Deep Science Publishing, 2026
https://doi.org/10.70593/978-93-7185-978-3_4

Chapter 4: Real-Time Clinical Data Exchange and Streaming Systems

4.1. Introduction

Technological advances, most recently driven by modern mobile devices, have transformed the way society uses the Internet. Real-time streaming data sources are proliferating and are expected to grow significantly. Streaming systems in healthcare make clinical data available in real time for immediate consumption, enabling real-time decision support, telemetry, remote patient monitoring, and other applications that demand or would benefit from real-time information. The considerable latency of conventional clinical data repositories necessitates a complementary approach that delivers data in live flow as it arrives from the sources. In streaming systems, raw data are presented in a more immediately consumable form, ready for both operational and analytical use. These complementary systems currently remain separate from the clinical repository in terms of development, deployment, operation, and support.

The foundations of these systems are based on a clear understanding of the data sources in healthcare and of the streaming data processing paradigms. The pioneering work in streaming data systems is clearly relevant to the broader healthcare domain. The architecture reflects the functionality of streaming systems in other domains while acknowledging the special requirements of the healthcare environment. Real-time decision support and telemetry of patients in critical condition are cited as immediate applications, along with other uses that require informative data made available as soon as they are generated. Quality and integrity issues are examined against the special context of clinical environments, and the challenges involved in evaluating and managing risk are explicitly addressed.

4.2. Foundations of Real-Time Clinical Data Exchange

Healthcare generates rich volumes of sensor data. A large share of them is time-oriented, appears in the format of continuous flows, commonly designated as streams, which are characterized by the autonomous generation of events at regular or irregular intervals. Monitoring devices produce streams with different temporal granularities. Some devices provide tick-by-tick updates, while others transmit aggregated values at fixed or adaptive time intervals. In addition to classical EHRs, all integration and monitoring systems must accommodate various Internet of Things (IoT) devices that generate regular or sporadic streams. In the telemetry domain, a plethora of heart-rate monitors, streaming echocardiograms, blood-gas analyzers, and other devices stream clinical attributes relevant to medical diagnosis. Therefore, real-time clinical data-exchange technology must cope with flows evolving with respect to their state-space dimension, temporal alignment, and correlation or anti-correlation.

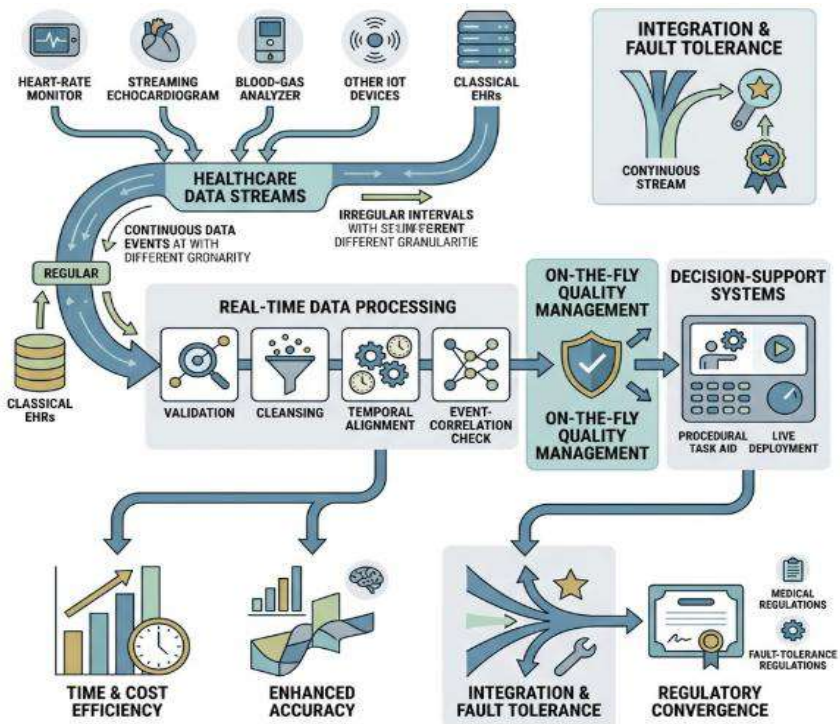

Fig 4.1: Real-Time Stream Quality Management for Decision-Support in IoHT: Challenges in Integration and Fault Tolerance

The quality of a stream is critical for real-time decision-support systems. Such applications are often deployed in live settings for a limited period of time to aid procedural tasks. Several data-quality aspects, including validation, cleansing, temporal alignment, and event-correlation checks, must be performed on-the-fly to maintain time and cost efficiency. Healthcare is notorious for the lack of data-quality standards and conventions, and no standard normative document specifically addresses data quality for continuous flows in health-related domains. Current literature presents proofs-of-concept deployed in proof-of-concept environments, where data-integration aspects addressed by medical and fault-tolerance regulations converge with those standard approaches employed in information-system practices. Research into quality or fault tolerance in streams is limited, but the preprocessing tasks involved introduce overhead that indirect downplay the crucial need for such tasks in decision-support streams.

4.2.1. Data sources in healthcare

The healthcare industry generates a wealth of clinical data. Any given patient can produce orders of magnitude more clinical data than can be examined and stored by healthcare organizations. Large amounts of data related to chronic conditions and surgical processes can be combined and stored for longer-term analysis; however, urgent clinical situations still require the analysis of short time windows from the data streams. The requirements of clinical personnel frequently change, leading to further diversity in the needs and demands of data exchange systems in healthcare. Furthermore, the clinical environment operates in real time and a continuous flow of data is generated. Beyond the traditional data sources of healthcare information systems, data is generated by monitoring devices for real-time displays, prognostic calculations, and clinical decision support tools, as well as data from telemetry and remote patient monitoring. The next logical step is to make these data automatically available for analysis and use. These attributes of clinical data suggest that streaming solutions are well suited for modern healthcare systems.

Real-time decision support systems can display the output of complex models as part of a regular clinical display, displaying many hypertensive patients on a single screen. In telemetry, data from multiple patients is transmitted to a dedicated nursing station for central monitoring. Cost containment and the shortage of nursing resources demand that each nurse supervise a larger number of patients, leading to the increased use of telemetry. Remote patient monitoring offers the ability to monitor well patients in their homes, thereby enhancing their quality of life. Finally, the stream-processing paradigm addresses such fast- and large-volume real-time data exchange between data sources and clients.

4.2.2. Streaming data paradigms

In clinical settings, as in many domains, some data sources provide continuous outputs, while others emit events that reflect certain states of a system. Signals from medical devices (electrocardiograms, oxygen saturation levels, etc.) are well-suited for continuous processing because they reflect the user's status in real time. Conversely, a new medical event (admission to hospital, discharge from hospital, or the prescription of a new treatment) expresses a change at a higher level of abstraction, emerging infrequently and warranting further consideration and possibly interaction with a human operator. Because such events modulate the input of many complex systems, it is crucial to process them with very low latency.

Stream processing systems are the natural fit for the treatment of data provided as continuous signals. Key characteristics of those systems include the following. They preserve the temporal ordering of the original data. They permanently associate a time instant or time interval with each datum, which may be used for time-based selection or aggregation operations. They process the data whenever possible and provide an up-to-date "view" of the entire data space, which may be queried at any time through a traditional database access. Operations are specified as queries expressed in procedural or declarative languages that share many of the characteristics of SQL. Filters, projections, joins, and aggregations are readily available in operators specifically tailored for streaming. Latency-driven delivery semantics guide both the internal architecture of the systems and the behavior of the operators.

4.3. Architecture of Real-Time Streaming Systems in Healthcare

A real-time streaming architecture is proposed to support clinical decision-making and the enhancement of patient monitoring systems. The architecture operates continuously and allows imminent events to be handled in a time-sensitive manner, often using short-lived stream processing units or servers. Several points from the system's design have been examined: the ingestion module, the data spec for a highly heterogeneous clinical environment, and the definition of stream processing engines. In addition, real-time readiness, an important property for streaming systems that typically goes unnoticed, is considered.

Real-time streaming systems in healthcare derive their real-time capability from two key architectural aspects. First, they are continually active, except during maintenance and unforeseen failures. Second, they manage temperature-sensitive streams that require rapid response times, explicitly steering short-lived units for possible bottlenecks in a time-sensitive manner. These architectural characteristics impose additional requirements on the system‟s components, especially the ingestion module and the stream processing engines. The need for further specification and validation of the data throughput specifications remains open for future work. The value of the architecture is illustrated by a telemetry service that supports clinical decision-making and augments remote patient monitoring systems.

4.3.1. Data ingestion and adapters

Real-time streaming systems depend on the timely influx and continuous processing of data from various sources to support immediate clinical response. Data may originate from different areas within a healthcare institution, such as emergency or intensive care units, hospitals, and laboratories, as well as from mobile devices or sensors used for remote patient monitoring. However, simply accepting data from different sources and tasks is not sufficient; dedicated, specialized systems called adapters are usually deployed to cater for each type of source, converting incoming information into the format required by the processing engine of the target streaming system. If carelessly designed, adapters can become bottlenecks, introducing latency into the streaming system. Their specifications should explicitly define achievable throughput, given real-world deployment scenarios and realistic operational conditions, such as load azimuth and distribution, and normal transient behaviour.

In principle, the concept of a stream adapter may be common to any application domain, yet the nature of the actual medical device or system generating the incoming data may require different considerations, such as protocol or message format and data content (i.e., structure, semantic, and context). Due to such peculiarities, it is not realistic to implement stream adapters for medical devices and systems generically; instead, such components are likely to be developed as dedicated single implementations. The integration of different remote-control medical devices for real-time support is also a possibility, extending the system’s reach, moving the treatment from a hospital to the patient’s home.

4.3.2. Stream processing engines

Due to the dynamic and time-critical nature of data in real-time streaming systems, stream processing engines offer a different execution model than traditional batch processing systems. Instead of processing data by breaking it into self-contained units, the stream processing approach enables operators to manage state, maintain correctness, and control windowing out of band using timestamps present in each input tuple. This model is not only resource efficient, but also provides the flexibility to execute complex temporal queries over input sequences with minimum latency. These engines can be deployed in either a standalone manner or with a specialized resource-efficient building block integrated into a complex event processing framework.

The information model of stream processing frameworks is also different at a conceptual level compared to the reduced information model used in other management systems. The resulting complexity must be handled with care: decisions in the design phase can have a significant effect on the correctness or performance of the resulting distribution.

The infrequent visitors arriving at streaming systems do not rely on the presence of complex aggregation semantics like queries with HAVING clauses or the flexibility of pure SQL queries. Moreover, when the processing system is accepting a data rate which exceeds its processing capacity, the quality of the response may be in doubt based on the latency-sensitive nature of the domains.

4.4. Data Quality and Integrity in Live Clinical Environments

Live clinical environments pose stringent requirements on data quality and integrity that must be addressed at the stream level, affecting every aspect of stream processing. Stream validation and cleansing mitigate untrustworthy data points, while the temporal alignment and event correlation operations ensure downstream analytics leverage fully assembled patient states and timely information.

Detection of outliers and missing values, and the correction or replacement of invalid records, are the most-supported steps in stream cleansing and validation phases. Predictive models, costly-checking services, expert-systems heuristics, and pre-trained classifiers are among the common approaches deployed. Constructing, operating, and testing such facilities constitute a specialized domain in their own right.

Temporal alignment condenses multiple streams from the same patient into a single stream containing only the most recent value for each attribute. Event correlation takes patient streams, correlates events from all patients, and computes aggregate attributes of interest from the correlated events, such as the number of active cardiac arrests or the number of patients in sepsis, for each time point.

4.4.1. Validation and cleansing in streams

Inherent data qualities must be monitored during data ingestion and processing to ensure dependable outcomes. Validating on-line data against existing knowledge-based rule systems can help identify data that does not meet requisites for further processing or storage, for example, during clinical-rule-based decision support. Processes that validate certain properties at stream or query time vary significantly from static cleansing, as the outcome is a filtering effect during life without modifying the original data streams.

Heuristic-based cleansing is a commonly employed approach for rectifying streams in general database environments. Such heuristics can be exploited in clinical environments by leveraging knowledge bases that retain frequently used transformation rules for readily identifiable errors. To be entirely trustworthy, the knowledge base must be regularly updated. Existing soft computing-based techniques can also be evaluated for their suitability in identifying potential cleansing heuristics.

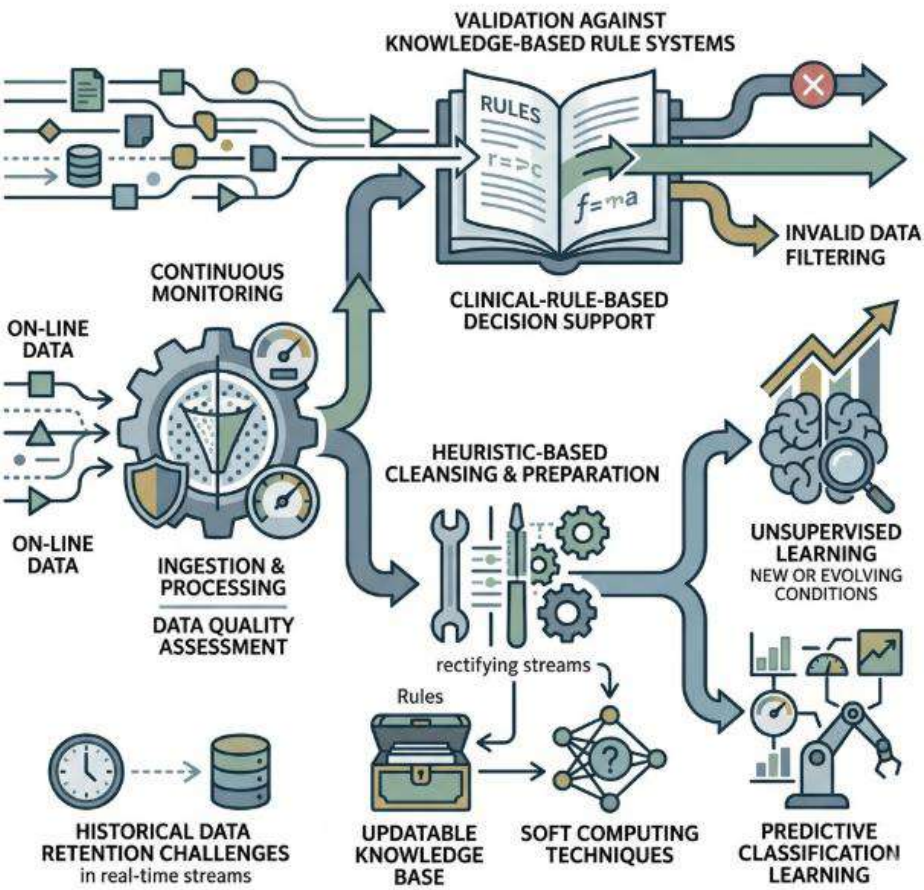

Fig 4.2: Ensuring Trustworthy Real-Time Clinical Data Streams: A Framework for Continuous Data Quality Monitoring, Knowledge-Based Validation, and Heuristic Cleansing to Enhance Machine Learning Outcomes

Data cleansing and preparation complete the data quality breadth in screening and filtering streams and should complement the validation. Such practices enhance the predictive performance of classification learning and the outcomes of unsupervised learning used for discovering new or evolving clinical conditions or other phenomena hidden in the data. Nevertheless, preparing real-time or near-real-time streams for machine learning is also a non-trivial task, primarily due to the loss of historical data associated with standing stream data approaches.

4.4.2. Temporal alignment and event correlation

Temporal alignment and event correlation The success of algorithms predicated on temporal congruity relies on the implicit assumption that temporally equidistant observations in distinct streams correlate closely. Nevertheless, shifts in acquisition epochs, sampling cadences, and rates of data drift among channels give rise to discordance even among detections encountered from diverse patients and clinical scenarios. Such phenomena render the mere temporal alignment of external data with the native telemetry stream insufficient to fulfil the requirements of analytics that seek to exploit temporally correlated features across multiple streams.

Substantial evidence has accumulated in other research domains to suggest that event-correlation detection models based on observer motion—even without specific details of the observers themselves—are proficient in capturing semantically similar but temporally varied events within a single stream. Thus, models trained solely on one telemetric channel can be deployed to identify landmark events in other largely independent telemetry streams, supporting the inference of alterations in risk states of patients under monitoring. Moreover, building on the notion of parsimony, state-aware classification models can be structured to directly leverage distinctive changes in the telemetry physiology of patients presenting in specific states, while also consuming temporally aligned external data for reinforcement.

4.5. Applications and Use Cases

Real-Time Clinical Data Exchange and Streaming Systems are the technological basis for many applications and use cases in the healthcare domain. Characteristic applications involve real-time clinical decision support in the context of critically ill patients, hybrid systems designed for remote diagnostics and imperative telemetry, and runtime event-driven processing in clinical data streaming. These implementations provide examples of streaming architectures and demonstrate how real-time clinical data exchange becomes possible.

Inspired by a personal involvement with research in hospital telemetry for artificial ventilation, an exposition of this work focuses on three major applications found in the literature: aggregate real-time decision support for critical care, a hybrid system supporting scientific research in remote diagnostics and imperative telemetry, and runtime event-driven processing in clinical data streams. The basic requirements for these classes of scenarios are also outlined, paying special attention to novel aspects when compared to the standard requirements for general real-time processing. A secondary aim of the discussion is to present recent directions in clinical data streaming that start to transcend simple continuous real-time processing and move towards adaptive handling of arbitrary events raised in the data stream. These developments exploit the true event-driven nature of clinical data streaming and suggest that continuous processing and decision support are just preliminary steps in the streaming data service chain.

4.5.1. Real-time decision support

Clinical decision support systems process information to produce knowledge. Real-time processing makes use of active data; knowledge generated from that data is automatically delivered to the users for consideration just before they are about to act—in a similar manner to how an advisory system would inform the user. Real-time clinical decision support systems typically provide advisory support by supplying alerts, such as warning clinicians about abnormal patient conditions derived from outside patient normal ranges defined for specific clinical variables (e.g. heart rate). Such alerts are typically based on data from dedicated vital sign monitors.

Alerting based on historical data is also possible. Alert conditions may be defined by a rule-based engine to trigger an advisory alert message when an accident or incident is about to occur based on historical series analysis of different univariate or multivariate data—such as a rapid increasing of CO2 concentration, depleting O2 in an underwater vehicle, etc. In this case, computations based on past streaming data are required in addition to live data acquisitions to determine when current events are triggering an alert. Unsupervised learning techniques are also being explored to detect and classify abnormalities in telemetry data.

4.5.2. Telemetry and remote patient monitoring

A common application of clinical data streaming is patient telemetry and remote monitoring, where streams convey the status of patients with chronic diseases or at high risk. Both traditional in-room telemetry and remote monitoring of at-home patients can benefit from a complete view of patient status that goes beyond monitoring of simple vital signs and includes multimodal records such as clinical observations, lab test results, and diagnoses.

In a telemetry context, event streaming enables telemetered patients to exhibit changes in preventive-care status that are relevant to their level of risk, thereby triggering alerts when a care provider's intervention is required. Telemetry feeds to the streaming system provide vital signs and other simple measurements, while streaming feeds from the rest of the hospital supply updates in the patients' preventive-care status. Assessment of the completeness of the status change and a check for possible relationships along the temporal dimension of the streams permit confident alert generation. These ideas are resilient to session interruptions such as those caused by sensor failures and network drop-outs, which are not uncommon in stream environments.

4.6. Challenges and Risk Management

Maintaining a high level of data quality, integrity, and completeness (Q IC) is a key challenge in live environments. The increased focus on streaming analytics raises several important risk-management aspects. Whereas one hundred percent accuracy may be desired in batch systems, real-time systems trade off some level of accuracy for reduced latency, particularly in live environments. Calibration, validation, and testing of the stream-based approach become even more important, particularly for applications reliant on real-time data that can carry high risk for patients, such as automated drug dosage administration and real-time monitoring of anomalies in subjects' physiological status.

LatEnt Agencies Users' Deployment scenarios have an impact on the basic concept as to when latency becomes quality. For example, system design for pollution in a city is very different from one related to tsunami warning systems. Pollutants are perhaps most pollination in the northern hemisphere during the summer months; thus, prediction and corrective actions, rather than sensor detection, must be the design strategy. Pollutants are not disposed of in the equatorial region but in the dry regions of the world. Conversely, tsunami waves have periodicities of 4–5 minutes, and humans have little ability to respond to them; detection and warning are key.

4.6.1. Latency versus accuracy trade-offs

Real-time clinical data exchange and streaming systems presented here are especially suited for environments in which clinical decisions are supported by real-time information. Events or conditions that may grow worse with time-such as a cardiac arrest—are particularly suitable for analysis in a streaming paradigm in order to rapidly leverage all available evidence. The calculus of the system is limited to visible sequences, those that lie within the current time window, and thus it can operate with low latency and high throughput. The demonstration focused on detecting trends in the patient's heart rate monitoring data, which correspond to a progressively worsening sinus bradycardia condition. Other telemetry and remote patient monitoring applications may benefit also, as these systems continuously collect and stream patient information and alert personnel of events of interest.

Streaming systems must be designed to integrate the non-similar data arriving from multiple sources. Nevertheless, as evidence receives a temporal stamp, tardy incoming evidence may incorrectly support an alert that should have been generated earlier—e.g., the patient "is bleeding badly"—or late or incomplete evidence may not support a decision that could have been generated after its receipt—e.g., a patient "may need a blood transfusion." Such lapses in accuracy reduce physicians' trust in the alerts and their agreement in accepting or rejecting them, and they may ultimately lead to patients' lives being placed in danger.

4.6.2. Scalability and resilience

Real-time clinical data streaming systems must maintain a fine equilibrium between low latency and complex processing. While present-day systems rely on central processing units (CPUs), it is likely that they will need to transition to adaptive architectures incorporating heterogenous processing resources to retain their competitive advantage. The enormous processing loads foretasted by several ICT-enabled innovative use cases and future requirements will only be met by systems having the capacity to scale out in the cloud by engaging graphic processing units (GPUs), tensor processing units (TPUs) or Field Programmable Gate Arrays (FPGAs), thereby improving performance without impacting growth factors like resilience and cost into the bargain. The additional risk of working at the extremes of trade-offs between latency, accuracy and speed is that even small equipment failures can have a major negative impact on the entire system. In situations where the stakes are unusually high, such as clinical environments assisting serious accident victims, sufficient redundancy must be built into such systems to achieve the desired service level and keep human lives out of harm's way.

In addition to Equipment Redundancy, the adoption of software architecture that provides Multiple Levels of Resilience is considered good practice in Mission Critical Systems. Platform Data Streams such as those derived from standard building block Data Ingestion units have been proven to exhibit an inherently high degree of resilience using simple techniques such as data resilience streaming analytics, temporal-spatial performance regression, and extreme anomaly detection based on extreme value theory, ensuring a good balance between capacity to absorb failure and complexity of processing. This aspect is considered relevant to the end-consumer, because the risk of failure of a low-complexity system may be perceived as being far greater than that of a more complex system, even if resilience measures in the latter compensate for the increased chance of failure.

4.7. Evaluation and Validation Methodologies

A streaming engine has traditionally been assessed based on throughput and latency – when analysing business processes these metrics are paramount for production control and decision-making. However, in clinical environments patient safety is essential, therefore when real-time support is provided for critical situations, such as alerts, the ability to issue accurate notifications is more essential than swift notifications. In such circumstances, the time window for producing a notification can be extended without a severe decrease in the usefulness of the alert. An unexpected event not causing human harm needs an immediate reaction, but the opportunity fence is wider than for dangerous conditions. Telemetry and telemetry monitoring services require a balance between these metrics, as typically a large volume of critical and non-critical events must be processed.

WS-Notification compliance testing displayed the expected scores sharing between message-oriented and event-driven areas. Moreover, testing the integration of a custom data-integration library with a third-party system highlighted the advantages of having a dedicated set of dedicated services to handle batch and streaming data consolidation. These libraries managed different configurations and connection parameters for disparate data sources. They controlled network communication protocols for message-oriented systems and were open for external extensions. Additionally, these components offered built-in data aggregation in the batches to reduce the number of published notifications, and the engine health information was made available for notifications about the reporting system. These areas of validation can simplify the process of maintaining streaming engine components tested at different granularity levels.

4.7.1. Performance metrics for streaming systems

Benchmarks that assess the performance of real-time streaming systems are fundamental for evaluating different approaches. Nevertheless, comparing the performance of distinct architectures or implementations is difficult. As a result, many performance metrics need to be kept in mind, and the main focus must remain on identifying, in a structured way, the most relevant metrics for each specific case. A general analysis has been devised for the performance evaluation of stream processing engines. The proposed metrics are categorized into five groups: throughput, latency, resource consumption, fault tolerance, and reliability and availability.

378. Throughput is defined as the number of successfully processed events during a given timeframe. Depending on the requirements, it can be defined per second, minute, hour, or day, and may also be presented as requests per second. Latency denotes the time required to process an incoming event. The subsequent category is resource consumption, which encompasses metrics pertaining to memory usage, CPU occupancy, and other relevant aspects. Fault tolerance is framed as the implementation's ability to recover from a failure without data loss or duplication. Finally, reliability and availability metrics are related to the capability of sustaining operations in the presence of failures.

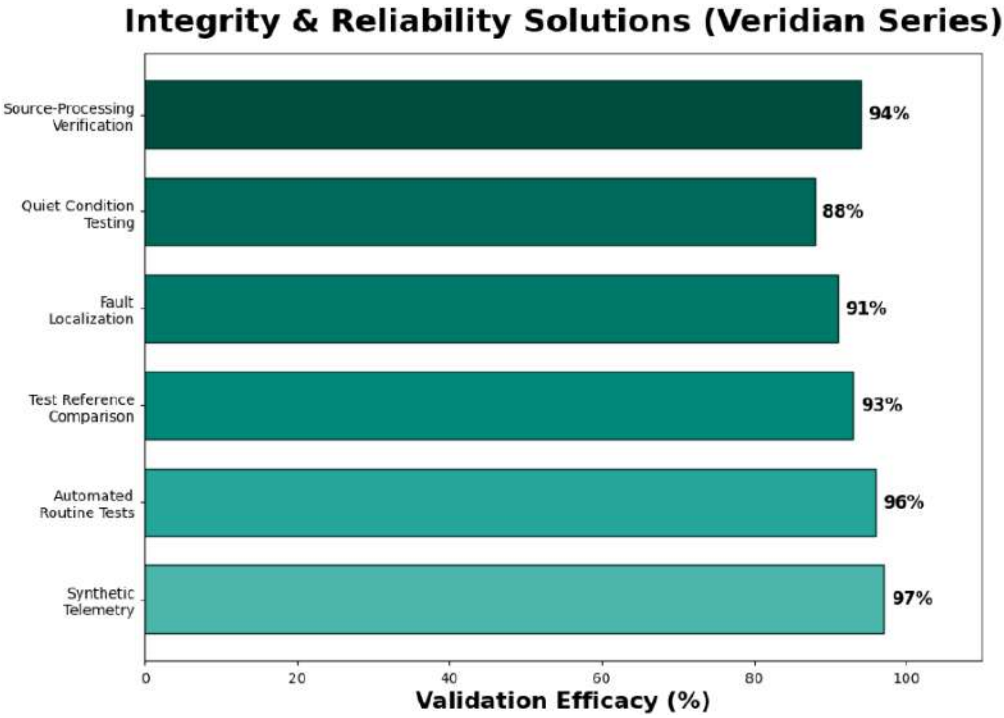

Fig 4.3: Integrity & Reliability Solutions (Veridian Series)

4.7.2. Interoperability testing

There are several challenges related to the validation of any streaming system in clinical environments. As the system is processing live clinical data, the connection to the data sources must be maintained and the possibility of playing back a recorded test data stream is impractical. To address this issue, automated synthetic telemetry streams can be generated to validate specific elements and expected behaviour during specific conditions. For example, a newly added equipment stream can be synthetically generated to ensure correct processing of a new data source. In parallel, the output can also be compared to additional data streams sourced from reliable test reference sources. Such comparisons can highlight potential faults that are not detected by the system itself and localise points of possible misbehaviour.

Even during routine operation, the validation of these systems, specifically their output, is essential. Malfunctioning source devices, streams being delivered out of specification and even neighbouring hospitals using incorrectly configured equipment can all feed the system with false data. Such events can lead to incorrect alerts and over-reliance on the streaming output. Building automatic validation tests into existing running conditions helps ensure both quiet and alert conditions are tested frequently.

4.8. Future Directions and Innovation

Artificial intelligence is being integrated into almost every field, and data analytics streaming systems are no exception. Although traditional machine learning models require earlier batches of training data, it is advisable to deploy them as soon as possible to gather more diverse patterns and fill uncertainty gaps. Sophisticated deep architectures also show great potential. Fundamental vision transformers require extreme amounts of data, while pseudo-Lidar transformers leverage implicit input into point clouds and achieve state-of-the-art performance on several tracking benchmarks. Nevertheless, user-defined algorithms and domain knowledge complement AI. Even in a totally unsupervised process, machine learning should still be considered "a research area that is still very active in the field of telemetry" as stated by the affinity group, which gathered over 100 experts from the telemetry and telecommunication disciplines.

A federated architecture is suitable for broad and dispersed applications, such as remote patient monitoring. Instead of centralizing information, only local features at edge and hub levels are transmitted upstream. Critical areas are coupled with local processing. By ensuring a straightforward relationship between receivers and senders, this event-driven architecture increases robustness and flexibility while overcoming latency issues. Redundant processing along streaming pipelines increases resilience. As each component has a controlled number of input and output events per unit of time, it becomes feasible to manually or automatically share current or future loads.

4.8.1. Artificial intelligence in streaming analytics

Machine learning and artificial intelligence (AI) techniques are gaining prominence in clinical data analysis for their ability to identify or predict events based solely on patterns in historical or near-real-time data of individuals or populations. In medical imaging, the high execution times of models, such as convolutional neural networks (CNN), necessitate adaptation to real-time decision paradigms. Often, these models are not integrated into clinical or telemetry systems but are implemented instead as isolated applications or web services that ingest the data, process it according to the model, and return an output item without user interaction. Clinical telemetry constitutes a proactive monitoring tool whose alerts warn of abnormal physiological conditions of patients. Telemetry systems continuously control parameters through graphical visualizations that incorporate historical, current, and predictive information. During the COVID-19 pandemic, several systems appeared in literature that monitor clusters of patients using these concepts.

These need not be limited to standard physiological conditions. Therefore, during patient admission, it becomes possible to execute a model that predicts clinical deterioration in a short period, such as septic shock. As the need for an alert arises, various alert mechanisms can be implemented, such as the standard telemetry alert or an additional layer that groups notifications of multiple patients whose estimated time to get in the state is about to arrive.

4.8.2. Federated and edge streaming architectures

Holistic AI systems require data from multiple sources and meet latency, security or privacy concerns. The architecture of federated learning distributes training to edge systems, while centralizing learning updates without sharing the complete datasets, to address issues with sensitive data. Federated learning aims to securely train AI models without revealing patient data or models trained at the different nodes. This is useful in many domains requiring privacy-preserving data sharing processes among untrusting parties, like the healthcare domain.

Federated learning creates cross-silo or cross-device settings. In cross-silo settings, organizations collaborate by sharing training results to create a shared model. Cross-device settings allow thousands or even millions of devices to participate in creating a shared model by uploading only control messages or parameters. Local model training on resource-constrained devices or at the edge contrasting with centralized training enables edge intelligence. Local model training and inference can be facilitated by edge devices, that enables automatic and personal service. Centralized orchestration allows the creation of an intelligent edge device serving for multiple persons.

4.9. Conclusion

Real-time Clinical Data Exchange and Streaming Systems provide an agile and efficient mechanism for deploying real-time services in healthcare, by enabling the seamless integration of streaming data from heterogeneous sources into automated decision-support systems. Such services have begun to include telemetry and remote patient monitoring systems, but additional capabilities are needed to support ongoing, data-rich clinical research in real-time settings while ensuring compliance with clinical data integrity policies. Latency, accuracy, and scalability remain key challenges, requiring considered annotations and innovative testing strategies. Furthermore, wider adoption will demand new capabilities, such as support for federated or edge-streaming architectures, and the ability to address AI training tasks with temporally-annotated streams.

Real-time systems present an agile and efficient mechanism for deploying real-time services in healthcare through the seamless integration of streaming data from heterogeneous sources into automated decision-support systems. Recent projects have demonstrated the feasibility of annotating streaming data with the information and precision required to support clinical decision-support services in live environments. Application examples include the integration of telemetry data from vehicles carrying patients needing urgent care, and a remote monitoring system based on a collection of simple yet effective rules, such as alarms triggered by symptoms or events outside normal ranges.

References

Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A distributed messaging system for log processing. https://doi.org/10.1145/1963405.1963412

Avinash Pamisetty, Vijaya Rama Raju Gottimukkala. (2024). Agentic AI-Driven Multi-Cloud Big Data Architecture For Predictive Demand, Credit Risk, And Inventory Financing In National Food Service Supply Chains. Metallurgical and Materials Engineering, 30(4), 959–975. Retrieved from https://metall-mater-eng.com/index.php/home/article/view/1933

Kreps, J., Narkhede, N., & Rao, J. (2023). Kafka: A distributed streaming platform. *LinkedIn Engineering*.

Aitha, A. R. (2021). Optimizing Data Warehousing for Large Scale Policy Management Using Advanced ETL Frameworks.

Miotto, R., Li, L., Kidd, B. A., & Dudley, J. T. (2016). *Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records*. Scientific Reports, 6, 26094. https://doi.org/10.1038/srep26094

Akidau, T., Chernyak, S., & Lax, R. (2018). Streaming Systems: The What, Where, When, and How of Large-Scale Data Processing. https://doi.org/10.5555/3288458

Thutari, R. T., Garapati, R. S., B M, Manjula., R K, Supriya., & M, Senbagan. (2025). Adaptive Access Control and Authentication Management for IoT Using Attention-GRU and Reinforcement Learning. In 2025 2nd International Conference on Software, Systems and Information Technology (SSITCON) (pp. 1–6). https://doi.org/10.1109/ssitcon66133.2025.11342003.

Vassiliadis, P. (2023). A survey of ETL and data pipeline systems. *ACM Computing Surveys*.

Nagabhyru, K. C., & Babu, A. J. Human In The Loop Generative AI: Redefining Collaborative Data Engineering For High Stakes Industries.

Vartak, M., Madden, S., Parameswaran, A., & Polyzotis, N. (2018). Model management systems: The next frontier of machine learning. IEEE Data Engineering Bulletin, 41(2), 13–26.

Amistapuram, K. (2024). Generative AI in Insurance: Automating Claims Documentation and Customer Communication. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 15(3), 461-475.

Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., et al. (2018). Accelerating the machine learning lifecycle with MLflow. IEEE Data Engineering Bulletin, 41(4), 39–45.

Sudhakar, A. V. V., Inala, R., Verma, A. K., Nag, K., Pandey, V., & Anand, P. S. (2025). Hybrid Rule-Based and Machine Learning Framework for Embedding Anti-Discrimination Law in Automated Decision Systems. In 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT) (pp. 1–6). https://doi.org/10.1109/icicnct66124.2025.11232861.

Karau, H., & Warren, R. (2017). High performance Spark: Best practices for scaling and optimizing Apache Spark. O'Reilly Media.

Nandan, B. P., & Chitta, S. S. (2023). Machine Learning Driven Metrology and Defect Detection in Extreme Ultraviolet (EUV) Lithography: A Paradigm Shift in Semiconductor Manufacturing. Educational Administration: Theory and Practice, 29(4), 4555-4568.

Januschowski, T., Bohlke-Schneider, M., Wang, S., Salinas, D., et al. (2021). Criteria for classifying forecasting methods. International Journal of Forecasting, 36(1), 167–177.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), Article 44.

Davuluri, P. N. Integrating Artificial Intelligence into Event-Driven Financial Crime Compliance Platforms.

Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., & Petitjean, F. (2016). Characterizing concept drift. Data Mining and Knowledge Discovery, 30(4), 964–994.

Nagubandi, A. R. (2025). Cryptocurrency Market Spillovers: Risk Contagion Across Global Financial Systems.

Baier, L., Kühl, N., & Satzger, G. (2020). How to cope with change? Preserving validity of predictive services over time. In Proceedings of the 53rd Hawaii International Conference on System Sciences (pp. 1381–1390).

Yandamuri, U. S. (2026). Scalable Cloud-Based Intelligent Decision Systems Leveraging AI and Big Data for Industry-Specific Optimization. Minnesota Journal of Business Law and Entrepreneurship, (1), 584-601.

Rabanser, S., Günnemann, S., & Lipton, Z. (2019). Failing loudly: An empirical study of methods for detecting dataset shift. In Advances in Neural Information Processing Systems, 32, 1396–1408.

Velangani Divya Vardhan Kumar Bandi. (2024). Intelligent Data Platforms For Personalized Retail Analytics At Scale. Metallurgical and Materials Engineering, 30(4), 1011–1027. Retrieved from https://metall-mater-eng.com/index.php/home/article/view/1011-1027

Lipton, Z. C., Wang, Y.-X., & Smola, A. J. (2018). Detecting and correcting for label shift with black box predictors. In Proceedings of the 35th International Conference on Machine Learning (pp. 3122–3130).

Suresh, H., & Guttag, J. V. (2021). A framework for understanding unintended consequences of machine learning. arXiv preprint arXiv:1901.10002.

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., et al. (2019). Model cards for model reporting. In Proceedings of the Conference on Fairness, Accountability, and Transparency (pp. 220–229).

Mandel, J. C., Kreda, D. A., Mandl, K. D., Kohane, I. S., & Ramoni, R. B. (2016). SMART on FHIR: A standards-based, interoperable apps platform for electronic health records. https://doi.org/10.1016/j.jbi.2016.03.004