DOI: https://doi.org/10.36910/6775-2524-0560-2025-58-05
УДК 004.8:004.93'1
Korostin Oleksandr, Master's degree, Principal Software Engineer
https://orcid.org/0009-0007-7510-6757
Shipnext BV, Koningin Elisabethlei 16, Antwerp, Belgium

OPTIMISING MACHINE LEARNING INTEGRATION IN REAL-TIME TEXT ANALYTICS PLATFORMS: TECHNICAL APPROACHES AND PERFORMANCE CRITERIA

Korostin O. Optimising Machine Learning Integration in Real-Time Text Analytics Platforms: Technical Approaches and Performance Criteria. The article investigates the integration of machine learning into real-time platforms for analysing text streams. The relevance of the topic is driven by the growing volume of unstructured textual data and the need for its prompt and accurate processing to support decision-making in such fields as media monitoring, cybersecurity, finance, and healthcare. The effectiveness of such platforms is shown to depend on the adaptability of algorithms, analysis accuracy, scalability, and transparency of results. Special attention is paid to the technical aspects of implementation, including distributed architecture, streaming data processing, optimisation of computing resources, and integration of explainable models. The purpose of the article is to study the possibilities of integrating machine learning algorithms into real-time platforms for analysing text streams, in particular, to develop approaches to improving the efficiency of data processing, ensuring their transparency and adaptability in a changing information environment. To achieve this goal, the study applies a combination of literature analysis, comparative evaluation of existing algorithms, and an experimental assessment of technical solutions. The findings indicate that the main challenges of integration include the computational complexity of deep models, scalability constraints, and delays in data stream processing. It has been shown that the use of distributed computing technologies, hardware accelerators (GPU/TPU), and online learning mechanisms significantly improves the performance of such platforms. The application of adaptive algorithms capable of real-time parameter updates increases analysis accuracy under unstable data conditions. The study concludes that integrating machine learning into real-time systems enhances the speed, reliability, and scalability of text analytics. Further research should focus on developing universal multilingual platforms that combine energy efficiency, modularity, and high analytical performance.
Keywords: machine learning, text streams, real-time platforms, algorithm adaptability, data analysis, distributed computing, performance optimisation, algorithm transparency.

Коростін О.О. Оптимізація інтеграції машинного навчання у платформи аналізу текстових потоків у реальному часі: технічні підходи та критерії ефективності. У статті досліджується інтеграція алгоритмів машинного навчання у платформи реального часу для аналізу текстових потоків. Актуальність теми зумовлена стрімким зростанням обсягів неструктурованих текстових даних та необхідністю їх своєчасної обробки для підтримки прийняття рішень у сферах медіа-моніторингу, кібербезпеки, фінансів і медицини. Показано, що ефективність таких платформ залежить від адаптивності алгоритмів, точності аналізу, масштабованості та прозорості результатів. Особливу увагу приділено технічним аспектам реалізації, зокрема розподіленій архітектурі, потоковій обробці даних, оптимізації обчислювальних ресурсів і впровадженню пояснюваних моделей. Метою статті є дослідження можливостей інтеграції алгоритмів машинного навчання у платформи реального часу для аналізу текстових потоків, зокрема розроблення підходів до підвищення ефективності обробки даних, забезпечення їх прозорості та адаптивності в умовах змінного інформаційного середовища. Для досягнення поставленої мети використано методи аналізу наукових джерел, порівняння існуючих алгоритмів та експериментального дослідження технічних рішень. У результаті встановлено, що основними проблемами інтеграції є висока обчислювальна складність глибоких моделей, обмеження масштабованості та затримки при обробці потоків. Доведено, що застосування технологій розподілених обчислень, апаратних прискорювачів (GPU/TPU) та механізмів онлайн-навчання суттєво підвищує продуктивність таких платформ. Впровадження адаптивних алгоритмів з можливістю оновлення параметрів у реальному часі покращує якість аналізу в умовах нестабільних даних. Зроблено висновок, що інтеграція машинного навчання у платформи реального часу забезпечує підвищення швидкості, надійності та масштабованості аналітики текстових потоків. Перспективи подальших досліджень пов'язані з розробленням універсальних багатомовних платформ, які поєднують енергоефективність, модульність і високу аналітичну продуктивність.
Ключові слова: машинне навчання, текстові потоки, платформи реального часу, адаптивність алгоритмів, аналіз даних, розподілені обчислення, оптимізація продуктивності, прозорість алгоритмів.

Problem statement. With the development of digital technologies, the amount of textual data generated in real-time is growing exponentially, creating challenges for the efficient collection, processing and analysis of information. Traditional rule-based or statistical methods often lack the flexibility and speed required to extract meaningful insights from unstructured, dynamic, and multilingual text streams. In contrast, machine learning offers a powerful toolkit for recognising patterns, adapting to changing data, and handling complexity without manual intervention.

Machine learning, as a key component of modern information systems, provides methods for identifying key trends, classifying data, predicting events, and even automatically generating content — all of which are increasingly needed in high-pressure, time-sensitive contexts. The application of machine learning to real-time analytics aims to address the problem of latency and information overload: when data arrives faster than it can be analysed using conventional tools, critical decisions may be delayed or misinformed. By integrating machine learning into real-time platforms, the goal is to enhance the speed, scalability, and adaptability of analytical systems to support decision-making in domains such as media monitoring, crisis response, financial forecasting, and cybersecurity. These are areas where milliseconds matter, and where insight must be extracted from large volumes of unstructured text with minimal delay.

The application of machine learning for real-time text stream analysis requires solving a number of important scientific and practical tasks. In particular, this includes the development of efficient algorithms capable of analysing text data in real-time, optimising the use of computing resources, and ensuring transparency and explainability of the analysis results. The scientific significance of the research lies in the formation of new approaches to text data processing that will contribute to the development of artificial intelligence methods and expand their applications. The practical significance is determined by the possibility of integrating these developments into real platforms for analysing information flows, which will help to increase the efficiency of decision-making, reduce risks and improve management processes.

Analysis of the latest research and publications. Integrating machine learning into real-time platforms for analysing text streams is a relevant area of research that demonstrates significant achievements in improving the accuracy, speed, and adaptability of data processing. A. Guha and D. Samanta [1] developed a multi-level text classification system that provides high accuracy of results in real time by combining machine learning algorithms. Their approach proved effective in minimising document classification errors, which is especially important for streaming data.

M. Yu, Q. Huang, H. Qin, C. Scheele and C. Yang [2] studied applying deep learning to social media analysis during disasters like hurricanes. Their approach proved highly effective in real-time, allowing for the rapid identification of critical information and situational awareness. In their review, Q. Li, H. Peng, J. Li, et al. [3] examinedail the evolution of text classification, focusing on deep learning methods. The results confirm that such approaches significantly increase classification accuracy, even for large amounts of streaming data.

G. Gomes, J. Read, A. Bifet, and J. Gama [4] studied the main challenges of working with streaming data and proposed adaptive machine-learning models to ensure the stable operation of real-time systems. Their study emphasises the importance of optimising algorithms for high-speed text processing. In their review, M. Bahri, A. Bifet, J. Gama, and S. Maniu [5] identified the key tasks of data stream analysis and emphasised the need to standardise processing methods to improve their efficiency.

B. Hammou, A. Lahcen, and S. Mouline [6] developed a distributed architecture for streaming analytics that uses recurrent neural networks and FastText. Their approach significantly reduced text processing time and demonstrated high performance even under heavy load. A. Rodrigues, R. Fernandes, A. Shetty, et al. [7] studied methods for detecting spam in social networks and showed that their model provides more than 90% accuracy in real time. These results confirm the importance of combining machine learning and deep learning methods.

D. Jayanthi and G. Sumathi [8] proposed a powerful framework for streaming analytics that demonstrates high speed of processing text streams. Their method allows for efficient management of dynamic data, particularly in large systems. S. Minaee, N. Kalchbrenner, E. Cambria, et al [9] performed a comprehensive analysis of deep learning methods for text classification, emphasising the high accuracy of hybrid models. K. Lopes Dias, M. A. Pongelupe, W. M. Caminhas, and L. de Errico [10] developed an innovative model for network traffic classification that demonstrated high adaptability and can be used for text analysis. M. Umer, Z. Imtiaz, M. Ahmad and co-authors [11] proved that the combination of FastText and convolutional neural networks significantly improves text classification results by increasing the accuracy and processing speed.M. Yu, H. Qin, Q. Huang et al. [12] presented an effective solution for analysing social media in emergency situations. Their approach made it possible to quickly identify critical information, providing fast decision-making. N. Sharma, R. Sharma, and N. Jindal [13] considered the application of machine learning in text analysis, emphasising its important role in processing large amounts of data. S. Boppiniti [14] studied streaming analytics with a focus on dynamic decision-making, proposing a model that optimises real-time text processing. O. Khodorkovskyi [15] examined methods of training artificial intelligence agents for process control systems that allow the integration of machine learning into current platforms.

The results of the analysis confirm that the use of machine learning to analyse text streams in real time allows achieving high accuracy, efficiency, and adaptability. The key areas are the development of adaptive algorithms, hybrid models, and optimisation of methods for processing large amounts of data.

Identification of previously unsolved parts of the problem. Despite significant progress in the application of machine learning to text stream analysis, the problems of algorithms' adaptability to dynamic conditions, noisy data, and multilingualism remain unresolved, which reduces the accuracy and speed of analysis. The absence of universal evaluation criteria that take into account scalability, transparency and noise resistance limits the effectiveness of current approaches.

Technical challenges of integration, including scalability and resource optimisation, make it difficult to implement productive real-time solutions due to significant computational costs.

The study aims to overcome these gaps by analysing the adaptability of algorithms, developing evaluation criteria, and optimising technical solutions to improve the efficiency of platforms in dynamic environments.

The **purpose of** the **article** is to study the possibilities of integrating machine learning algorithms into real-time platforms for analysing text streams, in particular, to develop approaches to improving the efficiency of data processing, ensuring their transparency and adaptability in a changing information environment.

Objectives of the article:

1. To analyse modern machine learning methods used to analyse text streams in real time and to develop criteria for their evaluation, taking into account the efficiency of working with large amounts of data.

2. Investigate the technical aspects of integrating machine learning algorithms into real-time platforms, including scalability, computational performance, and adaptability to dynamic conditions.

3. Develop recommendations for optimising machine learning algorithms to improve the accuracy, performance and stability of text stream analysis.

Summary of the main material. The key motivation for applying machine learning to real-time text analytics lies in its ability to overcome the limitations of traditional rule-based systems, which struggle to scale, adapt, and interpret dynamic, high-volume data flows. In real-time environments, it is essential to extract relevant insights from constantly updating unstructured text, where the volume and velocity of data often exceed the capacity of manual or static analytical tools. Machine learning enables automated, flexible, and context-aware processing, making it a critical component for timely and accurate decision-making.

Real-time analysis of text streams using machine learning methods is one of the key tasks of modern information systems. The methods used to solve these tasks include text classification, topic detection, sentiment analysis, anomaly detection, and text generation. These algorithms allow processing large amounts of data in a short time, providing the ability to make quick decisions based on the analysis of relevant information. The most popular approaches are deep learning models (neural networks), natural language processing (NLP) methods, and classical classification and regression algorithms (Table 1).

Table 1 – Modern machine learning methods for analysing text streams in real time

Method	Description	Advantages of using
Neural networks (RNN, LSTM, Transformer)	Provides context-aware text analysis by processing word sequences.	High accuracy of text analysis with contextualisation; scalability to work with big data.
Classification algorithms	Methods such as SVM or Random Forest are used to categorise text.	Easy to implement; fast processing of small amounts of data; efficient for basic tasks.
Thematic modelling (LDA)	Identify the main themes in large text datasets.	Automation of text stream analysis; ability to process heterogeneous texts.
Sentiment analysis	It is used to determine the emotional tone of a text, positive or negative.	Application in marketing and customer experience research; adaptability to different domains.
Anomaly detection	Methods for finding atypical texts or behavioural patterns in text streams.	Improved security; effective in detecting rare events or threats.

Source: compiled by the author on the basis of [1; 2; 4; 5].

Real-time analysis of text streams using machine learning methods is one of the key tasks of modern information systems. The methods used to solve these tasks include text classification, topic detection, sentiment analysis, anomaly detection, and text generation. The most popular approaches include deep learning models (neural networks), natural language processing (NLP) methods, and classical classification and regression algorithms. For example, neural networks such as GPT are widely used for automatic text generation, providing high accuracy and contextual relevance [16]. BERT, developed by Google, has become the basis for many NLP applications due to its ability to consider both the text's previous and subsequent context [17]. TensorFlow, a leading platform for developing machine learning models, offers tools for implementing sentiment analysis and text classification [18]. Scikit-learn effectively implements methods such as classification and topic modelling, in particular through the LDA algorithm [19]. IBM is actively implementing real-time solutions focused on data analysis and anomaly detection, which allows automating processes in various industries [20].

These methods and platforms are key to ensuring practical real-time analysis of text streams, improving the accuracy of data processing, and adapting to changing conditions.

Evaluation of algorithms' effectiveness for working with large amounts of text data is based on generally accepted criteria that allow determining whether an algorithm meets the tasks set. Such criteria are used to analyse algorithms' performance, accuracy, scalability, and adaptability in different conditions. They are universal and can be used to evaluate algorithms in many areas, including real-time text stream analysis. The main criteria include accuracy, which describes the level of conformity of the results to the expected ones, speed, which determines the ability of the algorithm to process data in real time, noise tolerance, which shows its effectiveness in cases with incomplete or corrupted data, and scalability, which ensures the ability to work with increasingly large amounts of information. The transparency of the algorithm also plays an important role, as it allows for the interpretation of the results, which is critical in areas such as medicine or finance (Table 2).

Table 2 – Criteria for evaluating the effectiveness of algorithms for working with large amounts of text data

Criterion	Description	Relevance to modern conditions
Analysis accuracy	The percentage of correctly processed text data.	Ensures the reliability of the results required for decision-making.
Speed of operation	The time required to process one text stream.	Allows you to process large amounts of data in real time.
Resistance to noise	The ability of the algorithm to work with incomplete or damaged data.	Increases efficiency in situations with a large number of incorrect records.
Scalability	The ability of the algorithm to work with increasingly large amounts of data.	Allows the algorithm to be used in large systems.
Transparency of results	The level of clarity of the results obtained for the end user.	Ensures confidence in the results and the ability to interpret them.

Source: compiled by the author on the basis of [3; 4; 5; 9; 11].

Each of these criteria has practical implications for data analytics. For example, accuracy is key in areas such as medicine or finance, where even minor errors can have serious consequences. The speed of the algorithm is especially important in real-time systems, for example, for monitoring crisis situations or detecting fake news on social media. Resistance to noise ensures efficiency in working with low-quality data, which is often the case in open sources. Scalability allows algorithms to maintain performance even if the volume of text streams increases significantly. Transparency of results helps to increase trust in algorithms, as users are able to understand how certain conclusions were reached.

In real life, these criteria are interrelated: a highly accurate algorithm may require more time to process data, and scalability may affect noise resistance. Therefore, their use in combination allows you to balance different aspects of efficiency and find the best solution for a particular task.

Integration of machine learning into real-time platforms for text stream analysis requires not only high accuracy of algorithms, but also the ability of systems to provide stable performance in dynamic conditions [4]. One of the key aspects is the optimal use of computing resources, which affects the speed of text data processing, system resilience to high loads, and scalability.

The technical implementation of such platforms includes three key elements: the architectural organisation of the system, data flow management, and algorithm adaptability. The architectural component involves the use of multi-level distributed systems that allow processing text streams in parallel, reducing delays. Data flows are managed through streaming processing platforms such as Apache Kafka, which allows for real-time information processing. The adaptability of algorithms implies their ability to dynamically adjust parameters to reduce computational complexity (Table 3).

Table 3 – Technical aspects of integrating machine learning into real-time platforms for analysing text streams

The integration aspect	Technical description	Expected effect
Distributed architecture	Organising computing on multiple nodes to reduce latency and increase speed.	Improved system performance and stability under high loads.
Data flow management	Processing information as it arrives with minimal delay.	Reduced time for analysing text streams.
Adaptability of algorithms	Dynamic optimisation of model parameters during processing.	Improved processing accuracy by adjusting to changing conditions.
API integration	Interaction with other systems for data transmission and processing.	Improved accessibility and interoperability between platforms.
Energy efficiency	Optimising hardware performance to reduce energy consumption.	Reduced energy consumption and infrastructure support costs.

Source: compiled by the author on the basis of [4; 5; 6; 8; 10; 15].

Table 3 demonstrates the key aspects of integrating machine learning into real-time platforms, each of which addresses specific technical challenges required to ensure the stable operation of such systems. Distributed architecture is the basis for efficient processing of large amounts of data, as it allows for even distribution of computational loads across multiple nodes, reducing latency and increasing system performance. This approach is indispensable for platforms that work with scalable text streams, where the amount of data can change dynamically.

Data flow management provides real-time information processing, which minimises delays in the transmission and analysis of text streams [7]. This allows platforms to respond to new data in a timely manner, which is critical for monitoring social media or identifying crisis events. The adaptability of the algorithms allows the parameters of machine learning models to be dynamically adjusted in accordance with changes in the structure and content of text streams, which ensures more accurate analysis even in the face of uneven or unpredictable data flow.

API integration helps to create flexible systems that can effectively interact with other platforms and data sources. This ensures synchronisation between different modules of the system, which is especially important for large corporate platforms where compatibility of various software is required. Energy optimisation minimises the cost of computing resources, reducing financial and environmental costs while ensuring high system performance.

Such technical solutions are the basis for creating effective platforms that can meet the requirements of a modern dynamic environment and provide high-quality real-time analysis of text streams.

To confirm the practical applicability of machine learning integration into real-time text stream analytics platforms, an experimental environment was developed to simulate the operation of a typical high-load analytical system. The setup included a streaming data input module, a processing infrastructure for machine learning models, and a real-time performance monitoring unit.

The input data stream consisted of 10,000 short multilingual text messages in English, German, and Ukrainian, arriving at a speed of 150 messages per second via Apache Kafka. The dataset included a variety of content types — news headlines, social media posts, and synthetic messages — with intentional noise such as spelling errors, abbreviations, emojis, and mixed language segments to reflect real-world conditions.

Text processing was performed using three different machine learning models: a classical Support Vector Machine (SVM) classifier, a Random Forest ensemble, and a BERT transformer model implemented in TensorFlow using the HuggingFace Transformers library. The system was deployed in a virtualized environment using Docker containers, with workload orchestration handled via Kubernetes. The computational infrastructure included an Intel Xeon Silver 4216 32-core processor, an NVIDIA Tesla T4

GPU (16 GB), and 128 GB of RAM. Model performance was tracked in real time using TensorBoard, capturing metrics such as latency, accuracy, and resource utilization.

Table 4 – Comparative evaluation of machine learning models for real-time text stream processing

Model	Classification accuracy (%)	Average latency per message (ms)	Resource utilization (CPU / GPU)	Noise robustness	Notes
SVM	78.4	13	25% / 0%	Low	Fast, but weak against unstructured or noisy data
Random forest	81.2	24	35% / 0%	Medium	Requires pre-cleaned input for optimal results
BERT (TensorFlow)	92.7	81	60% / 85%	High	Highest accuracy, but resource-intensive

Source: own author's development

The results of the experiment demonstrate that the BERT model achieves the highest classification accuracy and shows excellent robustness to noise and non-standard input, making it highly suitable for real-time scenarios where data structure varies rapidly. However, its performance comes at the cost of significant GPU usage and higher latency. The Random Forest model provides a reasonable compromise between accuracy and computational efficiency, performing well when the input stream is preprocessed. The SVM classifier delivers the lowest latency, but its accuracy degrades noticeably when handling noisy or multilingual text streams.

This simulation allowed the observation of trade-offs between accuracy, resource consumption, and real-time responsiveness—factors critical for designing efficient, scalable, and adaptive machine learning-based platforms for text analytics.

To visualize the architecture used in the experiment, a simplified block diagram is presented in Figure 1, illustrating the flow of data and interaction between system components.

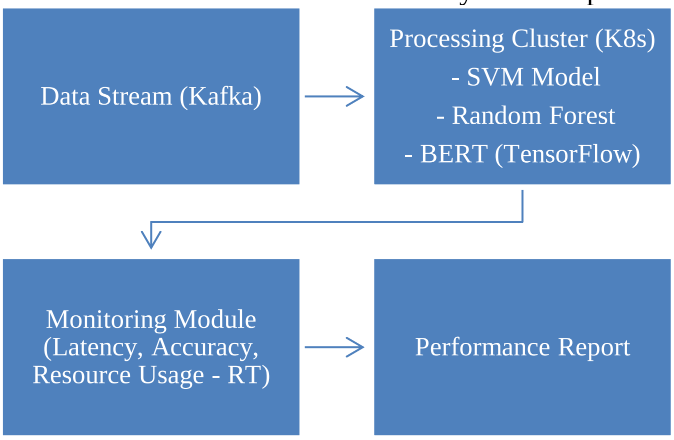

Fig.1. Experimental architecture for real-time text stream analysis

The diagram illustrates the sequential flow of data in the experimental setup: incoming text streams are transmitted via Apache Kafka to a Kubernetes-based processing cluster containing machine learning models (SVM, Random Forest, and BERT). The outputs of these models are forwarded to a real-time monitoring module that captures key performance metrics, including latency, classification accuracy, and system resource usage. The final evaluation is presented in the form of a performance report.

Integrating machine learning algorithms into real-time text stream analysis platforms faces numerous challenges that affect the efficiency, accuracy, and adaptability of such systems in dynamic environments. The main problem is the computational complexity of algorithms, especially deep learning models such as GPT [15] or BERT [16], which require significant resources to process large amounts of text data. Implementing such models in real-time systems where minimal latency is critical makes it difficult. This limitation is particularly evident in edge computing environments or mobile platforms with restricted CPU/GPU capacities, where large-scale models cannot be deployed without aggressive model compression or simplification.

Another significant problem is related to the variability of text streams. In dynamic environments, such as social media or crisis communications, the content and structure of text data are constantly changing, leading to a decrease in model accuracy due to outdated parameters. The ability of algorithms to adapt to these changes is limited, affecting the analysis quality. Most current models are trained in batch mode and lack support for continuous learning or online adaptation, which reduces their effectiveness when patterns shift rapidly. Additionally, re-training models in real time is often not feasible due to the high computational cost.

Data complexity, such as multilingualism, different formats, or a significant amount of noise, also creates challenges for effective analysis. For example, algorithms may not correctly process texts that contain grammatical errors, abbreviations, or specialised vocabulary, which is common in real-world data streams [14]. Moreover, many NLP models are initially trained on formal or structured datasets, which makes them poorly suited for analysing informal or domain-specific language unless fine-tuned — a process that again requires annotated data and time.

Another problem is the platforms' scalability. With a significant increase in the amount of data or the number of users, systems can lose stability, which affects the processing speed and accuracy of the results. This requires complex technical solutions, such as distributed computing, which are not always easy to integrate into existing infrastructure. Furthermore, scaling real-time systems involves challenges not only in computing but also in synchronising data flows, managing resource contention, and maintaining low-latency performance across nodes.

The transparency of algorithms and difficulties in interpreting the results are also serious challenges. Most modern models operate as black boxes, making it difficult to understand decision-making processes, especially when mistakes such as medicine or finance can have serious consequences. This lack of interpretability limits their practical use in regulated domains, where explainability is legally or ethically required. Additionally, concerns about algorithmic bias and the inability to audit model behaviour in real time are significant obstacles to their safe deployment.

Optimisation of algorithms for real-time text stream analysis requires consideration of both technical and methodological aspects to achieve high performance and accuracy. One of the key areas is the use of hybrid models that combine the advantages of multiple approaches, such as deep learning and topic modelling. This combination helps improve analysis flexibility and better handle heterogeneous, noisy, or multilingual data typical for real-time environments.

Another important aspect is the optimisation of computing resources, especially relevant for systems with limited processing capacity (e.g., mobile or edge devices). Distributed computing solutions and hardware accelerators such as GPUs or TPUs [7] allow for real-time processing of large text volumes. In addition, the use of containerisation platforms like Docker and orchestration systems such as Kubernetes ensures dynamic scalability and efficient resource allocation depending on workload fluctuations [8].

To address the problem of outdated model parameters and ensure responsiveness to changing input streams, it is necessary to implement online learning mechanisms that support incremental updates. This approach improves adaptability and maintains accuracy in fast-changing environments, such as social media monitoring.

Given the prevalence of noisy and unstructured data in real-world streams, preprocessing methods such as text normalisation, spelling correction, removal of duplicates, and handling of special symbols should be integrated as a standard pipeline component.

To overcome the challenge of algorithmic opacity and improve interpretability, it is recommended to use explainable models (XAI) [9], especially in high-risk domains like medicine or finance. These models support transparency and allow users to understand the basis for specific conclusions or predictions.

Regular testing and validation of algorithms on real-time data streams should be an essential part of deployment to identify performance degradation or bias early. It is also advisable to adopt energy-efficient computing practices and lightweight model architectures to ensure sustainability, especially in large-scale and always-on systems.

Thus, optimisation of machine learning algorithms must be multi-layered and closely aligned with real-time system constraints. Only such a holistic approach will ensure the reliable and efficient operation of text stream analysis platforms in dynamic and resource-sensitive environments.

Conclusions and prospects for further research. The study confirmed that integrating machine learning into real-time text analytics platforms significantly enhances their accuracy, adaptability, and efficiency in dynamic environments. Transformer-based models such as BERT demonstrated the highest classification performance and robustness to noisy data, though at the cost of high resource usage and increased latency. Classical models like SVM and Random Forest offered faster processing and lower computational demand but showed reduced accuracy, especially with unstructured input.

Key challenges include computational complexity, limited adaptability to changing data streams, and insufficient transparency of algorithmic decisions. These were addressed through experimental testing, which highlighted the trade-offs between accuracy, latency, and resource consumption.

To improve system performance, the use of distributed computing (e.g., Apache Kafka), hybrid modelling approaches, online learning, and explainable AI techniques is recommended.

Further research should focus on developing adaptive, resource-efficient algorithms and modular platforms that support real-time model updates and maintain explainability under varying data conditions.

References:

1. Guha, A., & Samanta, D. (2020). Real-time application of document classification based on machine learning. In L. Jain, S. L. Peng, B. Alhadidi, & S. Pal (Eds.), *Intelligent computing paradigm and cutting-edge technologies. ICICCT 2019* (Vol. 9, pp. 401–416). Springer. https://doi.org/10.1007/978-3-030-38501-9_37
2. Yu, M., Huang, Q., Qin, H., Scheele, C., & Yang, C. (2020). Deep learning for real-time social media text classification for situation awareness – Using Hurricanes Sandy, Harvey, and Irma as case studies. In *Social sensing and big data computing for disaster management* (pp. 33–50). Routledge. https://doi.org/10.4324/9781003106494-3
3. Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., Yu, P. S., & He, L. (2022). A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology (TIST), 13*(2), 1–41. https://doi.org/10.1145/3495162
4. Gomes, H. M., Read, J., Bifet, A., Barddal, J. P., & Gama, J. (2019). Machine learning for streaming data: State of the art, challenges, and opportunities. *ACM SIGKDD Explorations Newsletter, 21*(2), 6–22. https://doi.org/10.1145/3373464.3373470
5. Bahri, M., Bifet, A., Gama, J., & Maniu, S. (2021). Data stream analysis: Foundations, major tasks and tools. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 11*(3), e1405. https://doi.org/10.1002/widm.1405
6. Hammou, B., Lahcen, A., & Mouline, S. (2020). Towards a real-time processing framework based on improved distributed recurrent neural network variants with FastText for social big data analytics. *Information Processing & Management, 57*(1), 102122. https://doi.org/10.1016/j.ipm.2019.102122
7. Rodrigues, A. P., Fernandes, R., Shetty, A., et al. (2022). Real-time Twitter spam detection and sentiment analysis using machine learning and deep learning techniques. *Computational Intelligence and Neuroscience, 2022*(1), 5211949. https://doi.org/10.1155/2022/5211949
8. Jayanthi, D., & Sumathi, G. (2016). A framework for real-time streaming analytics using machine learning approach. In *Proceedings of National Conference on Communication and Informatics-2016* (pp. 15–21). Retrieved from https://www.researchgate.net
9. Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning-based text classification: A comprehensive review. *ACM Computing Surveys (CSUR), 54*(3), 1–40. https://doi.org/10.1145/3439726
10. Lopes Dias, K., Almeida Pongelupe, M., Caminhas, W. M., & de Errico, L. (2019). An innovative approach for real-time network traffic classification. *Computer Networks, 158*, 143–157. https://doi.org/10.1016/j.comnet.2019.04.004
11. Umer, M., Imtiaz, Z., Ahmad, M., et al. (2023). Impact of convolutional neural network and FastText embedding on text classification. *Multimedia Tools and Applications, 82*(4), 5569–5585. https://doi.org/10.1007/s11042-022-13459-x
12. Yu, M., Qin, H., Huang, Q., et al. (2020). Deep learning for real-time social media text classification for situation awareness – Using Hurricanes Sandy, Harvey, and Irma as case studies. Routledge. https://doi.org/10.4324/9781003106494-3
13. Sharma, N., Sharma, R., & Jindal, N. (2021). Machine learning and deep learning applications – A vision. *Global Transitions Proceedings, 2*(1), 24–28. https://doi.org/10.1016/j.gltp.2021.01.004
14. Boppiniti, S. (2021). Real-time data analytics with AI: Leveraging stream processing for dynamic decision support. *International Journal of Management Education for Sustainable Development, 4*(4), 15–25. Retrieved from https://ijsdcs.com
15. Khodorkovskyi, O. (2024). Methods of training and adaptation of AI agents in complex process control systems. *The American Journal of Engineering and Technology, 6*(11), 46–53. https://doi.org/10.37547/tajet/Volume06Issue11-06
16. OpenAI. (2024). GPT models for text generation and analysis. Retrieved from https://openai.com
17. Google AI. (2024). BERT: Pre-training of deep bidirectional transformers for language understanding. Retrieved from https://ai.google
18. TensorFlow. (2024). Machine learning frameworks and tutorials. Retrieved from https://www.tensorflow.org
19. Scikit-learn. (2024). Machine learning tools for data analysis. Retrieved from https://scikit-learn.org
20. IBM. (2024). AI solutions for real-time data processing. Retrieved from https://www.ibm.com