UDC 004.62
DOI https://doi.org/10.32782/IT/2023-1-13

Maksym ILIN
PhD Student at Department of Computer Systems Software of National Technical University of Ukraine «Igor Sikorsky Kyiv polytechnic institute», Peremohy Ave, 37, Kyiv, Ukraine, 03056, hellomax42@gmail.com
ORCID: *0009-0001-0803-3726*

Liubov OLESHCHENKO
Candidate of Technical Sciences, Associate Professor at Computer Systems Software Department of National Technical University of Ukraine «Igor Sikorsky Kyiv polytechnic institute», Peremohy Ave, 37, Kyiv, Ukraine, 03056, oleshchenkoliubov@gmail.com
ORCID: *0000-0001-9908-7422*
Scopus Author ID: *54795717500*

To cite this article: Ilin M., Oleshchenko L. (2023). Suchasnyi stan ta perspektyvy rozvytku metodiv obroblennia heterohennykh potokovykh danykh [Current state and development prospects of heterogeneous streaming data processing methods]. *Information Technology: Computer Science, Software Engineering and Cyber Security,* 1, 100–106, doi: https://doi.org/10.32782/IT/2023-1-13

CURRENT STATE AND DEVELOPMENT PROSPECTS OF HETEROGENEOUS STREAMING DATA PROCESSING METHODS

Heterogeneous streaming data processing is a rapidly expanding field of research and development in data processing and analytics. The proliferation of diverse data sources, including social media, sensor networks, and Internet of Things (IoT) devices, has resulted in an increasing heterogeneity of streaming data in terms of data types, formats, and velocities. This presents significant challenges in processing and analyzing real-time data for actionable insights. The diversity of data types, formats, and velocities in streaming data introduces complexities that require advanced techniques and algorithms for effective processing and analysis. Data streams can consist of various data types, such as text, images, videos, sensor readings, and social media posts, each with its unique characteristics and structures. Data streams can arrive in different formats, including structured, semi-structured, and unstructured data, which may require different processing approaches. The velocities at which data streams are generated can vary, ranging from high-velocity data streams that demand real-time processing to low-velocity data streams that allow batch processing. Addressing the heterogeneity of streaming data requires robust techniques that can handle diverse data types, formats, and velocities to ensure accurate and meaningful real-time data analysis. This review analyses the current research and publications on heterogeneous streaming data processing. The challenges and opportunities in processing diverse data streams in real-time, is discussed. The latest research and publications in this area are reviewed, including advancements in stream processing frameworks, machine learning algorithms, edge computing, IoT, AI, and quantum computing. The paper determines the purpose of the research, which is to provide an overview of the current state and development prospects of heterogeneous streaming data processing; presents the leading research material, including key findings and insights from recent studies. The paper concludes with prospects for further research and innovation in this field, highlighting the need to address challenges such as data heterogeneity, data velocity, concept drift, privacy and security, explainability, and the potential of quantum computing for real-time data processing.

Key words: *heterogeneous data streams, stream processing, real-time analytics, edge computing, Internet of Things (IoT), machine learning, quantum computing.*

Максим ІЛЬЇН
аспірант кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», просп. Перемоги, 37, Київ, Україна, 03056, hellomax42@gmail.com.
ORCID: *0009-0001-0803-3726*

Любов ОЛЕЩЕНКО
кандидат технічних наук, доцент кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», просп. Перемоги, 37, Київ, Україна, 03056, oleshchenkoliubov@gmail.com.
ORCID: *0000-0001-9908-7422*
Scopus Author ID: *54795717500*

Бібліографічний опис статті: Ільїн М., Олещенко Л. (2023). Сучасний стан та перспективи розвитку методів оброблення гетерогенних потокових даних. *Information Technology: Computer Science, Software Engineering and Cyber Security,* 1, 100–106, doi: https://doi.org/10.32782/IT/2023-1-13

СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ МЕТОДІВ ОБРОБЛЕННЯ ГЕТЕРОГЕННИХ ПОТОКОВИХ ДАНИХ

Оброблення гетерогенних потокових даних — це сфера досліджень і розробок у сфері аналітики даних, що в наш час розвивається доволі стрімко. Поширення різноманітних джерел даних, включаючи соціальні медіа, сенсорні мережі та пристрої Інтернету речей (IoT) призвело до зростання неоднорідності потокових даних з точки зору типів даних, форматів та швидкості їх генерації. Це створює значні труднощі в обробленні та аналізі даних у реальному часі для виділення корисної інформації. Різноманітність типів даних, форматів і швидкості генерування потокових даних створює нові завдання, вирішення яких потребує застосування передових методів і алгоритмів для ефективного оброблення й аналізу. Потоки даних можуть складатися з різних типів даних, таких як текст, зображення, відео, дані з датчиків і публікації в соціальних мережах, кожен зі своїми унікальними характеристиками та структурою. Потоки даних можуть надходити в різних форматах, включаючи структуровані, напівструктуровані та неструктуровані дані, для яких можуть знадобитися різні підходи до оброблення. Крім того, швидкість, з якою генеруються потоки даних, може змінюватися: від високошвидкісних потоків даних, які вимагають обробки в реальному часі, до низькошвидкісних потоків даних, які дозволяють пакетне оброблення. Щоб вирішувати проблеми неоднорідності потокових даних, потрібні надійні методи, які можуть обробляти дані різних типів, форматів і швидкостей, щоб забезпечити точний і значущий аналіз даних у реальному часі. У цьому огляді аналізуються поточні дослідження та публікації з оброблення гетерогенних потокових даних. У статті розглядаються проблеми та можливості оброблення потоків різних типів даних у режимі реального часу. Розглянуто останні дослідження та публікації в цій галузі, включно з досягненнями в структурах потокової обробки, алгоритмах машинного навчання, периферійних обчисленнях, Інтернеті речей, штучному інтелекті та квантових обчисленнях. Стаття містить огляд сучасного стану та перспектив розвитку оброблення гетерогенних потокових даних; представлені провідні сучасні дослідження, включаючи ключові висновки та ідеї. Наведено перспективи та пропозиції подальших досліджень та інновацій у даній галузі, підкреслюючи необхідність розв'язання таких проблем, як неоднорідність даних, швидкість передачі даних, дрейф концепції, конфіденційність і безпека, потенціал квантових обчислень для оброблення даних у реальному часі.

Ключові слова: *гетерогенні потокові дані, обробка потоків, аналітика в реальному часі, периферійні обчислення, Інтернет речей, машинне навчання, квантові обчислення.*

Introduction. Problem Statement. The proliferation of diverse data sources, such as social media, sensor networks, and IoT devices, has led to an explosion of data streams that are heterogeneous in terms of data types, formats, and velocities. Traditional data processing approaches are ill-equipped to handle the challenges posed by such diverse and rapidly changing data streams. Processing heterogeneous streaming data in real-time requires overcoming several challenges, including data integration, data transformation, data quality, data velocity, concept drift, privacy and security, explainability, and scalability. Furthermore, there is a need to develop advanced algorithms, techniques, and frameworks to efficiently process and analyze diverse data streams in real-time to derive actionable insights and make informed decisions.

Related research. Edge computing has emerged as a promising approach to process data streams closer to the source, reducing the need for data transfer and enabling real-time processing at the edge of the network (Bajić et al., 2019). IoT technologies have also played a significant role in processing heterogeneous streaming data, as IoT devices generate diverse data streams that require real-time processing for various applications such as smart cities, healthcare, and transportation (Nadeem et al., 2022). AI techniques, such as deep learning, reinforcement learning, and transfer learning, have been applied to heterogeneous streaming data processing to enable advanced analytics and decision-making (Seng et al., 2022). Quantum computing, a cutting-edge technology, has also shown potential in processing real-time data streams, leveraging the unique properties of quantum systems such as superposition and entanglement to perform complex computations efficiently.

The latest research and publications in this field have also addressed challenges related to data heterogeneity, data velocity, concept drift, privacy and security, explainability, and scalability. Various techniques have been proposed for data integration and transformation, including schema matching, ontology-based approaches, and data mapping techniques (Aydar & Ayvaz, 2017). Data quality assessment and data cleansing techniques have been developed to ensure the accuracy and reliability of processed data streams. Privacy-preserving techniques, such as data anonymization, encryption, and differential privacy, have been employed to protect sensitive data in real-time data streams (Majeed & Hwang, 2023). Explainable AI techniques, such as rule-based methods, model interpretability, and visual analytics, have been developed to enable transparency and interpretability in real-time analytics. Scalability has been addressed through techniques such as parallel processing, distributed computing, and cloud computing, to handle large-scale and high-velocity data streams (Bajić et al., 2019).

The main goal of the article is to provide a comprehensive overview of the current state and development prospects of heterogeneous streaming data processing. This includes analyzing the challenges and opportunities in processing diverse data streams in real-time, reviewing the latest research and publications in this area, presenting the leading research material, and discussing prospects for further research and innovation.

An overview of the leading research material. Several key findings and insights have emerged from recent research and publications in the field of heterogeneous streaming data processing. Firstly, stream processing frameworks such as Apache Flink, Apache Kafka, and S4 have become popular for processing large-scale data streams in real-time, providing essential features such as windowing, event time processing, and fault tolerance. These frameworks have enabled efficient processing of diverse data streams in real-time, facilitating various applications such as real-time analytics, anomaly detection, and recommendation systems. Secondly, machine learning algorithms have been developed to handle concept drift and adapt to changing data distributions in real-time. Online learning algorithms, such as online clustering, online classification, and online regression, have been proposed to update models continuously as new data streams arrive, allowing for adaptive and dynamic modeling. Adaptive algorithms, such as adaptive windowing, adaptive feature selection, and adaptive ensemble methods, have been proposed to handle concept drift and enable accurate and robust prediction in real-time (Díaz et al., 2015). Transfer learning techniques have also been applied to streaming data processing to leverage knowledge from related tasks or domains to improve model performance in real-time. Deep learning algorithms, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), have been utilized for processing sequential and time-series data streams in real-time, achieving state-of-the-art performance in various applications. Thirdly, data integration and transformation techniques have been developed to handle data heterogeneity in real-time data streams. Ontology-based approaches, such as RDF (Resource Description Framework) and OWL (Web Ontology Language), have been used for semantic data integration, enabling meaningful integration of diverse data streams with varying data schemas and formats. RDF is a W3C standard for representing and exchanging data on the web, providing a flexible and extensible framework for describing resources and their relationships. OWL, on the other hand, is a powerful ontology language used for expressing rich and complex knowledge representations, allowing for advanced reasoning and inference capabilities. Data mapping techniques, such as schema matching and data schema evolution, have been proposed to automatically map data from different sources to a common schema for processing and analysis. RDF and OWL provide semantic modeling capabilities that facilitate data mapping by allowing the specification of semantic relationships between data elements from different sources. Schema matching techniques leverage RDF and OWL to infer mappings between data schemas based on their semantic representations, enabling automatic alignment of data with varying schemas. Data schema evolution, on the other hand, refers to the dynamic adaptation of data schemas over time, and RDF and OWL provide a flexible framework for expressing and managing schema changes, ensuring that data mappings remain valid and accurate even as data sources evolve. Furthermore, various techniques have been proposed to ensure data quality and reliability in real-time data streams. Data cleansing techniques, such as outlier detection, missing value imputation, and data validation, have been developed to ensure the accuracy and integrity of processed data streams. Data profiling and data lineage techniques have been used to track and trace data quality from the source to the processing stage in real-time. Privacy and security have also been important concerns in heterogeneous streaming data processing. Techniques such as data anonymization, encryption, and differential privacy have been employed to protect sensitive data in real-time data streams. Access control and authentication mechanisms have been proposed to ensure data security and prevent unauthorized access to streaming data. Moreover, explainable AI techniques have been developed to enable transparency and interpretability in real-time analytics. Rule-based methods, such as decision trees and rule induction, have been used to generate interpretable rules for explaining model predictions in real-time. Model interpretability techniques, such as LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations), have been proposed to provide local explanations for model predictions on individual data instances in real-time. LIME generates interpretable explanations for complex models by approximating them with locally-linear models around each prediction (Fig.1).

At the same time, SHAP uses game theory to allocate the contribution of each feature towards a prediction, providing a more globally consistent explanation. (SHAP vs. LIME vs. Permutation Feature Importance – Medium, n.d) These techniques help users understand the factors that influenced a model's prediction on a specific data instance, enhancing the transparency and trustworthiness of the AI system. Visual analytics approaches, such as interactive dashboards and visualization techniques, have been utilized to enable human-understandable representations of complex streaming data and model outputs. Scalability has been a critical consideration in processing heterogeneous streaming data. Techniques such as parallel processing, distributed computing, and cloud computing have been employed to handle large-scale and high-velocity data streams. Stream partitioning and load balancing techniques have been proposed to distribute data streams across multiple processing nodes for efficient and parallel processing. Containerization and microservices architectures have been used to enable flexible and scalable deployment of streaming data processing systems in cloud and edge computing environments.

Leading approaches to heterogenous data processing

Several key research areas and approaches have emerged in the field of heterogeneous streaming data processing.

Data Preprocessing Techniques. Data preprocessing plays a crucial role in handling heterogeneous streaming data. Various techniques have been proposed to handle data heterogeneity, such as data cleansing, data transformation, and data enrichment. For example, data profiling and data cleansing approach for cleaning and validating streaming data streams with varying data quality. This approach utilized semantic web technologies to perform data profiling and data cleansing tasks in real-time, and achieved improved data quality for further processing. Also a data transformation approach based on ontology alignment was proposed for transforming data streams with different schemas and formats into a unified schema, enabling seamless integration of diverse data streams for real-time analytics (Doan et al., 2020).

Machine Learning Algorithms for Real-time Analytics. Machine learning algorithms have been widely used for real-time analytics on streaming data. Various approaches have been proposed to handle heterogeneous streaming data using machine learning algorithms, such as transfer

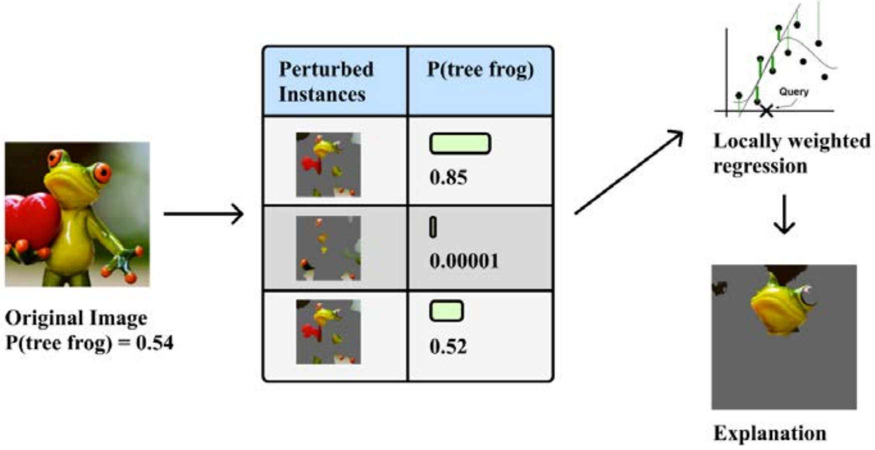

Figure 1. Explanation of a prediction conduction using LIME

learning, deep learning, and ensemble methods. For example, a transfer learning approach for streaming data processing, where the leveraged knowledge from related tasks or domains to improve the accuracy and robustness of models in real-time. It uses transfer learning techniques, such as domain adaptation and domain generalization, to adapt models to different data distributions and achieve improved performance on streaming data streams. Another proposed deep learning approach is based on recurrent neural networks (RNNs) and convolutional neural networks (CNNs) for processing sequential and time-series data streams in real-time (Zhu et al., 2022). It achieved state-of-the-art performance in applications such as speech recognition and natural language processing, where the data streams were heterogeneous in nature with varying data types and velocities.

Data Integration and Transformation Techniques. Data integration and transformation techniques are critical for handling data heterogeneity in streaming data processing. Several approaches have been proposed to enable meaningful integration of diverse data streams with different schemas, formats, and velocities. For example, an ontology-based approach for semantic data integration, which uses RDF and OWL to model and represent heterogeneous data streams and perform semantic matching and reasoning tasks for data integration achieved improved data integration accuracy and semantics-aware processing of streaming data streams. Another proposed data mapping approach is aimed at automatically mapping data from different sources to a common schema for processing and analysis. It uses schema matching and data schema evolution techniques to handle schema heterogeneity in real-time data streams and achieved efficient and accurate data integration for further processing.

Privacy and Security Techniques. Privacy and security are critical concerns in processing heterogeneous streaming data, especially when dealing with sensitive data or data from different sources with varying security requirements. Various techniques have been proposed to ensure privacy and security in the processing of streaming data. For example, a privacy-preserving data publishing approach for protecting sensitive information in streaming data streams, which utilizes techniques such as data anonymization, data perturbation, and data aggregation to protect privacy while allowing meaningful processing of streaming data. In addition, a secure data-sharing approach was proposed for the collaborative processing of streaming data from different sources, which uses secure multi-party computation techniques to enable secure and private data sharing while preserving the confidentiality and integrity of data (Maximizing Collaboration Through Secure Data Sharing - Accenture, n.d).

Scalability and Distributed Processing Technique. Scalability is a key requirement in processing heterogeneous streaming data, as data streams can be massive and continuously evolving. Various approaches have been proposed to enable scalable and distributed processing of streaming data. For example, Apache Storm, a distributed stream processing framework for processing large-scale and high-velocity data streams. Apache Storm provides fault-tolerance, reliability, and scalability for processing streaming data in real-time across multiple nodes and clusters. The architecture of Apache Storm is built upon the concept of spouts and bolts, which are the fundamental building blocks of the system. Spouts act as the sources of information and are responsible for fetching data from external sources and transferring it to one or more bolts for processing. Bolts, on the other hand, are responsible for processing the data received from spouts and performing various operations on it. The data flow between spouts and bolts is organized in a directed acyclic graph (DAG), where spouts and bolts are interconnected based on the data flow requirements defined by developers. In other words, developers have the flexibility to define how the spouts and bolts are connected, allowing them to design custom data processing topologies that suit their specific use cases and requirements. Apache Flink is distributed stream processing framework that supports batch and stream processing, event time processing, and state management for processing streaming data with high throughput and low latency. In Flink, data is treated as a continuous stream of records that flow through the system from sources to sinks, undergoing various transformations along the way. At the heart of Apache Flink's processing model are streams, which represent the continuous flow of data records that are generated from various sources, such as Kafka, Kinesis, or custom sources. These streams are processed in real-time, allowing for near-instantaneous processing of data as it arrives into the system. Streams in Flink are designed to handle large volumes of data with high throughput and low latency, making it well-suited for processing big data in real-time scenarios. Flink supports a wide range of transformations that can be applied to the data as it flows through the system. Transformations

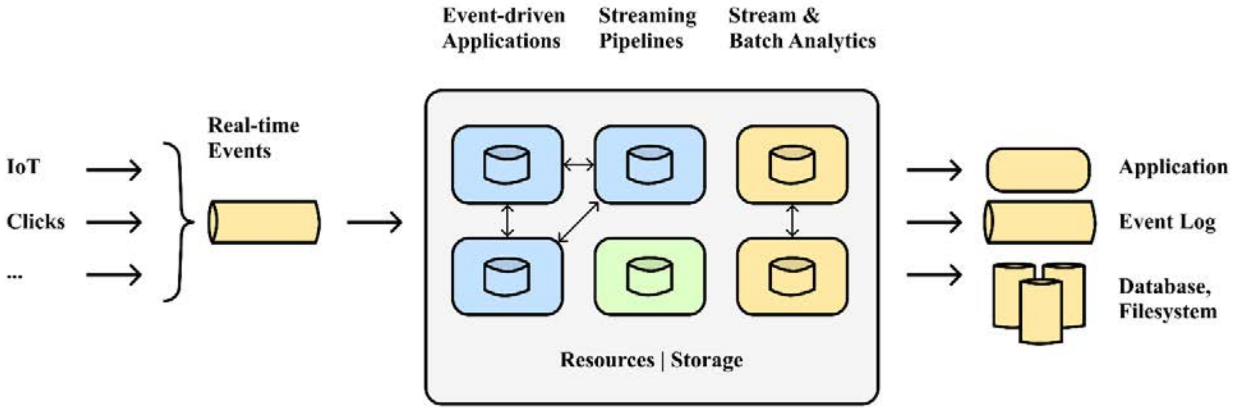

Figure 2. The Flink application scenario for working with real-time data

are operations that can be applied to streams to modify, enrich, or aggregate the data. Examples of transformations in Flink include filtering, mapping, aggregating, and joining data streams, among others. These transformations allow developers to define complex data processing logic using a declarative API or a functional programming approach, making it easy to express complex data processing pipelines in a concise and expressive manner (Fig. 2).

Conclusions and future work. In conclusion, the processing of heterogeneous streaming data is an important and challenging research area that has gained significant attention in recent years. This literature review provided an overview of the current state and development prospects of this field. It highlighted the challenges, latest research trends, and advancements in processing streaming data that is diverse in nature, such as data streams with different data types, formats, schemas, and velocities. The review presented the key techniques, approaches, and methodologies proposed and utilized for addressing the challenges of processing heterogeneous streaming data. The prospects for further research in this field are promising. As the volume, velocity, and diversity of streaming data continue to grow, there are several areas that warrant further exploration. Some potential future research directions include:

1. Developing advanced data preprocessing techniques for handling complex and diverse data streams, such as data streams with varying data quality, missing values, and data inconsistencies.

2. Exploring novel machine learning algorithms and techniques for real-time analytics on heterogeneous streaming data, including transfer learning, meta-learning, and online learning approaches.

3. Investigating advanced data integration and transformation techniques for enabling seamless integration of diverse data streams with different schemas, formats, and velocities, including ontology-based approaches, data mapping, and schema evolution techniques.

4. Advancing privacy and security techniques for protecting sensitive information in streaming data streams, including privacy-preserving data publishing, secure multi-party computation, and data encryption techniques.

5. Enhancing scalability and distributed processing techniques for processing massive and high-velocity streaming data, including distributed stream processing frameworks, event time processing, and state management approaches.

In summary, the field of heterogeneous streaming data processing has made significant progress in recent years, with advancements in data preprocessing, machine learning algorithms, data integration and transformation, privacy and security, and scalability and distributed processing. There are still challenges to be addressed, and there are ample opportunities for further research and development in this area.

REFERENCES

1. Bajić, B. et al. (2019) «Edge Computing vs. Cloud Computing: Challenges and Opportunities in Industry 4.0», p. 0864-0871. Available at: https://doi.org/10.2507/30th.daaam.proceedings.120.

2. Nadeem, M., Lee, U, S. and Younus, M. (2022) «A Comparison of Recent Requirements Gathering and Management Tools in Requirements Engineering for IoT-Enabled Sustainable Cities», Sustainability, 14(4), p. 2427. Available at: https://doi.org/10.3390/su14042427.

3. Seng, P, K. et al. (2022) «Artificial Intelligence (AI) and Machine Learning for Multimedia and Edge Information Processing», Electronics, 11(14), p. 2239. Available at: https://doi.org/10.3390/electronics11142239.

4. Aydar, M. and Ayvaz, S. (2017) «A Suggestion-Based RDF Instance Matching System», International Journal of Computer Theory and Engineering, 9(5), p. 380-384. Available at: https://doi.org/10.7763/ijcte.2017.v9.1170.

5. Majeed, A. and Hwang, O, S. (2023) «Quantifying the Vulnerability of Attributes for Effective Privacy Preservation Using Machine Learning», IEEE Access, 11, p. 4400-4411. Available at: https://doi.org/10.1109/access.2023.3235016.

6. Díaz, O, A. et al. (2015) «Fast Adapting Ensemble: A New Algorithm for Mining Data Streams with Concept Drift», The Scientific World Journal, 2015, p. 1-14. Available at: https://doi.org/10.1155/2015/235810.

7. Ribeiro, T, M., Singh, S. and Guestrin, C. (2016) Local Interpretable Model-Agnostic Explanations (LIME): An Introduction. Available at: https://www.oreilly.com/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/.

8. SHAP vs. LIME vs. Permutation Feature Importance - Medium (no date). Available at: https://pub.towardsai.net/model-explainability-shap-vs-lime-vs-permutation-feature-importance-98484efba066.

9. Doan, Q. et al. (2020) «Integration of IoT Streaming Data With Efficient Indexing and Storage Optimization», IEEE Access, 8, p. 47456-47467. Available at: https://doi.org/10.1109/access.2020.2980006.

10. Zhu, Y. et al. (2022) «Deep Learning in Diverse Intelligent Sensor Based Systems», Sensors, 23(1), p. 62. Available at: https://doi.org/10.3390/s23010062.

11. Maximizing Collaboration Through Secure Data Sharing - Accenture (no date). Available at: https://www.accenture.com/us-en/insights/digital/maximize-collaboration-secure-data-sharing.