Poster: CIRRUS – Data-Driven Preprocessing Pipelines for Eye-Tracking Data

Jennifer Landes
University of Regensburg
Germany
jennifer.landes@informatik.uni-regensburg.de

Meike Klettke
University of Regensburg
Germany
meike.klettke@informatik.uni-regensburg.de

Oliver Ring
University of Regensburg
Germany
oliver.ring@stud.uni-regensburg.de

ACM Reference Format:
Jennifer Landes, Meike Klettke, and Oliver Ring. 2026. Poster: CIRRUS – Data-Driven Preprocessing Pipelines for Eye-Tracking Data. In *The 20th ACM International Conference on Distributed and Event-based Systems (DEBS '26), June 23–26, 2026, Lisbon, Portugal.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3809481.3816717

1 Motivation

Eye-tracking data is widely used to study human attention and behavior, yet raw gaze and pupil signals are noisy, incomplete, and highly variable [2, 3]. Preprocessing is therefore critical, but current workflows often lack standardization and rely on ad-hoc decisions. Small choices—such as imputation, outlier handling, or normalization—can significantly affect downstream analysis and model performance. Existing tools provide isolated functions but little guidance on how to combine them into coherent pipelines. These challenges become particularly relevant when preprocessing decisions must be applied consistently across multiple datasets or repeated analysis runs. CIRRUS therefore focuses on transparent offline preprocessing and reproducible pipeline selection.

2 Core Idea

We present **CIRRUS**, an interactive system that transforms preprocessing into a structured decision process. The key idea is to model preprocessing as a mapping from data quality characteristics to suitable methods. CIRRUS profiles each dataset and applies a transparent rule-based decision logic to recommend candidate preprocessing pipelines. The goal is not to replace expert judgment or to claim a universally optimal preprocessing strategy, but to make preprocessing decisions explicit, inspectable, and reproducible. The system combines (i) automated data profiling, (ii) rule-based method selection, and (iii) interactive visualization of preprocessing effects.

3 Contribution

This work makes the following contributions: (i) A transparent rule-based decision model that formalizes preprocessing recommendations based on data quality metrics, (ii) An interactive system that integrates profiling, recommendation, and execution into a unified pipeline, (iii) A visual and explainable framework enabling comparison of preprocessing strategies, (iv) A structured approach toward reproducible preprocessing across repeated analysis workflows.

DEBS '26, Lisbon, Portugal

ACM ISBN 979-8-4007-2693-4/2026/06
https://doi.org/10.1145/3809481.3816717

Table 1: Rule-based preprocessing decisions in CIRRUS based on data quality metrics.

Condition	Metric Range	Recommended Methods
Missingness (m)	$m < 5\%$	LOCF, Mean
	$5\% \leq m \leq 10\%$	KNN
	$10\% < m \leq 20\%$	KNN, MICE
	$m > 20\%$	MICE (flag for review)
Outliers (c, s)	$c < 5\%, \lvert s \rvert < 0.5$	Z-Score
	$c \leq 15\%$	IQR, MAD
	$c > 15\%$	Isolation Forest
Normalization	$\lvert s \rvert < 0.5$, low c	Z-Score
	$\lvert s \rvert > 1$ or high c	Robust Scaling
	otherwise	Min-Max

4 Approach

CIRRUS models preprocessing as a three-step pipeline:

(1) **Profiling:** Each feature is analyzed using three quality metrics: missing-value rate (m), outlier rate (c), skewness (s).
(2) **Decision Logic:** A rule-based system maps (m, c, s) to preprocessing strategies. Advanced imputation methods such as MICE [1] are applied for higher missingness, while robust outlier detection techniques such as Isolation Forest [4] are used for contaminated or non-Gaussian data. The rule-based decision logic is summarized in Table 1.
(3) **Pipeline Execution:** The selected preprocessing pipeline is applied and visualized interactively.

These rules define a structured decision space for consistent and explainable preprocessing. The current rules are heuristic and manually specified by design: the recommendation logic should remain transparent, inspectable, and adaptable to different eye-tracking settings. They are derived from common data quality indicators such as missingness, outlier rate, skewness, and distributional deviations. Future versions will compare these rules against learned or optimization-based recommendation strategies.

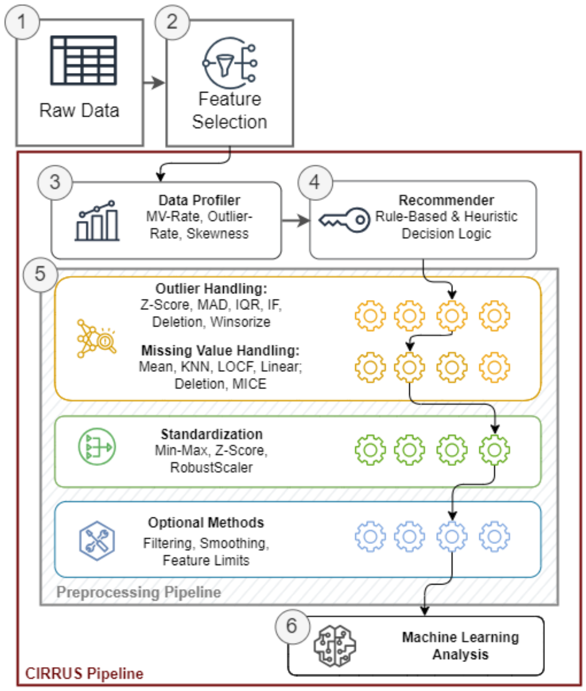

Figure 1: CIRRUS pipeline: data profiling, rule-based decision logic, and interactive preprocessing execution.

5 System Overview

CIRRUS is implemented in Python using Streamlit and integrates data upload, feature selection, profiling, rule-based recommendation, pipeline execution, visualization, and export. Its modular architecture separates profiling, decision logic, and execution. This supports reproducibility, as each preprocessing step is explicitly represented and can be re-applied or modified independently. The same separation provides a basis for future streaming and event-based extensions, which are not part of the current prototype.

6 Demonstration Scenarios

We illustrate CIRRUS through three representative scenarios: high missingness, outlier handling, and normalization. Each scenario combines automatic profiling, rule-based recommendations, and interactive comparison. CIRRUS turns preprocessing from trial-and-error into a structured and transparent workflow design process.

S1: High Missingness. We consider datasets with a missing-value rate $m > 15\%$, which frequently occurs due to blinks, tracking loss, or calibration drift. After upload, CIRRUS automatically identifies the high missingness level and recommends advanced imputation strategies such as KNN or MICE. Users can interactively switch between different imputation methods and observe their impact on temporal continuity and signal smoothness. In particular, the system highlights how simpler methods (e.g., mean imputation) introduce artificial plateaus, whereas structure-aware methods preserve local dynamics. This scenario demonstrates how CIRRUS translates abstract data quality metrics into concrete, interpretable preprocessing decisions.

S2: Outlier Handling. In the second scenario, we focus on datasets with moderate contamination ($c \leq 15\%$) and potentially skewed distributions. CIRRUS profiles both outlier rate and skewness and recommends robust methods such as MAD or Isolation Forest. Users can explore how different outlier handling strategies affect the reconstructed signal by comparing before/after visualizations. The system explicitly exposes the trade-off between noise reduction and structure preservation: aggressive methods remove extreme values but may distort natural variability, while conservative approaches retain noise. This scenario emphasizes the role of data-aware decision logic in selecting appropriate outlier strategies.

S3: Normalization and Distributional Effects. The third scenario examines the impact of normalization strategies under varying distributional properties. Based on skewness and contamination, CIRRUS recommends Z-score, Min–Max, or robust scaling. Users can interactively compare normalization outcomes and inspect how distributional characteristics (e.g., variance, range, and symmetry) change across methods. A summary view reports key statistics before and after preprocessing, enabling users to assess the implications of scaling choices for downstream analysis. This scenario highlights the importance of aligning normalization strategies with underlying data characteristics.

7 Initial Validation and Limitations

The current evaluation is exploratory and focuses on transparency and usability. As an initial quantitative indication, we tested 193 preprocessing pipeline configurations in previous experiments on eye-tracking data. These experiments showed that preprocessing choices can substantially influence composite data quality scores and downstream signal interpretation. Configurations combining IQR-based clipping with Z-score normalization achieved consistently strong composite scores across several settings. The results also indicate that intrinsic data quality metrics only partially align with downstream performance, motivating interactive comparison rather than a single fixed preprocessing strategy.

8 Conclusion

CIRRUS demonstrates how preprocessing can be formalized as a transparent decision-support process over data characteristics. By combining profiling, rule-based reasoning, and interactive visualization, the system enables more reproducible preprocessing workflows for eye-tracking data. Beyond eye-tracking, the proposed approach may be generalized to other types of time series and data-intensive applications where preprocessing plays a critical role. Future work includes systematic quantitative validation against downstream tasks, comparison with learned or optimization-based recommendation strategies, and extensions toward streaming and real-time data scenarios.

References

[1] Melissa J Azur et al. 2011. Multiple imputation by chained equations: What is it and how does it work? *International Journal of Methods in Psychiatric Research* (2011).
[2] Tanja Blascheck et al. 2014. State-of-the-art of visualization for eye tracking data. *EuroVis STAR* (2014).
[3] Kenneth Holmqvist et al. 2011. *Eye Tracking: A Comprehensive Guide to Methods and Measures.* Oxford University Press.
[4] Fei Tony Liu et al. 2008. Isolation Forest. In *IEEE ICDM.*