UDC 614.4:004.8:004.6 **doi:** https://doi.org/10.20998/2522-9052.2025.4.08

Yurii Parfeniuk[1], Kseniia Bazilevych[2], Ievgen Meniailov[1], Dmytro Chumachenko[2, 3]

[1] V.N. Karazin Kharkiv National University, Kharkiv, Ukraine
[2] National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine
[3] Balsillie School of International Affairs, Waterloo, ON, Canada

A MULTI-LAYER DELTA LAKEHOUSE FOR EPIDEMIOLOGICAL MONITORING AND FORECASTING UNDER EMERGENCIES

Abstract. Public health emergencies demand fast, dependable analytics that combine real-time signals with trustworthy historical data. Open, interoperable platforms that support streaming and batch workflows can shorten the time from detection to action while preserving data quality and auditability. **Aim:** To design and justify an information system architecture for analyzing epidemic threats under emergency conditions that is scalable, reliable, and fit for integration with clinical and non-traditional data sources. **Methods:** We conducted a structured review of three data analytics architectures (Lambda, Kappa, Delta) and mapped their strengths and limits to crisis surveillance needs. Based on functional and non-functional requirements, we specified a Delta Lake–based lakehouse with bronze-silver-gold tiers, unified batch/stream ingestion with Spark Structured Streaming, ACID tables with time travel and schema control, and an analytics layer that supports forecasting with MLOps for monitoring, drift checks, retraining, and lineage. **Results:** The proposed architecture meets core emergency needs for timeliness, integrity, and reproducibility through ACID transactions, versioned datasets, and curated tiers; supports standards-based interoperability and the inclusion of wastewater, mobility, and other environmental feeds; provides a single code path for batch and streaming to reduce reconciliation burden; and sets operational guardrails for latency versus cost when running many near-real-time tables. We outline practical considerations for quality checks in the silver tier, promotion rules to gold, and model governance. **Conclusions:** A Delta-based lakehouse offers a clear path to an emergency-ready surveillance platform that scales with data growth, integrates heterogeneous sources, and supports reliable forecasting. The next steps are a pilot deployment with public health partners, live latency and cost measurements, and prospective validation of forecasting and alerting in real-world settings.

Keywords: epidemic surveillance; outbreak analytics; Lakehouse; Delta lake; machine learning.

Introduction

In the context of globalization and the rising threat of infectious disease outbreaks, timely analysis and processing of epidemiological surveillance data are critical, particularly during emergencies. Early detection, assessment, and response to these threats are essential to prevent large-scale crises, protect population health, and reduce socioeconomic impacts [1].

The relevance of advanced information systems is especially pronounced during emergencies, when surveillance must pivot from periodic reporting to continuous, decision-grade intelligence. The World Health Organization's Early Warning, Alert and Response (EWAR) guidance and its "EWARS in a box" toolkit illustrate how rapidly deployable platforms shorten the interval from signal detection to public health action, including in conflict and post-disaster settings [2]. Deployments can be configured within days to restore core outbreak detection and alerting capacity when routine systems are disrupted.

Open-source epidemic intelligence further increases situational awareness when formal reporting is delayed. WHO's Epidemic Intelligence from Open Sources (EIOS) and regional implementations have shown that triaging large volumes of media and nontraditional data can surface actionable signals early [3]. A recent evaluation in the WHO African Region reported measurable performance in early detection [4]. These results support integrating EIOS-style pipelines into emergency operations centers to complement indicator-based surveillance.

Environmental ("wastewater-based") surveillance has matured into a practical early-warning layer well suited to emergencies, mass gatherings, and settings with constrained clinical testing. Syntheses from 2024–2025 show wastewater measurements can anticipate community transmission, including detection among asymptomatic populations, and guide targeted interventions [5]. National academies and systematic reviews recommend institutionalizing wastewater surveillance for both endemic and emerging pathogens [6].

Interoperable data exchange is critical for surge decision-making. Recent public-health informatics work highlights HL7 FHIR–based specifications (e.g., SANER and the US Situational Awareness Framework for Reporting) that automate hospital capacity, respiratory disease metrics, and other feeds to give authorities near-real-time dashboards [7]. Peer-reviewed studies and federal progress reports indicate growing production use of FHIR (including Bulk FHIR for population-level extracts), with more states automating bed-capacity and respiratory reporting as part of data-modernization initiatives [8]. These capabilities are directly relevant during acute events.

Dedicated early-warning networks remain a backbone of outbreak control in humanitarian and conflict emergencies. Recent analyses of Syria's early-warning systems and digital health assessments from Yemen emphasize that lightweight, resilient architectures and streamlined reporting and alert thresholds sustain detection and response despite infrastructure damage and workforce constraints [9, 10]. These findings reinforce the value of deployable, low-bandwidth solutions that preserve timeliness under adverse conditions.

Multi-sector "One Health" integration strengthens emergency readiness by linking human, animal, and environmental data, which is essential given the

zoonotic profile of many emerging threats [11]. Recent reviews and frameworks document timeliness gains and practical pathways for data sharing across sectors, and call for embedding One Health analytics within routine and emergency surveillance architectures to manage cross-border and climate-sensitive risks [12].

Effective epidemic control requires rapid response and predictive models built on in-depth analysis of large volumes of heterogeneous data using machine learning and artificial intelligence [13]. There is a need to develop systems capable of real-time data processing [14]. Both accuracy and processing speed are critical for timely action. Historical data must be incorporated to improve forecasting accuracy, which requires the system to handle both streaming data arriving in real time and large stores of historical data [15].

A literature review shows that such information systems are widely used not only in the Big Data industry but also across the broader economy and commerce and in many public sector organizations.

An information system for epidemiological data analysis and processing must meet requirements that can be grouped into several categories.

1. Functional requirements:

• Data collection from multiple sources: the system must integrate with diverse data sources, including laboratory information systems (LIS), electronic medical record (EMR) systems, disease surveillance systems, hospitalization databases, mortality data, and other relevant sources [16]. This requires support for various data formats and exchange protocols (e.g., HL7, FHIR).

• Data storage and management: the system must provide reliable storage for large volumes of structured and unstructured data (text descriptions, medical images), effective version control, and assurance of data integrity. The system must be scalable to handle growing data volumes.

• Data processing and analysis: the system must offer tools for data cleaning, transformation, and aggregation, statistical analysis, construction of epidemiological models, and application of machine learning methods for forecasting and outbreak detection.

• Data visualization: the system must provide tools to visualize data in multiple formats (charts, maps, tables) to support interpreting analytical results. Visualizations should be interactive and configurable.

• Reporting and alerts: the system must generate reports on user requests and automatically issue alerts about potential outbreaks based on predefined thresholds.

• User and access management: the system must enforce role-based access to data and system functions, ensuring data confidentiality and security.

2. Non-functional requirements:

• Reliability and availability: the system must be reliable and available 24/7. This may require data redundancy and a fault-tolerant architecture [14].

• Security: the system must ensure the confidentiality, integrity, and availability of data in line with personal data protection laws (e.g., GDPR, HIPAA). Protection against unauthorized access, modification, and data destruction must be provided.

• Scalability: the system must handle growing data volumes and increase performance.

• Performance: the system must support fast query processing and report generation.

• Usability: the system must be intuitive and convenient for healthcare professionals and epidemiologists with varying levels of technical training.

• Integration: the system must integrate smoothly with existing healthcare information systems.

• Support: technical support and user training must be provided.

Thus, the main aim of this study is to develop an architecture for an information system to analyze epidemic threats under emergency conditions.

The current research is part of a comprehensive information system for assessing the impact of emergencies on the spread of infectious diseases [17].

1 Architectures Review

To implement information systems of this kind, the first task is to choose the data processing system's architecture. The most appropriate types are the Lambda, Kappa, and Delta architectures [18].

Each of these architectures has features that must be considered when choosing. The system should be planned for the steady growth in the volume and data types to be processed. The costs of building and maintaining such systems are also important factors.

We consider several approaches to data processing.

The Lambda architecture is a data-processing system composed of two pipelines [19]. The first is a traditional batch pipeline for accurate processing of historical (batch) data, and the second is a streaming pipeline that can process data quickly in real time. A system of this kind includes three layers:

- Batch layer: responsible for batch processing of data.
- Speed layer: responsible for real-time data processing.
- Serving layer: responsible for handling queries and returning results.

A structural diagram of data processing based on the Lambda architecture is shown in Fig. 1.

Despite combining batch and stream processing methods, implementing this architecture entails several challenges:

- high maintenance and support costs;
- the need to develop two pipelines separately;
- data-reconciliation difficulties due to different computation engines;
- different storage formats for streaming and batch processing.

The most significant drawbacks for this study are batch processing latency and two separate data processing pipelines. Because streaming outputs may be approximate, they are refined using results produced by batch processing. When real-time monitoring is required, this delay can become a serious issue, and its mitigation depends directly on data volume and the resources available for processing.

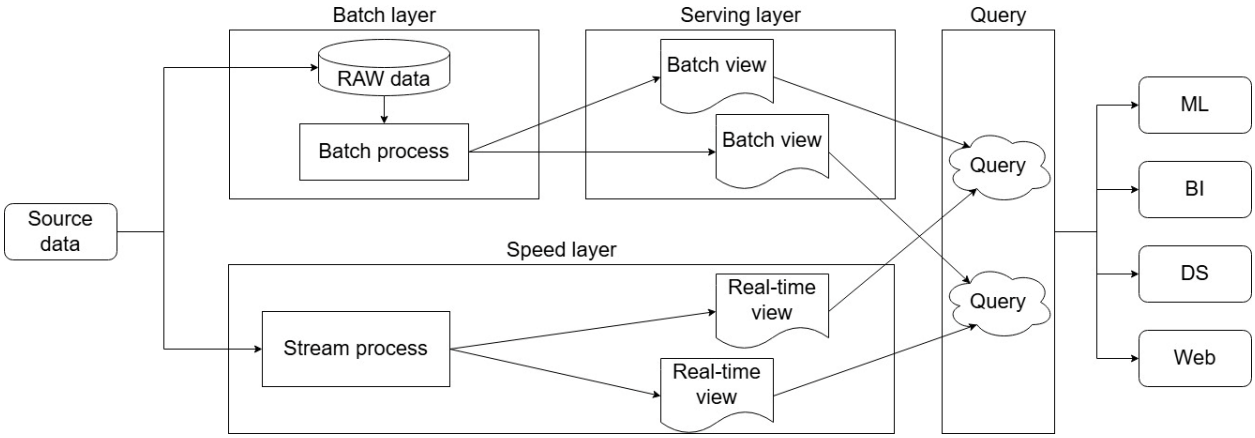

Fig. 1. Structural diagram of data processing based on the Lambda architecture

Maintaining two separate, and, crucially, different, pipelines also imposes a strict requirement for data consistency, which, given their different origins and processing logic, can create substantial operational and maintenance complexity.

The Kappa architecture avoids the main drawback of the Lambda approach, namely, the existence of two data processing systems and the need to support them separately [20]. In Kappa, stream analytics is performed within a dedicated stream-processing system, and the key distinction from Lambda is the absence of a separate batch analytics system. All computations are executed in the streaming system. Under this design, historical (batch) data are sent from storage to a streaming bus and consumed by the stream-processing analytics engine.

A structural diagram of data processing based on the Kappa architecture is shown in Fig. 2.

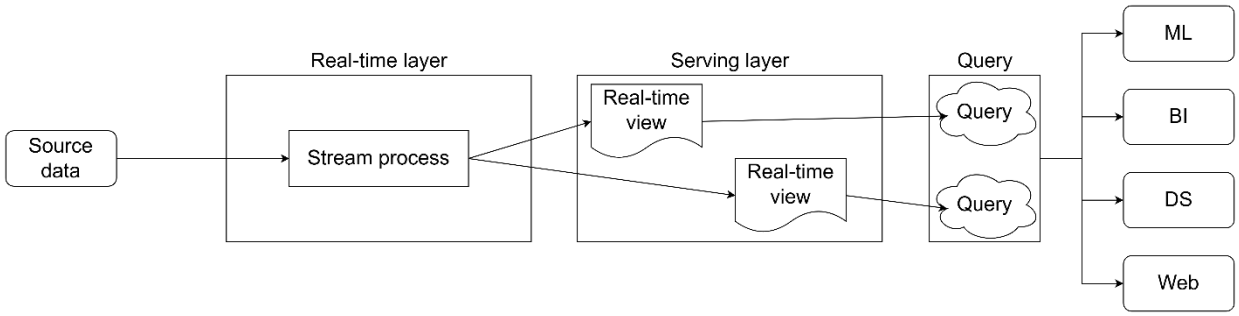

Fig. 2. Structural diagram of data processing based on the Kappa architecture

This approach eliminates most drawbacks inherent in the Lambda architecture because a single system performs both streaming and analytical processing. There is no need to maintain two different systems, and the problem of data inconsistency does not arise.

However, a limitation of this architecture is the need to pass analytical data for processing through the streaming data bus. This requires copying all analytical data and loading it into the event-stream bus. Given the large volumes of these data, this can lead to noticeable delays, inefficient resource use, and increased complexity in administering the streaming bus under high data throughput.

The limitations of the Lambda and Kappa architectures motivated the development of a new type, the Delta architecture [21]. Like Kappa, it unifies batch and streaming data in a single processing pipeline with one codebase. The Delta architecture can be viewed as a lightweight evolution of the Kappa model without the drawbacks of the Lambda approach. It was designed to avoid synchronization constraints in existing solutions and, importantly, enable on-the-fly data enrichment during processing. The initial goal was to reduce data processing complexity for application developers by providing ready-to-use outputs.

A structural diagram of data processing based on the Delta architecture is shown in Fig. 3.

The Delta architecture differs in structure from the architectures discussed above. As data enters the system, they are gradually sorted and enriched.

Standard query mechanisms and processing methods can be applied at any layer to move data between layers. The core idea is to distinguish data by quality and to build higher-level datasets by cleaning, enriching, and aggregating data from lower levels.

The Delta architecture divides data work into three storage tiers: bronze, silver, and gold. These conceptual, logical layers help classify data maturity and readiness for querying and processing.

Bronze tables ingest raw data and serve as the entry point for subsequent loading into data lake storage. Data are accepted in their original form and format, then converted to Apache Parquet for processing.

After initial processing, the system routes the data to the next layer using Apache Spark.

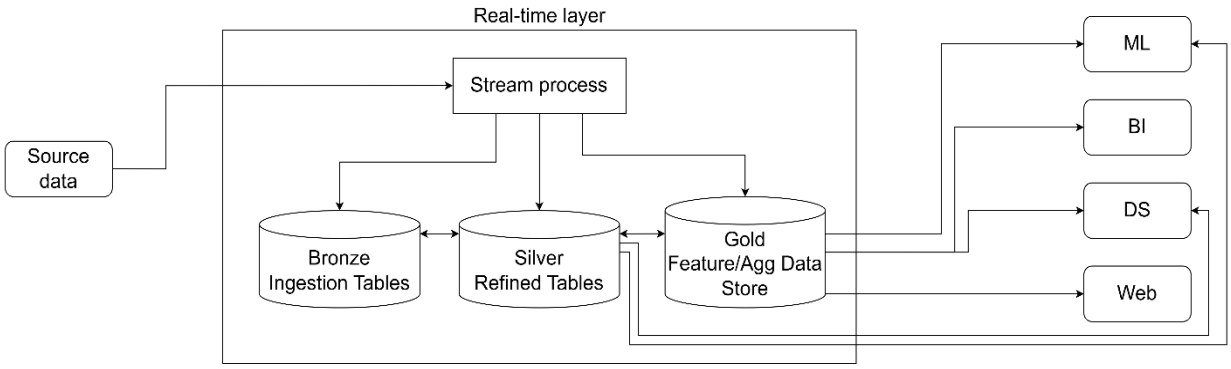

Fig. 3. Structural diagram of data processing based on the Delta architecture

Silver tables store data in an optimized state, which makes them usable for business analytics and data processing.

Raw inputs from the previous layer are filtered, cleaned, transformed, joined, and aggregated into curated silver datasets.

Where applicable, Delta Engine can be used as a consistent compute module when Azure Databricks is the base service for these tasks. For further analysis, suitable programming languages, such as SQL, Python, R, or Scala, may be used. DevOps processes and temporary compute clusters can be employed for specific jobs.

Gold tables contain structured and enriched datasets ready for analytics or reporting. For data analysis, one may use the most suitable language or toolset, such as Koalas, SQL, Power BI, Excel, or many other tools.

One of the most common implementation options relies on open-source software. Delta Lake is an open-source storage layer that brings reliability to data lakes [22], which is crucial when data volumes are large and update/processing rates are high. Delta Lake supports ACID transactions [23], scales metadata handling, and unifies streaming and batch data during processing. Delta Lake runs on top of an existing data lake (Apache Hadoop HDFS, Amazon S3, or Azure Data Lake Storage) and is fully compatible with all Apache Spark APIs.

Performance limits for compute operations on HDFS are only one reason Delta Lake emerged. In principle, batch and stream processing can be built within a Lambda architecture. Additional drivers for the rise and evolution of Delta Lake include the digitalization of the economy and government processes, the spread of hybrid and cloud data warehouses (DWH), and the service-oriented model for Big Data technologies. Using Apache Hadoop (Hadoop-as-a-Service) as an example, convenience comes with challenges:

- some core Hadoop services (e.g., YARN, HDFS) were not originally designed for cloud use, having been built for on-premises environments with specific architectural traits and constraints;

- maintenance and operations demand significant time and resources.

These issues become more pressing as workloads move to the cloud and system requirements change over time. It is therefore desirable that such systems provide:

- Big Data clusters that are easy to use and available on demand as PaaS/SaaS services;

- elastic scaling with cost-of-ownership control and strong site reliability (SRE) practices, ideally with clear SLAs;

- high data quality, and thus trustworthy data lakes, for sound, analytics-driven decisions;

- high processing speed at a very large scale;

- reconfigurable cloud services to match changing tasks, without vendor lock-in;

- a simple GUI so even less experienced users can configure and use cloud services.

With this design, leveraging cloud technologies and high throughput from Apache Spark, Delta Lake offers the following advantages [24]:

- ACID transactions. Typical data lakes run multiple pipelines that read and write concurrently, forcing engineers to ensure integrity without transactional support. Delta Lake brings ACID transactions to data lakes, delivering serializability and strong isolation [23].

- Scalable metadata processing. Given that Big Data's scale can be large, Delta Lake uses Apache Spark's distributed computing to process metadata, enabling work with extremely large tables [25].

- Data versioning and management. Delta Lake provides dataset snapshots, allowing access to and rollback to earlier versions for audit, recovery, or experiment reproduction.

- Data format. All data are stored in the columnar Apache Parquet format, which enables efficient compression and encoding.

- Unified batch/stream source and sink. A Delta Lake table acts as a batch table, streaming source, and sink. Streaming ingestion, batch processing of historical data, and interactive queries work immediately after deployment.

- Schema enforcement. The system enforces declared structures, ensuring correct data types and required columns, preventing corruption from malformed inputs.

- Schema evolution. Table schemas can change automatically without heavy DDL.

- Change-history auditing. A transaction log records every data change.

- Updates and deletes. APIs in Scala/Java support update/delete operations, simplifying change-data capture and compliance with GDPR and CCPA.

- Full Spark API compatibility. Existing Apache Spark data pipelines can run on Delta Lake with minimal changes.

This architecture and its supporting tools are widely used by Big Data companies worldwide to power large-scale analytics with data science and machine learning methods.

2 Proposed Architecture

Thus, an information system for analyzing epidemic threats in emergencies, built on the Delta architecture, should be a multi-layer system that ensures reliable, scalable, and efficient data management and analytical model development. It combines the advantages of different data storage and processing approaches to address epidemiological monitoring and forecasting tasks optimally.

An example architecture of an information system for analyzing epidemic threats in emergencies is shown in Fig. 4.

The system consists of the following key components:

- Data sources: diverse inputs, including laboratory information systems (LIS), electronic medical record (EMR) systems, disease surveillance systems, hospitalization databases, mortality records, satellite imagery, population mobility data, etc. Data may arrive in both structured and unstructured formats.

- Data preparation and transformation layer: this layer converts raw data from the data lake into a structured format suitable for analysis. ETL/ELT processes (Extract–Transform–Load / Extract–Load–Transform) and big-data tools (e.g., Spark) are used for cleaning, transformation, aggregation, and enrichment. The output is a processed, structured dataset stored in Delta Lake.

- Delta Lake: the system's core, providing reliable, ACID-compliant storage using the open Parquet standard. Delta Lake guarantees data consistency, supports change tracking, and enables batch and streaming modes.

- Analytics and forecasting layer: this layer holds aggregated and prepared data for specific analytical tasks. It may include summary tables and model-ready datasets for machine learning. Data can be stored in a data warehouse (e.g., Snowflake, BigQuery) or in databases optimized for analytics.

- Machine learning block: this layer contains trained models for forecasting disease spread, identifying risk factors, and detecting outbreaks. It may include regression models, time-series models, and neural networks. The model lifecycle is managed with MLOps tools.

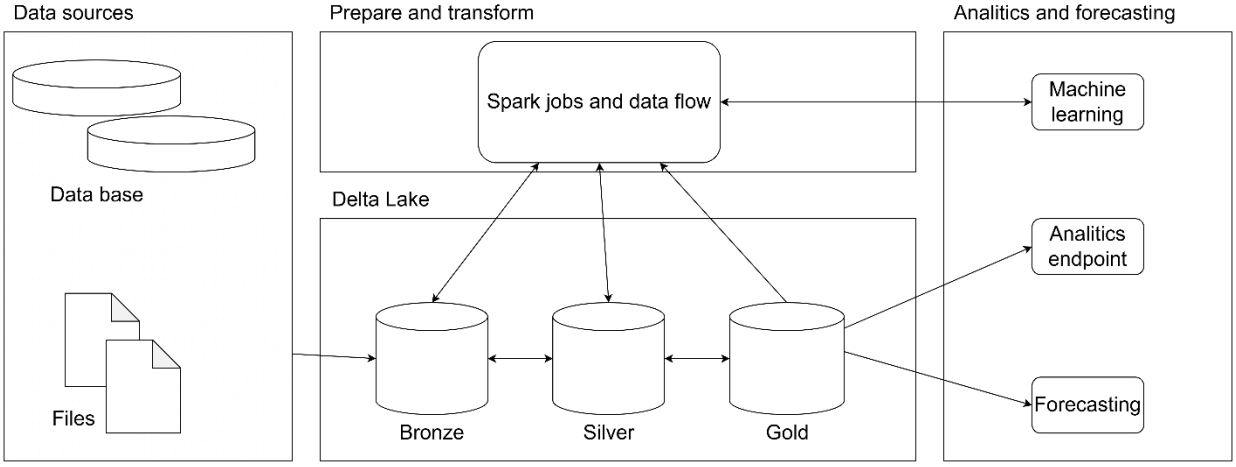

Fig. 4. Architecture of an information system for analyzing epidemic threats in emergencies

This architecture enables a modern, flexible epidemiological monitoring and forecasting information system that can handle growing data volumes and complex analytical tasks. It supports comprehensive data analysis, accurate predictive modeling, and evidence-based decision making to prevent and control infectious disease outbreaks.

It should be noted that the practical implementation of the Delta architecture is not trivial, even with powerful compute resources. In particular, Apache Spark Structured Streaming supports batch and streaming by splitting incoming data into configurable micro-batches that can be processed via the DataFrame and Dataset APIs. However, one must balance low latency with cost efficiency when operating many near-real-time tables. Accordingly, when designing a Lakehouse with a Delta architecture, the following requirements should be defined:

- the maximum permissible processing latency for each batch and streaming job;

- the execution cadence of batch jobs and the data volume they handle;

- the number of structured streams that should run in parallel within a single Apache Spark cluster to keep datasets current and to execute both batch and streaming workloads.

3 Discussion

The paper proposes a Delta Lake-based, multi-layer "lakehouse" design to support near-real-time

epidemic intelligence. The design includes unified batch and streaming ingestion, ACID guarantees, versioning, and an MLOps layer for forecasting. The approach is consistent with recent evidence that lakehouse systems can combine scalable analytics with reproducible data management and audit trails, which are essential during emergencies.

Choosing Delta Lake as the storage layer is defensible from both performance and governance standpoints [23]. Empirical studies of Delta Lake show that its transaction log and Parquet-based metadata compaction provide ACID properties, time travel, and fast operations at large table scales, capabilities that reduce data corruption risk during periods of high ingest and rapid schema change typical of emergencies [25]. The broader lakehouse model demonstrates competitive analytics performance while keeping data in open formats, easing integration with diverse sources that public health relies on.

Interoperability at the edge of the system is equally important. The plan to integrate LIS/EMR and other clinical systems aligns with the growing production use of HL7 FHIR, including Bulk FHIR for population-level extracts, which has been shown to enable "push-button population health" and is now widely implemented [26]. These standards can shorten data latency and reduce custom interfaces during surges [27].

Nontraditional sources (e.g., wastewater, mobility, satellite) are well supported. Reviews from 2023-2025 report that wastewater-based surveillance can anticipate community transmission and support targeted response when clinical testing lags, reinforcing the value of treating environmental feeds as first-class inputs in the pipeline [28, 29].

Emergency decisions are sensitive to data quality issues (completeness, timeliness, validity). The paper's emphasis on schema enforcement, schema evolution, and versioning (snapshots) matches best practice: recent reviews of EHR and health information data quality highlight standardization, provenance, and auditable change logs as primary levers to mitigate bias and errors [30]. Delta Lake's ACID semantics and time-travel address these needs by design. Adding routine quality checks at the silver tier (e.g., rule-based tests) would further strengthen the pipeline [31].

The bronze-silver-gold tiering also supports reproducible analytics. By freezing curated "silver" datasets and promoting only validated aggregates to "gold," teams can rerun models against known snapshots and compare outputs across deployments, an approach recommended in recent public health informatics and surveillance reviews [32].

Operational value in emergencies depends on reducing the reporting delay between signal and decision. Studies in 2024-2025 show that nowcasting can shorten detection lags by days to weeks across different pathogens and settings [33]. A streaming feature pipeline that computes delay-aware indicators (e.g., right-truncated counts, wastewater load, ED visit signals) and writes them to "gold" tables for dashboards aligns with these findings and should be prioritized in the analytics layer [34].

The proposed architecture relies on Spark Structured Streaming (micro-batch). While micro-batching is often sufficient for public health cadence (minutes to hours), teams should validate end-to-end latency against use cases that need tighter loops (e.g., facility capacity reporting) and consider sources that support true event processing where required. This trade-off is common in lakehouse deployments and should be documented in runbooks.

Maintaining calibrated models during an emergency is a nontrivial challenge. Recent work in healthcare MLOps underscores continuous monitoring, drift detection, automated retraining, and governance (model cards, audit logs) as core practices for safe deployment [35]. Complementary studies show that distribution shift and concept drift are frequent during outbreaks and that explicit drift detection improves reliability [36]. The proposed MLOps block should include: monitored performance metrics with alerting; drift tests at input and prediction levels; scheduled or triggered retraining with data snapshots; and a registry with lineage to the exact silver snapshot, code commit, and hyperparameters.

Ensembles and post-processing have recently been shown to improve nowcast accuracy and calibration in forecasting workflows [37]. Designing the analytics tier to support model ensembling and statistical post-processing (e.g., bias correction, uncertainty quantification) will likely yield more stable operational signals.

The system will manage identifiable health data during periods of heightened scrutiny. Reviews on implementing ML in healthcare stress that technical controls must be paired with organizational policies to ensure equity and accountability (access control, PHI minimization, robust audit) [38]. Where feasible, integrating de-identification and privacy-preserving analysis for secondary analytics (e.g., training with de-identified or synthetic cohorts) can reduce risk without blocking urgent operational use cases.

The proposed data source list can be expanded to support One Health operations. Recent reviews document practical gains in timeliness and coverage when human, animal, and environmental data are integrated under a shared framework [39]. Building interfaces that can ingest veterinary and environmental indicators using common schemas will make the platform more useful for zoonotic threats.

Experience from lakehouse evaluations suggests that the combination of open columnar formats and ACID tables scales well for public sector workloads while preserving flexibility [40]. Still, teams should plan for capacity bursts during major events, backfills that recompute silver/gold tables from bronze with reproducible code, and cost controls for always-on streaming jobs. These points align with recent comparative studies of lakehouse storage systems and should be reflected in SRE playbooks.

Two limitations are clear. First, data representativeness remains a risk: EHR, syndromic, and environmental signals each have biases that can lead to drift or spurious correlations, and mitigation requires

data quality monitoring and domain-informed feature design. Second, the micro-batch streaming model may not meet the lowest-latency requirements for certain facility-level metrics, and selective use of event streaming or vendor systems may be needed.

Overall, the architecture advanced in the paper is well aligned with current evidence on scalable, auditable analytics for public health emergencies: a lakehouse foundation (for integrity and reproducibility), standards-based interoperability (for speed), and an MLOps-enabled analytics tier (for adaptive forecasting).

Conclusions

The study analyzed existing information system architectures used in data analytics and commerce, identifying three main types: Lambda, Kappa, and Delta. The structure of each architecture was examined in detail, and the suitability of each type, along with specific implementation tools, was assessed in light of the study's objectives.

The advantages of the Delta architecture include the following: systems built on this architecture scale readily to handle large data volumes; Delta Lake provides ACID transaction properties, ensuring data reliability and integrity; a range of analytical methods is supported; data processing workflows can be optimized; and Delta Lake enables data versioning, allowing changes to be tracked and earlier versions to be restored. The practical challenges of implementing systems based on this architecture are also described.

An architecture for an information system aimed at analyzing epidemic threats in emergencies was developed. The proposed design effectively integrates diverse data sources, enabling comprehensive analysis of the epidemiological situation under resource and time constraints. Its modular structure allows adaptation to different types of emergencies and specific needs. Further research is needed to validate the system in real-world conditions and to optimize individual components. In particular, a deeper analysis of forecasting algorithms and the development of more robust methods for data errors are required. Nevertheless, the architecture presented offers a promising approach to managing epidemic threats in emergencies and can improve the effectiveness of prevention and response efforts.

Acknowledgements

This study was funded by the National Research Foundation of Ukraine in the framework of the research project 2023.03/0197 on the topic "Multidisciplinary study of the impact of emergency situations on the infectious diseases spreading to support management decision making in the field of population biosafety".

REFERENCES

1. Dotsenko, N., Chumachenko, I., Kraivskyi, B., Railian, M. and Litvinov, A. (2024), "Methodological Support for Managing of Critical Competences in Agile Transformation Projects within a Multi-Project Medical Environment", *Advanced Information Systems*, vol. 8, no. 4, pp. 26–33, doi: https://doi.org/10.20998/2522-9052.2024.4.04
2. (2023), *Early Warning Alert and Response (EWAR) in Emergencies: An Operational Guide*, World Health Organization, available at: https://www.who.int/publications/i/item/9789240063587
3. (2025), *The Epidemic Intelligence from Open Sources Initiative*, World Health Organization, available at: https://www.who.int/initiatives/eios
4. Williams, G.S., Koua, E.L., Abdelmalik, P., Kambale, F., Kibangou, E., Nguna, J., Okot, C., Akpan, G., Moussana, F., Kimenyi, J.P. and Gueye, A. S. (2025), "Evaluation of the Epidemic Intelligence from Open Sources (EIOS) System for the Early Detection of Outbreaks and Health Emergencies in the African Region", *BMC Public Health*, vol. 25, 857, doi: https://doi.org/10.1186/s12889-025-21998-9
5. Singh, S., Ahmed, A.I., Almansoori, S., Alameri, S., Adlan, A., Odivilas, G., Chattaway, M.A., Salem, S.B., Brudecki, G. and Elamin, W. (2024), "A Narrative Review of Wastewater Surveillance: Pathogens of Concern, Applications, Detection Methods, and Challenges", *Frontiers in Public Health*, vol. 12, 1445961, doi: https://doi.org/10.3389/fpubh.2024.1445961
6. (2025), *Wastewater Surveillance for Emerging Pathogen Threats*, National Academies of Sciences, Engineering, and Medicine, available at: https://www.ncbi.nlm.nih.gov/books/NBK610710/
7. (2025), *Public Health US Situational Awareness Framework for Reporting Home - US Situational Awareness Framework for Reporting (US SAFR) Implementation Guide*, V1.0.0, HL7 International, available at: https://build.fhir.org/ig/HL7/us-safr/
8. Essaid, S., Andre, J., Brooks, I.M., Hohman, K.H., Hull, M., Jackson, S.L., Kahn, M.G., Kraus, E.M., Mandadi, N., Martinez, A.K. and Soares A. (2024), "MENDS-On-FHIR: Leveraging the OMOP Common Data Model and FHIR Standards for National Chronic Disease Surveillance", *JAMIA Open*, vol. 7, doi: https://doi.org/10.1093/jamiaopen/ooae045
9. Alhaffar, B.A., Abbara, A., Almhawish, N., Tarnas, M.C., AlFaruh, Y. and Eriksson, A. "The Early Warning and Response Systems in Syria: A Functionality and Alert Threshold Assessment", *IJID Regions*, vol. 14, article number 100563, doi: https://doi.org/10.1016/j.ijregi.2024.100563
10. Alhammadi, O.A.S., Mohamed, H.I., Musa, S.S., Ahmed, M.M., Lemma, M.A., Joselyne, U., Roméo, B., Abdullahi, Y., Othman, Z.K., Hamid, M.R. and Okesanya O.J. (2024), "Advancing Digital Health in Yemen: Challenges, Opportunities, and Way Forward", *Exploration of Digital Health Technologies*, vol. 2, pp. 369–386, doi: https://doi.org/10.37349/edht.2024.00035
11. Fieldhouse, J.K., Nakiire, L., Kayiwa, J., Mirzazadeh, A., Brindis, C.D., Mitchell, A., Sepulveda, J., Makumbi, I., Ario, A.R., Fair, E. and Lamorde M. (2025), "An Analysis of One Health Timeliness Metrics across Multisectoral Public Health Emergencies in Uganda", *Communications Medicine*, vol. 5, 192, doi: https://doi.org/10.1038/s43856-025-00893-9
12. Brown, H.L., Pursley, I.G., Horton, D.L. and La, R.M. (2024), "One Health: A Structured Review and Commentary on Trends and Themes", *One Health Outlook*, vol. 6, article number 17, doi: https://doi.org/10.1186/s42522-024-00111-x
13. Chen, B., Zhu, L., Da, W. and Cheng, J. (2021), "Research on the Design of Mass Recommendation System Based on Lambda Architecture", *Journal of Web Engineering*, vol. 20, pp. 1971–1990, doi: https://doi.org/10.13052/jwe1540-9589.20614

14. Daki, H., El Hannani, A. and Ouahmane, H. (2020), "Big Data Architectures Benchmark for Forecasting Electricity Consumption", *2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech)*, Marrakesh, Morocco, pp. 1–6, doi: https://doi.org/10.1109/cloudtech49835.2020.9365912
15. Izonin, I., Tkachenko, R., Berezsky, O., Krak, I., Kováč, M. and Fedorchuk, M. (2024), "Improvement of the ANN-Based Prediction Technology for Extremely Small Biomedical Data Analysis", *Technologies*, vol. 12, article number 112, doi: https://doi.org/10.3390/technologies12070112
16. Farki, A., Noughabi, E.A. (2023), "Real-Time Blood Pressure Prediction Using Apache Spark and Kafka Machine Learning", *2023 9th Int. Conf. on Web Research*, pp. 161–166, doi: https://doi.org/10.1109/icwr57742.2023.10138962
17. Chumachenko, D., Bazilevych, K., Butkevych, M., Meniailov, I., Parfeniuk, Y., Sidenko, I. and Chumachenko, T. (2024), "Methodology for Assessing the Impact of Emergencies on the Spread of Infectious Diseases" *Radioelectronic and Computer Systems*, vol. 3, pp. 6–26, doi: https://doi.org/10.32620/reks.2024.3.01
18. Bazilevych, K., Kyrylenko, O., Parfeniuk, Y. and Meniailov, I. (2025), "Emerging Technologies in Infectious Disease Surveillance and Control: Current Solutions and Future Directions", *Lecture notes in networks and systems*, article number 1473, pp. 196–207, doi: https://doi.org/10.1007/978-3-031-94845-9_17
19. Cerezo, F., Cuesta, C.E., Moreno-Herranz, J.C. and Vela, B. (2019), "Deconstructing the Lambda Architecture: An Experience Report", *Proceedings - 2019 IEEE International Conference on Software Architecture - Companion, ICSA-C 2019*, pp. 196–201, doi: https://doi.org/10.1109/icsa-c.2019.00042
20. Nkamla Penka, J.B., Mahmoudi, S. and Debauche, O. (2021), "A New Kappa Architecture for IoT Data Management in Smart Farming", *Procedia Computer Science*, vol. 191, pp. 17–24, doi: https://doi.org/10.1016/j.procs.2021.07.006
21. Vouros, G., Glenis, A. and Doulkeridis, C. (2020), "The Delta Big Data Architecture for Mobility Analytics", *Proceedings - 2020 IEEE 6th International Conference on Big Data Computing Service and Applications, BigDataService 2020*, pp. 25–32, doi: https://doi.org/10.1109/bigdataservice49289.2020.00012
22. Chen, Z., Shao, H., Li, Y., Lu, H. and Jin, J. (2022), "Policy-Based Access Control System for Delta Lake", *Proceedings - 2022 10th International Conference on Advanced Cloud and Big Data, CBD 2022*, pp. 60–65, doi: https://doi.org/10.1109/cbd58033.2022.00020
23. Armbrust, M., Das, T., Sun, L., Yavuz, B., Zhu, S., Murthy, M., Torres, J., Van Hovell, H., Ionescu, A., Łuszczak, A. and Zaharia M. (2020), "Delta Lake: High-Performance ACID Table Storage over Cloud Object Stores", *Proceedings of the VLDB Endowment*, vol. 13, pp. 3411–3424, doi: https://doi.org/10.14778/3415478.3415560
24. (2025), *Build Reliable Data Lakes with Delta Lake*, Announcing Delta Lake 4.0 on Apache Spark™ 4.0, The Linux Foundation, Delta Lake, available at: https://delta.io/
25. Armbrust, M., Ghodsi, A., Xin, R. and Zaharia, M. (2021), "Lakehouse: A New Generation of Open Platforms That Unify Data Warehousing and Advanced Analytics", *11th Annual Conference on Innovative Data Systems Research (CIDR '21)*, available at: https://www.cidrdb.org/cidr2021/papers/cidr2021_paper17.pdf
26. Mandl, K.D., Gottlieb, D., Mandel, J.C., Ignatov, V., Sayeed, R., Grieve, G., Jones, J., Ellis, A. and Culbertson, A. (2020), "Push Button Population Health: The SMART/HL7 FHIR Bulk Data Access Application Programming Interface", *npj Digital Medicine*, vol. 3, 151, doi: https://doi.org/10.1038/s41746-020-00358-4
27. Jones, J.R., Gottlieb, D., McMurry, A.J., Atreja, A., Desai, P.M., Dixon, B.E., Payne, P.R.O., Saldanha, A.J., Shankar, P., Solad, Y. and Mandl K.D. (2024), " Real World Performance of the 21st Century Cures Act Population-Level Application Programming Interface", *Journal of the American Medical Informatics Association*, vol. 31, pp. 1144–1150, doi: https://doi.org/10.1093/jamia/ocae040
28. Parkins, M.D., Lee, B.E., Acosta, N., Bautista, M., Hubert, C., Hrudey, S.E., Frankowski, K. and Pang, X.-L. (2023), "Wastewater-Based Surveillance as a Tool for Public Health Action: SARS-CoV-2 and Beyond", *Clinical Microbiology Reviews*, vol. 37, e00103-22, doi: https://doi.org/10.1128/cmr.00103-22
29. van der Drift, A.-M.R., Welling, A., Arntzen, V., Nagelkerke, E., van der Beek, R.F.H.J. and Maria, A. (2025), "Wastewater Surveillance Studies on Pathogens and Their Use in Public Health Decision-Making: A Scoping Review", *The Science of The Total Environment*, vol. 993, 179982, doi: https://doi.org/10.1016/j.scitotenv.2025.179982
30. Lewis, A.L., Weiskopf, N.G., Abrams, Z.B., Foraker, R.E., Lai, A.M., Payne, P. and Gupta, A. (2023), "Electronic Health Record Data Quality Assessment and Tools: A Systematic Review", *Journal of the American Medical Informatics Association*, vol. 30, pp. 1730–1740, doi: https://doi.org/10.1093/jamia/ocad120
31. Ghalavand, H., Shirshahi, S., Rahimi, A., Zarrinabadi, Z. and Amani, F. (2024), "Common Data Quality Elements for Health Information Systems: A Systematic Review", *BMC Medical Informatics and Decision Making*, vol. 24, 243, doi: https://doi.org/10.1186/s12911-024-02644-7
32. Rilkoff, H., Struck, S., Ziegler, C., Faye, L., Paquette, D. and Buckeridge, D. (2024), "Innovations in Public Health Surveillance: An Overview of Novel Use of Data and Analytic Methods", *Canada Communicable Disease Report*, vol. 50, pp. 93–101, doi: https://doi.org/10.14745/ccdr.v50i34a02
33. Bizzotto, A., Guzzetta, G., Marziano, V., Manso, M.D., Urdiales, A.M., Petrone, D., Cannone, A., Sacco, C., Poletti, P., Manica, M. and Merler S. (2024), " Increasing Situational Awareness through Nowcasting of the Reproduction Number", *Frontiers in Public Health*, vol. 12, article number 1430920, doi: https://doi.org/10.3389/fpubh.2024.1430920
34. Richard, D.M., Susswein, Z., Connolly, S., Myers y Gutiérrez, A., Thalathara, R., Carey, K., Koumans, E.H., Khan, D., Masters, N.B., McIntosh, N. and Gostic K. (2024), "Detection of Real-Time Changes in Direction of COVID-19 Transmission Using National- and State-Level Epidemic Trends Based on *R* Estimates – United States Overall and New Mexico, April–October 2024", *MMWR. Morbidity and Mortality Weekly Report*, vol. 73, pp. 1058–1063, doi: https://doi.org/10.15585/mmwr.mm7346a3
35. Rajagopal, A., Ayanian, S., Ryu, A.J., Qian, R., Legler, S.R., Peeler, E.A., Issa, M., Coons, T.J. and Kawamoto, K. (2024), "Machine Learning Operations in Health Care: A Scoping Review", *Mayo Clinic Proceedings Digital Health*, vol. 2, pp. 421–437, doi: https://doi.org/10.1016/j.mcpdig.2024.06.009
36. Ng, M.Y., Youssef, A., Pillai, M., Shah, V. and Hernandez-Boussard, T. (2024), "Scaling Equitable Artificial Intelligence in Healthcare with Machine Learning Operations", *BMJ Health & Care Informatics*, vol. 31, article number e101101, doi: https://doi.org/10.1136/bmjhci-2024-101101

37. Ribeiro, V., Wolffram, D., Moraga, P. and Bracher, J. (2025), “Post-Processing and Weighted Combination of Infectious Disease Nowcasts”, *PLoS Computational Biology*, vol. 21, e1012836, doi: https://doi.org/10.1371/journal.pcbi.1012836
38. Yan, A.P., Guo, L.L., Inoue, J., Arciniegas, S.E., Vettese, E., Wolochacz, A., Crellin-Parsons, N., Purves, B., Wallace, S., Patel, A. and Sung L. (2025), “A Roadmap to Implementing Machine Learning in Healthcare: From Concept to Practice”, *Frontiers in Digital Health*, vol. 7, 1462751, doi: https://doi.org/10.3389/fdgth.2025.1462751
39. Hayman, D., Adisasmito, W., Almuhairi, S., Behravesh, C.B., Bilivogui, P., Bukachi, S.A., Casas, N., Margarita, N., Charron, D., Chaudhary, A. and Koopmans M. (2023), “Developing One Health Surveillance Systems”, *One Health*, vol. 17, article number 100617, doi: https://doi.org/10.1016/j.onehlt.2023.100617
40. Jain, P., Kraft, P., Power, C., Das, T., Stoica, I. and Zaharia, M. (2023), “Analyzing and Comparing Lakehouse Storage Systems”, *13th Annual Conference on Innovative Data Systems Research (CIDR ’23)*, available at: https://www.cidrdb.org/cidr2023/papers/p92-jain.pdf

Надійшла (received) 27.06.2025
Прийнята до друку (accepted for publication) 10.09.2025

ВІДОМОСТІ ПРО АВТОРІВ / ABOUT THE AUTHORS

Парфенюк Юрій Леонідович – кандидат технічних наук, викладач кафедри теоретичної та прикладної інформатики, Харківський національний університет імені В. Н. Каразіна, Харків, Україна;
Yurii Parfeniuk – PhD in Automation and Computer-Integrated Technologies, Lecturer of the Theoretical and Applied Informatics Department, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine;
e-mail: parfuriy.l@gmail.com; ORCID Author ID: http://orcid.org/0000-0001-5357-1868;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=58194260300.

Базілевич Ксенія Олексіївна – кандидат технічних наук, доцент кафедри математичного моделювання та штучного інтелекту, Національний аерокосмічний університет «Харківський авіаційний інститут», Харків, Україна;
Kseniia Bazilevych – PhD in Information Technologies, Associate Professor of the Department of Mathematical Modelling and Artificial Intelligence, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine;
e-mail: ksenia.bazilevich@gmail.com, ORCID: http://orcid.org/0000-0001-5332-9545;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57202239038.

Меняйлов Євген Сергійович – кандидат технічних наук, декан факультету математики та інформатики, Харківський національний університет імені В. Н. Каразіна, Харків, Україна;
Ievgen Meniailov – PhD in Mathematical Modelling and Optimization Methods, Dean of Mathematics and Informatics Faculty, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine;
e-mail: evgenii.menyailov@gmail.com, ORCID Author ID: https://orcid.org/0000-0002-9440-8378;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57202229519

Чумаченко Дмитро Ігорович – кандидат технічних наук, доцент, доцент кафедри математичного моделювання та штучного інтелекту Національного аерокосмічного університету «Харківський авіаційний інститут», Харків, Україна; запрошений науковець Школи міжнародних відносин Балзілі, Ватерлу, Канада;
Dmytro Chumachenko – PhD, Associate Professor, Associate Professor of Mathematical Modelling and Artificial Intelligence Department, National Aerospace University “Kharkiv Aviation Institute”, Kharkiv, Ukraine; Visiting Scholar, Balsillie School of International Affairs, Waterloo, ON, Canada;
e-mail: dichumachenko@gmail.com; ORCID Author ID: https://orcid.org/0000-0003-2623-3294;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=58194260300.

Багаторівнева архітектура Delta Lakehouse для епідеміологічного моніторингу та прогнозування в умовах надзвичайних ситуацій

Ю. Л. Парфенюк, К. О. Базілевич, Є. С. Меняйлов, Д. І. Чумаченко

Анотація. Надзвичайні ситуації у сфері громадського здоров’я потребують швидкої та надійної аналітики, що поєднує сигнали реального часу з достовірними історичними даними. Відкриті, інтероперабельні платформи, які підтримують потокові та пакетні робочі процеси, дають змогу скоротити час від виявлення до реагування, зберігаючи якість даних і можливість аудиту. **Мета:** спроєктувати та обґрунтувати архітектуру інформаційної системи для аналізу епідемічних загроз в умовах надзвичайних ситуацій, яка є масштабованою, надійною та придатною до інтеграції з клінічними й некласичними джерелами даних. **Методи:** проведено структурований огляд трьох архітектур аналітики даних (Lambda, Kappa, Delta) та зіставлено їхні сильні сторони й обмеження з потребами нагляду під час криз. Виходячи з функціональних і нефункціональних вимог, визначено Lakehouse на базі Delta Lake із рівнями bronze–silver–gold, уніфікованим прийманням пакетних/потокових даних за допомогою Spark Structured Streaming, ACID-таблицями з можливістю «подорожі в часі» (time travel) та контролем схеми, а також аналітичним шаром, що підтримує прогнозування з використанням MLOps для моніторингу, перевірки дрейфу, перевчитування та відстежуваності (lineage). Результати: запропонована архітектура задовольняє ключові потреби надзвичайних умов щодо своєчасності, цілісності та відтворюваності завдяки ACID-транзакціям, версіонуванню наборів даних і курованим рівням; підтримує інтероперабельність на основі стандартів та підключення даних стічних вод, мобільності й інших екологічних джерел; забезпечує єдиний кодовий шлях для пакетної та потокової обробки, зменшуючи тягар узгодження; визначає операційні межі між затримкою та вартістю під час роботи з багатьма таблицями, що оновлюються майже в реальному часі. Окреслено практичні підходи до перевірок якості на «срібному» рівні, правил промоції до «золотого» рівня та управління моделями. **Висновки:** Lakehouse на основі Delta пропонує чіткий шлях до платформи нагляду, готової до роботи в надзвичайних умовах, яка масштабується разом зі зростанням даних, інтегрує різнорідні джерела та підтримує надійне прогнозування. Наступні кроки включають пілотне розгортання з партнерами у сфері громадського здоров’я, вимірювання фактичних затримок і вартості, а також проспективну валідацію прогнозування та оповіщення в реальних умовах.

Ключові слова: епідеміологічний нагляд; аналітика спалахів; Lakehouse; Delta Lake; машинне навчання.