Making Stream Processing Auditable with Evidence-Carrying Windows

Ada Lovelace
Institute for Analytical Engines, Northbridge University
ada@example.edu

Charles Babbage
Department of Computing, Northbridge University
charles@example.edu

ABSTRACT

Streaming systems make decisions while data is still arriving, which makes it difficult to reconstruct why a particular alert fired. We present a small evidence-carrying window abstraction that stores the input range, feature summary, and decision threshold alongside each result. In a worked evaluation, the additional evidence makes incident review faster without changing the detector's alert rate.

KEYWORDS

stream processing, observability, reproducibility, audit trails

1 INTRODUCTION

Operational data products need both timely decisions and explanations that survive a later investigation. This starter illustrates the classic two-column ACM proceedings layout and its explicit title metadata.

2 EVIDENCE-CARRYING WINDOWS

For each processing window W_k, the system emits a result tuple

$$E_k = (t_k^{\mathrm{start}}, t_k^{\mathrm{end}}, \bar{x}_k, \theta_k, r_k), \tag{1}$$

where r_k is the decision and θ_k is the threshold in force at the time of evaluation. A reviewer can therefore replay the decision without guessing which configuration was active.

Copyright © 2026 The Authors

Table 1: Review effort in the worked evaluation.

Record type	Without evidence	With evidence
Median review time (min)	18	7
Replayed alerts	41	41

3 EVALUATION

We replay a month of anonymised equipment events and compare the standard log-only pipeline with the evidence-carrying variant. The latter adds a small fixed payload to each window, while Table 1 shows the reduction in review time.

4 LIMITATIONS AND CONCLUSION

The example assumes that the input stream can be retained for replay. A production deployment should define retention, access control, and schema versioning explicitly. The same metadata structure can also support cross-team incident hand-offs.

REFERENCES