kaggle competition defines a quadratic weighted kappa (QWK) metric to evaluate predictions. In addition to the four-class scale, we are also interested in a binary view: *normal* (`sii` = 0) versus *problematic* (`sii` \geq 1).

III. SYSTEM ARCHITECTURE

The system architecture follows a modular design inspired by the course workshops on systems thinking, analysis, design and simulation. It comprises four main stages:

- 1) **Data ingestion and preparation:** load, clean and transform the CMI-PIU dataset.
- 2) **Scenario 1 – Data-driven prediction:** train a tree-based model to predict `sii`.
- 3) **Scenario 2 – Event-driven simulation:** simulate the evolution of `sii` in a cellular automaton.
- 4) **Application and monitoring:** expose the model through a web interface and record predictions.

Data ingestion uses the original CSV files provided by the competition. We remove identifier columns, drop the target from the feature matrix and handle missing values using simple filtering and imputation strategies. Categorical variables are encoded using the preprocessing pipeline and numerical variables are scaled when needed.

Scenario 1 and Scenario 2 share the same preprocessing module, implemented in Python. Both scenarios are connected through a common configuration file and can be executed independently. Finally, a small web application built with Flask loads the trained model and offers an interface for uploading CSV or Parquet files, running predictions and downloading the results.

IV. METHODS

A. Scenario 1: Data-Driven Prediction

Scenario 1 implements a classical machine-learning pipeline. After preprocessing, we train a gradient boosting model on decision trees (CatBoostClassifier) using the labelled portion of `train.csv`. The features include demographic variables, sleep-related scores and internet usage variables that are shared between the training and test sets.

The model is treated as a four-class classifier over `sii` $\in \{0, 1, 2, 3\}$. During evaluation we report both the standard classification metrics (accuracy, precision, recall and F1 score)

and the QWK metric used in the Kaggle leaderboard. For interpretability we compute feature importance scores, which indicate which variables contribute most to the prediction of higher sii levels.

In addition to the four-class output, we derive a binary risk indicator: normal ($sii = 0$) versus problematic ($sii \geq 1$). This mapping is used later in the web interface to present a simple yes/no answer while still keeping the underlying 0–3 score.

B. Scenario 2: Event-Driven Simulation

Scenario 2 models a synthetic population of adolescents using a two-dimensional cellular automaton. Each cell represents one individual and stores two pieces of information: the current sii level and a discretised internet-use category (low, medium, high, plus a category for missing values). The initial state of the grid is obtained by sampling rows from `train.csv` after preprocessing; sii values and internet-use indicators come directly from the dataset and are not invented.

The automaton uses a Moore neighbourhood with periodic boundary conditions. At each time step, the system computes a risk score for each cell based on three components: its current sii (normalised to $[0, 1]$), the risk associated with its internet-use level, and the proportion of neighbours with $sii \geq 2$. These elements are combined through weights that control the importance of individual behaviour versus neighbour influence. Small Gaussian noise is added to avoid perfectly deterministic trajectories.

If the final risk exceeds an upper threshold, the cell may move to a higher sii state (up to 3); if the risk is below a lower threshold, it may improve and move to a lower sii state (down to 0). Otherwise, the state remains unchanged. All these rules are implemented in a `step()` function, and the simulation loop is encapsulated in a `run_simulation()` function.

V. IMPLEMENTATION AND FINAL APPLICATION

Both scenarios are implemented in Python. The data preprocessing and model training code is organised into modules such as `preprocess.py`, `train_model.py` and `config.py`. The trained model is stored on disk and later loaded by the web application.

The final application is a small Flask server (`app.py`) combined with HTML, CSS and JavaScript files in the `templates` and `static` folders. Users can upload CSV or Parquet files with the same structure as `test.csv`. The server preprocesses the incoming data, runs the CatBoost model and returns a JSON object containing:

- the predicted sii value (0–3) for each record,
- a binary risk flag (0 = normal, 1 = problematic),
- estimated probabilities for the normal vs problematic classes,
- and a confidence score.

On the frontend, predictions are displayed in a table and summarised as counts of normal versus problematic cases. Users can also download a CSV file with the detailed predictions. This design provides a simple “connector” between

the system architecture and the trained model, as requested in the course project.

VI. SIMULATION AND TESTING

For Scenario 2, the simulation is executed through the `run_simulation()` function, which iterates the cellular automaton for a fixed number of steps. At each step, we record the number of cells in each sii state and save the results into a `scenario2_history.csv` file. These histories are used to generate plots showing how the distribution of sii levels evolves over time.

We run several experiments by changing the neighbour-influence weight and the risk thresholds. When neighbour influence is strong, high-sii states tend to form clusters and spread more rapidly. When individual risk is dominant and internet use is mostly low or medium, the population tends to stabilise at sii levels 0 or 1. These experiments do not claim clinical validity but demonstrate that the simulation behaves coherently with the assumptions encoded in the rules.

For Scenario 1, we evaluate the model using a validation split of the labelled data. The final configuration is selected based on QWK and F1 score. We also submit at least one prediction file to the Kaggle competition to verify that the submission format is correct and to compare our model with the public leaderboard.

VII. RESULTS AND DISCUSSION

The trained CatBoost model achieves acceptable performance on the validation set, with QWK and F1 scores indicating that the model can distinguish between normal and problematic cases, although the most severe classes remain more difficult to predict. Feature importance analysis highlights variables related to internet usage and sleep quality as key factors.

The simulation results complement these findings by illustrating possible trajectories of sii in a population. Under certain parameter settings, high-risk states propagate quickly through the grid, while under more favourable conditions the system shows a tendency to recover. From a systems analysis perspective, this dual view (prediction plus simulation) helps to reason about both static risk estimation and dynamic behaviour.

VIII. CONCLUSIONS AND FUTURE WORK

This project shows how concepts from systems analysis and design can be combined with machine learning and simulation to study problematic internet use in adolescents. Scenario 1 provides a data-driven model that can be directly evaluated in the Kaggle competition, while Scenario 2 offers a simple but expressive simulation of how impairment might evolve under different conditions.

Future work could include using more advanced models (such as neural networks or ordinal regression), incorporating time-series features from actigraphy data, and calibrating the simulation rules with expert knowledge. Another extension would be to connect the web application with an online database and use it as a decision-support tool in real scenarios.

ACKNOWLEDGMENT

This project was developed as part of the Systems Analysis & Design course. We thank the instructor for providing the project guidelines and the Child Mind Institute for releasing the dataset used in this work.

REFERENCES

- [1] Child Mind Institute, "Child Mind Institute Problematic Internet Use Dataset," Kaggle, 2024.
- [2] Course notes and slides from the Systems Analysis & Design class.