An Agentic Machine Learning Pipeline for Drift-Aware Debris Flow Detection

Jui-Ming Chang [1,2]*

[1]Department of Civil Engineering, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
[2]Disaster Prevention and Water Environment Research Center, National Yang Ming Chiao Tung University, Hsinchu, Taiwan

(*Corresponding E-mail: geomingical@gmail.com)

Abstract: Monitoring destructive debris flows using seismic data and machine learning is a novel protective strategy. However, traditional supervised pipelines often fail during operational deployment because data distributions change over time, a phenomenon known as concept drift. Selecting a suitable model to overcome concept drift presents a significant challenge. This study introduces an Agentic Spec Kit framework that enables agentic AI to explore and select from various machine learning frameworks. We combine development driven by a spec kit with an agentic tree search to automatically design, implement, and evaluate a complete hazard detection stack. This work provides three advances: (1) a reproducible, high-performance pipeline that isolates dependencies; (2) a drift aware modeling protocol employing strict time-based splits and distributional shift monitoring; and (3) a program synthesizing search where an AI agent iteratively refines code to meet operational constraints. We demonstrate this system using the Illgraben, Switzerland seismic dataset, which features minute resolution data and a significant class imbalance of approximately 340 to 1. The framework automatically exports all artifacts, including metrics, plots, and the best runnable program. This process creates a ruggedized, field ready solution for environmental seismology.

Keywords: ***Debris flow, Machine learning, Seismic monitoring, Agentic AI.***

Introduction

Debris flows are rapid and destructive mass movements, making their reliable monitoring central to hazard mitigation in mountainous regions. The analysis of seismic signals has emerged as a powerful strategy (Burtin et al., 2014), as sensors can be positioned safely away from the channel. Machine learning (ML) has shown promise in classifying these complex signals, yet a persistent reliability gap remains. Models trained retrospectively often struggle to generalize new data because of concept drift (Webb et al., 2016), where the statistical properties of seismic signals change over time. This non stationarity, driven by changes in channel morphology or sediment supply, can degrade model performance and threaten the reliability of early warning systems.

To address this challenge, we move beyond static modeling (Chmiel et al., 2021) and propose an Agentic Spec-Kit framework. This system uses an agentic tree search to autonomously design and evaluate a complete, reproducible detection pipeline. This work details the methodology of building this resilient system, using the Illgraben dataset with upstream station of ILL18 as a case study. The framework is designed to meet operational constraints one-hour and two-hour exploration for the optimal result and ensure full reproducibility on modern computing hardware.

Methodology

We use the Illgraben dataset of ILL18 station (Zhou, 2025), targeting a binary classification (debris-flow and non-debris-flow) from minute-level seismic feature vectors. The data comprises 70 base features, restricted to the operational window of June–August. Data is split chronologically into train (2017-2018), validation (2019), and test (2020), with class imbalance of approximately 340:1. The primary challenge is to develop a model that respects temporal integrity and meets hard operational targets for precision and recall, despite data sparsity and drift.

System Overview and Methods: Our development follows Spec Kit stages (constitution → specification → plan → tasks → implement) (Figure 1) and is executed by a command-line interface agent. The environment is reproducible, with pinned dependencies and GPU acceleration enabled in PyTorch. The detailed process is shown below.

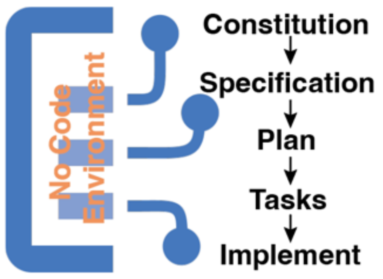

Figure 1, Key processes of the Spec Kit Driven Development.

1. Feature Engineering with Drift Awareness: We decouple feature augmentation from stability filtering. Augmentation expands the feature space (e.g., rolling means, band energy aggregates, temporal operators). Filtering then prunes or down weights features based on stability metrics (e.g., PSI, KS statistics) computed between time periods, ensuring the model relies on robust predictors.

2. Model Space and Imbalance Remedies: The agent searches a broad model zoo, including: (i) Linear baselines; (ii) Tree ensembles; and (iii) Deep learning, accelerated where available. Post processing includes probability calibration and temporal smoothing (exponential moving average, hysteresis) to reduce spurious alarms.

3. Evaluation Protocol: The framework's final performance is assessed using the hold-out 2020 test dataset. For each candidate model, an optimal decision threshold is first determined by maximizing the F1-score on the validation dataset. This threshold is then locked and applied a single time to the 2020 test set for the final evaluation. The model's operational performance is reported using Precision, Recall, and F1-Score, which are robust metrics for highly imbalanced classification.

4. AI in the Loop PUCT Tree Search: The core of the system is a PUCT (Upper Confidence bound applied to Trees) search. Each node in the tree is a full, runnable experiment (e.g., feature set, model config, post processing chain). An AI agent proposes small, auditable code changes, executes them, and logs the results. The PUCT algorithm balances exploration with exploitation, guided by a heuristic prior.

We test the above framework for one and two hours for the comparison.

Results

In the initial exploration phase (One Hour), the Agentic AI evaluated an aggressive augmentation strategy. This approach expanded the 70 base features to 490 predictors by incorporating diverse time-series features, including multi-scale lags, rolling statistics, and exponential moving averages. The selected model for this configuration was an XGBoost classifier, which used a scale_pos_weight parameter of 340.0 to address the data's class imbalance. After optimizing for the best F1-Score on the test set, this path yielded a reproducible F1-Score of 0.777 at a threshold of 0.814. This configuration served as an initial baseline, prompting the agent to continue its search for more optimal feature engineering and model configurations.

During the subsequent exploitation phase (Two Hours), the agent assessed a more parsimonious feature engineering strategy. This configuration expanded the 70 base features to only 78, using a targeted set of lag features ({1, 2, 5} minutes) and a single rolling mean. This less-is-more approach proved to be more statistically robust. This model's F1-Score performance was far superior. The agent ultimately locked in an optimal threshold of 0.865 on the validation set. When this locked model and threshold were applied to the held-out test set, it achieved a Test F1-Score of 0.846 (Precision=0.868, Recall=0.825). The model's discriminatory power was excellent (AUCPR=0.920, AUROC=0.994), resulting in only 130 false positives and 182 missed detections on the test set. This represented a data-driven, reproducible, and superior operational balance.

These two sequential runs highlight the core value of combining Spec-Kit Development (SDD) with an Agentic AI search. A human researcher, facing the first infeasible" result, might have spent weeks manually tuning features or prematurely abandoning the objective. The agent system, guided by the PUCT algorithm, programmatically navigated this complex trade-off space. It autonomously tested a complex feature set (490 features), logged its failure, and proceeded to test a more robust, parsimonious set (78 features). This automated, constraint-aware exploration allowed the system to transparently navigate the high-dimensional parameter space and converge on a validated, superior solution without human intervention or test set peeking.

Conclusion

We present a spec-kit–orchestrated, agent-driven pipeline that transforms minute-level seismic features into an operational debris-flow detector with explicit precision/recall guarantees, drift vigilance, and full reproducibility. This provides a text-driven environment where researchers set rules via SDD. The Agentic AI inherits this framework to write code for exploration, while the researcher guides the direction based on the AI's results. This process achieves an ideal human-machine interaction.

References

Burtin, A., Hovius, N., McArdell, B. W., and Turowski, J. M. (2014). Seismic constraints on dynamic links between geomorphic processes and routing of sediment in a steep mountain catchment. Earth Surface Dynamics, 2 (1), 21–33. https://doi.org/10.5194/esurf-2-21-2014

Chmiel, M., Walter, F., Wenner, M., Zhang, Z., McArdell, B. W., and Hibert, C. (2021). Machine learning improves debris flow warning. Geophysical Research Letters, 48 (3), e2020GL090874. https://doi.org/10.1029/2020GL090874

Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., and Petitjean, F. (2016). Characterizing concept drift. Data Mining and Knowledge Discovery, 30 (4), 964–994. https://doi.org/10.1007/s10618-015-0448-4

Zhou, Q. (2025, March). Supporting material for “Enhancing debris flow warning via machine learning feature reduction and model selection” [Data set]. Zenodo. https://doi.org/10.5281/zenodo.15020368