Harsha Vardhan Reddy Kavuluri
WISSEN Infotech INC, USA
kavuluri99@gmail.com

Received: 18-04-2022; Revised: 04-06-2022; Accepted: 27-06-2022

Data Quality Scoring Models for Streaming Pipelines with Rule-Based and Learned Validation

Abstract
Streaming pipelines need continuous data quality scoring because errors can appear while events are still moving through ingestion, validation, transformation, and delivery stages. This article presents a hybrid quality scoring model that combines rule-based validation with learned anomaly detection to measure completeness, timeliness, validity, consistency, uniqueness, schema stability, distribution stability, and source reliability. The proposed model profiles each event, applies deterministic validation rules, detects abnormal stream behavior using learned signals, and aggregates scores across sliding processing windows with temporal weighting. Simulated results show that the total quality score declined during unstable windows, mainly due to timeliness and uniqueness degradation, and later recovered as the stream stabilized. The hybrid model also showed stronger diagnostic value than rule-only validation because it detected both explicit failures, such as schema drift and duplicate spikes, and hidden behavioral changes, such as distribution shift. These findings indicate that streaming data quality should be represented as an interpretable dynamic score rather than a simple pass-or-fail decision.

Keywords: Streaming data quality, rule-based validation, learned validation, anomaly detection, quality scoring, sliding-window aggregation, data governance.

1. Introduction

Streaming data pipelines have become a central part of real-time digital systems because many organizations now depend on continuous event processing for monitoring, analytics, automated decision-making, fraud detection, operational dashboards, IoT supervision, and machine learning feature generation. Unlike batch pipelines, where data can be checked after collection and before loading, streaming pipelines require quality assessment while events are still moving through ingestion, transformation, enrichment, and delivery stages. This creates a more difficult validation environment because missing values, delayed events, duplicated records, schema changes, invalid formats, and abnormal distributions may appear at any time and must be detected without stopping the full stream. Streaming data quality metrics are important because continuous monitoring must evaluate data reliability during live processing rather than after the stream has already been stored [1]. A streaming quality model therefore needs to work as an active control layer that measures trust, detects degradation, and supports operational routing in near real time.

Traditional data validation usually depends on deterministic rules that define whether incoming data satisfies expected structural and semantic conditions. These rules may check whether required fields are present, whether data types are correct, whether numeric values fall within acceptable ranges, whether timestamps are valid, whether categorical values belong to approved lists, and whether duplicate event identifiers are present. Rule-based validation is valuable because it is transparent, easy to audit, and directly linked to business or engineering expectations. However, streaming environments often contain problems that do not appear as simple rule violations. Stream-first data quality monitoring is useful because live pipelines require validation methods that operate continuously across event windows, source behavior, and time-dependent conditions [2]. This means that a practical scoring model must combine fixed rules with adaptive signals that can detect unusual stream behavior even when the data remains technically valid.

Streaming data quality also depends strongly on the source environment that generates the events. In smart manufacturing, sensors may send noisy readings, delayed measurements, or incomplete telemetry. In financial systems, transaction streams may contain bursts, retries, reversals, or duplicate notifications. In healthcare and IoT environments, device instability, network delay, and measurement inconsistency can reduce the trustworthiness of incoming data. Data quality assessment in smart manufacturing shows that missing values, noise, synchronization problems, and unstable sensor behavior can directly affect data reliability [3]. Data quality management in IoT systems also shows that device-level and communication-level problems can influence the completeness, timeliness, and validity of collected data [4]. These examples show that streaming data quality must be measured across multiple dimensions instead of being reduced to a single acceptance rule.

A rule-only validation model can reject clearly invalid records, but it cannot always identify slow deterioration in stream behavior. A field may satisfy its expected range but gradually move away from its historical distribution. A timestamp may satisfy the expected format but still arrive with abnormal delay during a burst. A source may continue sending data but gradually increase its null rate, duplicate rate, or schema instability. Data quality modeling frameworks emphasize that quality should be represented through measurable dimensions rather than informal inspection or isolated error counts [5]. This supports the need for numerical scoring models that can combine deterministic rule failures, learned anomaly indicators, recent degradation patterns, and source reliability into a single interpretable quality signal.

This article proposes a hybrid data quality scoring model for streaming pipelines using rule-based and learned validation. The model evaluates event-level quality, aggregates quality across sliding windows, applies temporal weighting to recent degradation, and routes low-quality streams to alerts, quarantine paths, or additional inspection stages. The framework is designed to produce interpretable quality scores that show which dimension is failing and whether the failure is caused by deterministic rule violations, learned behavioral anomalies, or both. The aim is to support real-time quality governance without treating streaming data only as accepted or rejected records.

2. Methodology

The methodology begins with event-level profiling, where each incoming event is inspected before it enters downstream transformation or analytical stages. The profiler extracts key attributes such as source identifier, event timestamp, ingestion timestamp, schema version, event key, required-field presence, numeric value ranges, categorical values, and basic statistical properties. This profiling step creates a compact quality vector that represents the current condition of the event. Automatic data validation methods can improve validation precision when checks are generated from structured constraints and observed data behavior [6]. In the proposed model, event-level profiling becomes the shared evidence base for both deterministic validation and learned anomaly detection, allowing the scoring process to begin at the earliest point of ingestion.

The rule-based validation layer performs checks that are explicit, reproducible, and easy to explain. It evaluates null constraints, data type conformity, accepted value ranges, regular expression matches, timestamp delay limits, duplicate event identifiers, schema compatibility, and cross-field dependency rules. For example, a transaction event may require that amount is positive, currency belongs to an approved list, and event_time does not exceed a defined delay threshold. Data quality assertion systems show that validation rules can be attached directly to pipeline behavior and used as executable quality constraints [7]. This layer is particularly useful when engineering teams already know the expected structure, accepted values, and business rules for a stream.

The learned validation layer is designed to identify quality problems that are not captured by deterministic checks. It uses historical stream behavior to evaluate whether current values, delays, frequencies, source activity, field distributions, and event sequences are unusual. Learned validation may include unsupervised anomaly detection, distribution-shift scoring, sequence similarity analysis, source behavior modeling, and temporal deviation detection. Actionable data monitoring in modern streams requires methods that identify changes in data behavior and connect them to operational decisions [8]. This layer is therefore important when a stream remains formally valid but starts behaving differently from its expected historical pattern.

The hybrid quality score combines rule-based penalties with learned anomaly signals. Each event begins with a maximum score, and penalties are applied according to missing fields, late arrival, invalid values, duplicate risk, schema instability, distribution deviation, and source-level reliability. The score is not designed only to reject records; it is designed to quantify how trustworthy the record is within the current processing context. Unsupervised anomaly detection is useful in this setting because many streaming failures are not labeled in advance and may appear as unusual patterns rather than known error categories [9]. The hybrid score therefore captures both known quality failures and unknown behavioral deviations within a unified scoring structure.

Temporal weighting is applied because streaming quality is time-sensitive. A stream that experienced poor quality several windows ago but has now stabilized should not be penalized as strongly as a stream that is currently degrading. The model gives higher weight to recent windows and lower weight to older windows so that the score reflects current operational risk. Streaming anomaly detection studies show that temporal behavior and sequence-level variation are important for identifying abnormal conditions in continuous data [10]. In the proposed model, temporal weighting helps distinguish persistent quality problems from short-lived disturbances, making alerts more meaningful and reducing unnecessary escalation.

Table 1. Streaming Data Quality Scoring Dimensions and Validation Logic

Quality Dimension	Streaming Risk Captured	Rule-Based Signal	Learned Signal	Score Impact
Completeness	Missing or null event fields	Required-field failure rate	Expected-field presence probability	Reduces event-level score
Timeliness	Late or delayed events	Event-time delay threshold	Learned delay distribution	Reduces window-level score
Validity	Invalid values or formats	Regex, range, enum, type rules	Value-pattern anomaly score	Reduces field-level score
Consistency	Conflicting values across fields	Cross-field dependency rules	Correlation deviation score	Reduces event integrity score
Uniqueness	Duplicate or repeated events	Event ID duplicate check	Sequence similarity anomaly	Reduces stream reliability score

Schema stability	Unexpected field changes	Schema compatibility rule	Structural drift probability	Triggers schema risk penalty
Distribution stability	Sudden value-distribution shift	Threshold-based distribution check	Learned drift score	Reduces adaptive quality score
Source reliability	Recurrent source-specific failures	Source error count	Source behavior risk score	Reduces source trust weight

Sliding-window aggregation converts event-level scores into stream-level quality indicators. The model calculates completeness, timeliness, validity, consistency, and uniqueness scores for each processing window. These component scores are then combined into a total quality score that represents the current trust level of the stream. A very short window improves responsiveness but may produce unstable scores when only a few events are abnormal. A long window improves stability but may delay the detection of rapid quality degradation. The proposed approach uses overlapping windows so that sudden quality changes can be detected quickly while still preserving enough context to avoid overreacting to isolated events.

Alert thresholding and quality-based routing are applied after the window-level score is produced. Events from high-quality windows move through the trusted stream path and are made available for real-time analytics or downstream applications. Events from moderate-quality windows are allowed to continue but are tagged with warning metadata, allowing downstream systems to treat them with caution. Events from low-quality windows are routed to quarantine, delayed correction queues, or manual review paths. The scoring layer also produces an explanation record that identifies whether the dominant quality loss comes from completeness, timeliness, validity, consistency, uniqueness, schema stability, distribution shift, or source reliability.

The evaluation design uses simulated streaming conditions that represent common quality failure scenarios in real-time pipelines. These include normal flow, late-event bursts, schema drift, duplicate spikes, and distribution shifts. The model is evaluated using component-score behavior, rule-based rejection rate, learned anomaly detection rate, and hybrid quality score. The first result examines how individual quality dimensions contribute to the total score across processing windows. The second result compares rule-based validation, learned anomaly detection, and hybrid quality scoring under different stream conditions. This evaluation focuses on whether the scoring model can explain quality degradation, not only whether it can reject bad records.

3. Results and Discussion

The simulated results show that streaming data quality changes across processing windows rather than remaining constant throughout the pipeline. During stable periods, quality dimensions remain high and balanced, which indicates that events are complete, timely, valid, consistent, and mostly unique. During

unstable periods, specific dimensions decline more sharply than others, revealing the type of quality problem affecting the stream. A simple pass-or-fail validation strategy cannot represent this behavior because it only shows whether a record crossed a threshold. The proposed component-level scoring approach provides a more useful view because it shows how quality changes, which dimensions are responsible, and whether the stream is recovering.

The component score pattern shows that total quality declines during the middle processing windows and then improves as the stream stabilizes. Completeness decreases from 22 in W1 to 18 in W4, while timeliness decreases from 19 to 14 during the same interval. Validity, consistency, and uniqueness also decline, although their reductions are less severe than the timeliness drop, as shown in Figure 1. This indicates that the simulated degradation is not caused by one isolated field error but by a combined disturbance in arrival delay, missing-field behavior, and duplicate risk. The later recovery from W6 to W8 shows that the model can represent quality restoration instead of treating the stream as permanently degraded.

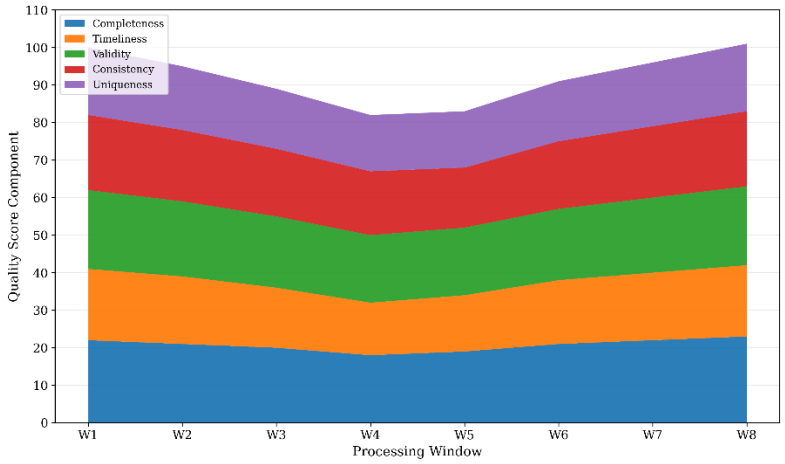

Figure 1. Streaming Data Quality Score Components Across Processing Windows

The middle-window decline is mainly driven by timeliness and uniqueness loss, which is realistic for streaming systems under burst ingestion or source retry behavior. When a source experiences delivery delay, events may arrive late and may also be resent, creating both timeliness penalties and duplicate-risk penalties. Completeness and validity remain less volatile because field presence and value-format rules are usually more stable during short operational disturbances. This distinction is important because different quality failures require different responses. A late-event burst may require buffering or watermark adjustment, while schema drift may require engineering intervention before downstream processing continues.

The comparison across stream conditions shows that rule-based validation and learned validation respond differently to quality failures. Rule-based rejection is strongest for schema drift and duplicate spikes because these problems can be captured through schema compatibility rules and event identifier

checks. Learned anomaly detection is strongest under distribution shift because the values may remain within valid ranges while becoming statistically unusual. Hybrid quality scoring combines these two perspectives and produces a more operationally useful trust signal, as shown in Figure 2. This combined signal is especially useful when a stream contains both explicit rule violations and hidden behavioral changes.

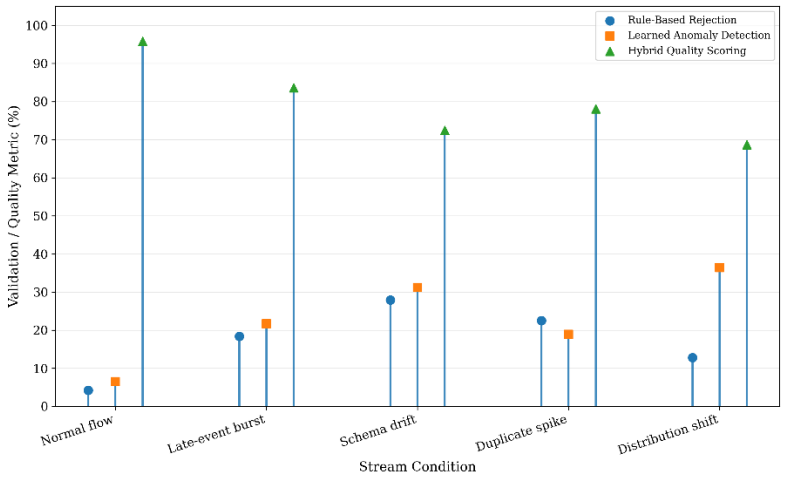

Figure 2. Rule-Based Rejection, Learned Anomaly Detection, and Hybrid Quality Scoring Across Stream Conditions

The results indicate that hybrid scoring is more informative than rejection counting alone. A high rejection rate explains that records are failing known validation rules, but it does not capture statistically unusual patterns that still satisfy deterministic constraints. A learned anomaly score detects hidden behavioral changes, but it may not explain which business rule or field-level expectation failed. The hybrid model balances interpretability and adaptability by preserving rule-level explanations while adding sensitivity to drift, abnormal delay, and unusual stream behavior. This makes the scoring model suitable for real-time monitoring, alerting, quarantine routing, and downstream trust control.

4. Conclusion

Streaming data quality scoring requires a different design from traditional batch validation because quality must be measured continuously while events are still being processed. The proposed model combines rule-based validation, learned anomaly detection, temporal weighting, sliding-window aggregation, and quality-based routing to produce a more complete view of stream trustworthiness. Instead of treating streaming records only as accepted or rejected, the model assigns interpretable scores across completeness, timeliness, validity, consistency, uniqueness, schema stability, distribution stability, and source reliability. This allows pipeline operators to understand not only that quality has degraded, but also why it has degraded and which part of the stream is responsible.

The results show that quality scores can reveal temporary degradation, recovery behavior, and dimension-specific failure patterns. The component-score analysis shows how completeness, timeliness, validity, consistency, and uniqueness contribute differently across processing windows. The validation comparison shows that deterministic rules are stronger for known structural and duplicate-related failures, while learned methods are stronger for distribution shifts and hidden behavioral changes. The hybrid scoring model combines these strengths and provides a balanced signal that supports both explainability and adaptive detection. This is important for real-time systems because a quality issue can affect dashboards, alerts, machine learning features, or automated decisions within seconds.

Future work can extend the model with adaptive score weights, source-specific learning, downstream feedback loops, and automated correction recommendations. The scoring method can also be tested under larger streaming workloads, multi-tenant environments, high-cardinality event streams, and stricter event-time disorder conditions. Additional research may examine how quality scores should influence feature freshness, model-serving confidence, alert suppression, and trusted data product certification. A hybrid quality scoring model provides a practical foundation for streaming pipeline governance because it connects validation, anomaly detection, routing, and operational decision-making in one interpretable framework.

References

1. Yu, M., Wu, C., & Tsung, F. (2019). Monitoring the data quality of data streams using a two-step control scheme. *IISE Transactions*, *51*(9), 985-998.
2. Hynes, N., Sculley, D., & Terry, M. (2017, February). The data linter: Lightweight, automated sanity checking for ml data sets. In *NIPS MLSys Workshop* (Vol. 1, No. 5, p. 10).
3. Schelter, S., Lange, D., Schmidt, P., Celikel, M., & Biessmann, F. (2018). Automating large-scale data quality verification.
4. Zhang, L., Jeong, D., & Lee, S. (2021). Data quality management in the internet of things. *Sensors*, *21*(17), 5834.
5. Taleb, I., Serhani, M. A., & Dssouli, R. (2018, November). Big data quality assessment model for unstructured data. In *2018 International Conference on Innovations in Information Technology (IIT)* (pp. 69-74). IEEE.
6. Polyzotis, N., Zinkevich, M., Roy, S., Breck, E., & Whang, S. (2019). Data validation for machine learning. *Proceedings of machine learning and systems*, *1*, 334-347.
7. Doehmen, T., Raasveldt, M., Mühleisen, H., & Schelter, S. (2021). Duckdq: Data quality assertions for machine learning pipelines. In *Workshop on Challenges in Deploying and*

Monitoring ML Systems at the International Conference on Machine Learning (ICML) (Vol. 2021).

8. Ahmad, S., Lavin, A., Purdy, S., & Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, *262*, 134-147.
9. Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*.
10. Fernandes Jr, G., Rodrigues, J. J., Carvalho, L. F., Al-Muhtadi, J. F., & Proença Jr, M. L. (2019). A comprehensive survey on network anomaly detection. *Telecommunication Systems*, *70*(3), 447-489.