13th International Conference on Applied Informatics
Eszterházy Károly Catholic University
Eger, Hungary, February 19–21, 2026

Data cleaning and spam filtering methods in the TAWOS database*

Márk Szabó[a], **Ádám Kovács**[ab], **Gábor Kusper**[a]

[a]Eszterházy Károly Catholic University
{szabo.mark,kovacs2.adam,kusper.gabor}@uni-eszterhazy.hu

[b]University of Debrecen, Doctoral School of Informatics

Abstract

The simulation of IT projects plays an important role in developing students' practical skills, but the solutions currently in use only partially reflect real-world software development workflows, such as the continuous evolution of customer requirements or complex decision-making situations. The aim of this research is to improve the quality of project related data by focusing on data cleaning and the identification and removal of spam and non informative records from the TAWOS database. By ensuring higher quality and more reliable data, this work establishes the necessary foundation for future deep learning based modeling and simulation of project level events.

Improving data quality is a fundamental step. Ensuring that data is reliable, interpretable, and machine-processable is a prerequisite for any research that relies on software development event data, as high quality inputs are essential for meaningful analysis, modeling, and learning. The TAWOS database [3], which serves as the basis of this research, contains more than half a million events from 44 different open-source projects. It also includes a substantial amount of misleading, irrelevant, or noisy records that may distort downstream analyses if left unfiltered.

During preprocessing, we first removed tables and attributes that were irrelevant to the research objectives, thereby reducing redundancy and computational overhead. We implemented a rule-based, multi-criteria scoring system to detect

*This work was supported by the Slovak–Hungarian TéT project entitled "Realistic Project Simulation and Intelligent Product Owner Assistant for Improving Software Engineering Education" (Project ID: 2024-1.2.5-TÉT-2024-00072). This work was also supported by the Slovak Research and Development Agency under the Contract no. SK-HU-24-0037.

low quality, spam like, or non informative records. The rules and heuristics were chosen based on a thorough review of numerous spam and low quality records, allowing us to identify the patterns that most reliably indicate poor content. The classifier evaluates a combination of textual structure, stylistic signals, and domain-specific heuristics, including title and description length, word counts, character distributions, formatting patterns, and the presence of technical markers such as code blocks, stack traces, version numbers, file paths, or configuration references. Additional checks target gibberish, placeholder content, spam phrases, and unprofessional language. Individual subscores are aggregated and normalized to a 0-100 range. The overall quality score reflects relevance, clarity, and technical usefulness.

Typical examples of non informative records include automatically generated bot messages and issues containing only boilerplate or template text. Each entry receives a validity score from the mentioned scoring system, and for benchmarking purposes, records falling below a predefined threshold are classified as spam.

Validating such heuristic filtering mechanisms typically necessitates labeled datasets. A typical approach to obtain such datasets is manually labeling the records [2], which can be time consuming. To overcome this limitation and strictly evaluate our approach, we adopted an automated ground-truth generation strategy inspired by prior work [1] using GitHub archive data to get issues that were locked by moderators with the reason ”spam”. Using this reference dataset, we benchmarked our proposed scoring system against locally deployable transformer-based Large Language Models (LLMs) [4]. This comparative analysis allows us to verify whether our lightweight metrics can approximate the semantic filtering capabilities of transformer based LLMs. The evaluated performances are presented in Table 1. Multiple tresholds were tested for each method and the best tresholds were selected based on F1-score.

Table 1. Performance evaluated on GitHub Archive–derived labeled data (680 issues, 220 spam, 460 non-spam)

Method	**Accuracy**	**F1-score**	**Time (s)**
Own Metrics	91.2	86.0	0.25
Mistral-7B-Instruct-V0.3	90.4	84.1	514.09
Gemma-3-4B	88.1	82.3	301.22
Phi-3.5-Mini-Instruct	86.2	78.4	270.16
Meta-Llama-3.1-8B-Instruct	76.6	44.1	330.68

Notably, our scoring system achieves high overall accuracy and F1-score among all evaluated methods, outperforming or matching small transformer-based LLMs while requiring only a fraction of their computational resources. These results demonstrate that the proposed metrics reliably capture spam and non-informative patterns in issue content.

Applying the complete filtering pipeline based exclusively on the proposed scoring system to the full TAWOS dataset resulted in the identification of **6.8%** of all records, corresponding to approximately **31.500** events, as spam or non informa-

tive content. Due to their high computational cost, LLM-based filtering methods were evaluated only on a limited subset of the TAWOS dataset. The removal of low quality entries significantly improves the signal-to-noise ratio of the data and provides a reliable foundation for subsequent analytical or simulation-based systems that rely on this dataset.

Moreover, the resulting high quality dataset enables the extraction of statistically meaningful properties of software development processes, including distributions of event frequencies, communication delays, and issue resolution patterns. These distributions make it possible to generate realistic project event streams for simulation frameworks, ensuring that simulated project behavior preserves the dynamics of real-world development rather than relying on artificial or manually tuned assumptions.

References

[1] R. Ehsani, M. M. Imran, R. Zita, K. Damevski, P. Chatterjee: *Incivility in Open Source Projects: A Comprehensive Annotated Dataset of Locked GitHub Issue Threads*, in: Proceedings of the 21st IEEE/ACM International Conference on Mining Software Repositories (MSR 2024), Data and Tool Showcase Track, Lisbon, Portugal: Association for Computing Machinery, 2024, pp. 515–519, doi: 10.1145/3643991.3644887.

[2] D. Firake, B. Wakode: *Machine Learning-Based Spam Filter for GitHub Repository Issues*, in: Indian Journal Of Technical Education Volume 48 Issue 1, New Delhi, India: Indian Society For Technical Education, 2025, pp. 249–256.

[3] V. Tawosi, A. Al-Subaihin, R. Moussa, F. Sarro: *A Versatile Dataset of Agile Open Source Software Projects*, in: Proceedings of the 19th International Conference on Mining Software Repositories (MSR '22), Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2022, pp. 707–711, doi: 10.1145/3524842.3528029.

[4] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, C. M. Sam Shleifer Patrick von Platen, J. P. Yacine Jernite, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush: *Transformers: State-of-the-Art Natural Language Processing*, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, 2020, pp. 38–45, doi: 10.18653/v1/2020.emnlp-demos.6.