Journal of Theoretical and Applied Information Technology
30th September 2025. Vol.103. No.18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

STREAM PROCESSING ALGORITHMS FOR UNSTRUCTURED DATA ANALYSIS

VIKTORIIA SHUTENKO[1], MYKOLA MEDVEDIEV[2], YURII DOROSHENKO[3], VALERIA DROMENKO[4], NATALIIA OMETSYNSKA[5]

[1]PhD student, Department of Mathematical Support of Computer Systems, Odesa I.I. Mechnikov National University, Ukraine

[2]Professor, Department of Engineering Systems and Technologies, Educational and Scientific Institute of Municipal Management and City Economy, V. I. Vernadsky Taurida National University, Ukraine

[3]Professor, Department of Computer and Information Technologies of the Educational and Scientific Institute of Municipal Management and City Economy, V. I. Vernadsky Taurida National University, Ukraine

[4]Associate Professor, Department of Engineering Systems and Technologies, Educational and Scientific Institute of Municipal Management and City Economy, V. I. Vernadsky Taurida National University, Ukraine

[5]Associate Professor, Department of Engineering Systems and Technologies, Educational and Scientific Institute of Municipal Management and City Economy, V. I. Vernadsky Taurida National University, Ukraine

E-mail: [1]viktoriatech23@gmail.com, [2]mykolamedvediiv4@gmail.com, [3]yriiidoroshencko3@gmail.com, [4]valeridromenko0@gmail.com, [5]nataliometsyinska@gmail.com

ABSTRACT

Relevance of the selected research

The rapid growth of unstructured streaming data and the need for real-time decision-making necessitate the development of effective algorithms tailored for dynamic processing, semantic enrichment, and adaptive analytics.

Purpose

The purpose of the study is to develop an optimized mathematical model for the processing of unstructured data streams to support real-time decisions.

Methods

The study used the following methods: decomposition analysis, mathematical formalism, optimization modeling, reinforcement learning, simulation modeling.The research employed a diverse array of methodologies, including decomposition analysis, mathematical formalism, optimization modeling, reinforcement learning, and simulation modeling.

Results obtained

This study presents an optimized algorithm for real-time processing of unstructured data streams, derived from the decomposition of tested models and the formalization of a generalized mathematical framework. By integrating Kafka with Zero-Copy I/O, CEEMDAN-based preprocessing, ONNX-inference models, knowledge graph enrichment, and QoS-driven orchestration (Kubernetes, KEDA), the proposed solution achieved a 3.8× reduction in Latency (from 250.0 to 65.0 ms), increased Accuracy to 91.3%, and doubled Throughput (1100 to 2200 events/sec). Additionally, CPU Load was reduced by 27%, Memory Usage by 45%, while the Adaptability Score and Semantic Alignment Score improved from 0.52 to 0.88 and from 0.64 to 0.91, respectively. These results confirm the algorithm's efficiency, scalability, and applicability in intelligent Decision Support Systems under real-time constraints.

Scientific novelty of the study

The scientific novelty of the research lies in the development of a mathematical model for streaming processing, characterized by dynamic ingesting, semantic enrichment, and optimization through reinforcement learning, all aimed at enhancing real-time productivity.

Prospects for future research
Further research involves validating the optimized algorithm within high-entropy streams, the scaling to multimodal datasets, and the enhancement through adaptive retraining and dynamic orchestration.

Keywords: *Kafka, Flink, Semantic Enrichment, Reinforcement Learning, RocksDB, CEEMDAN Decomposition, ONNX Inference.*

1. INTRODUCTION

The rapid proliferation of unstructured streaming data coming from diverse sources (logs, sensor systems, social networks, multimodal platforms) necessitates the development of highly efficient algorithms for their processing to facilitate real-time decision-making. Traditional batch-oriented analysis methods demonstrate limited applicability due to high latency, limited scalability, and lack of dynamic semantic adaptation of data streams. Against this backdrop, the selected research is aimed at formalizing a generalized mathematical model of stream-native processing of unstructured data, taking into account the aspects of ingestion, adaptive pre-processing, semantic enrichment, low-latency stream analytics, and QoS-aware orchestration. The proposed direction aims to transcend existing limitations of classical architectures, increase cognitive consistency of conclusions, and ensure real practical integration into smart Decision Support Systems.

The purpose of the study is to develop and optimize a generalized mathematical model for processing unstructured data streams to support real-time decision-making and evaluate its effectiveness based on key performance metrics.

Research objectives:

- conduct a decomposition analysis of tested models to ascertain the median component composition of unstructured data stream processing;
- formulate a generalized mathematical model of stream processing taking into account transformations, semantic enrichment, and state management;
- develop a multi-criteria objective optimization function based on performance metrics;
- implement adaptive optimization of streaming processing configurations based on reinforcement learning;
- conduct simulation modeling and comparative evaluation of the algorithm's effectiveness within a digital environment.

Key Terminology:

– *Unstructured Data* refers to information that lacks a predefined data model or schema, such as free-text documents, audio, video, sensor streams, or log files. It typically requires advanced parsing, semantic interpretation, or AI-driven transformation for effective analysis.
– *Streaming Processing* denotes the real-time or near-real-time processing of continuous data flows, enabling timely computation and decision-making. It contrasts with batch processing by supporting low-latency, incremental computation over dynamic datasets.
– *Semantic Enrichment* is the process of augmenting raw data streams with contextual meaning through metadata tagging, ontology mapping, or knowledge graph integration. This improves machine interpretability and analytical value.
– *Fault Tolerance* is the capability of a system to maintain functionality in the presence of partial failures, ensuring data continuity, resilience, and uninterrupted processing in high-throughput environments.
– *Real-Time Decision Support Systems (RT-DSS)* are intelligent computational architectures designed to ingest, process, and analyze live data streams for immediate decision-making. They rely on optimized pipelines, low-latency inference, and adaptive orchestration to support mission-critical operations.
– *Adaptive Orchestration* refers to the dynamic management of computational and data resources (e.g., via Kubernetes/KEDA) in response to workload fluctuations, ensuring efficiency, scalability, and QoS compliance.

2. LITERATURE REVIEW

The analytical review of current publications on streaming processing algorithms for analyzing unstructured data is presented as follows.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In this regard, Jiang et al. [1] showed the effectiveness of streaming unstructured data from Baidu Index through CEEMDAN decomposition, wavelet denoising, and SVR modeling for predicting carbon quota prices. The introduced random perturbation mechanism enhances the interpretability of the model and increases its accuracy in a multi-source flow.

Another area of data management was explored by Chuacharoen and Aiemsuwan [2], who conducted an empirical analysis of unstructured multimedia data streams in an online commerce environment based on behavioral metrics of social platform users. A management model was developed that takes into account the typology of storage media, cross-platform distribution, as well as data classification by media format under conditions of decentralized access.

A structural perspective of the problem under study is demonstrated by de Haan, et al. [3], who presented a three-section approach to integrating unstructured streams into managerial decision-making under conditions of high information entropy. The developed framework, grounded on organizational learning theory, classifies unstructured data usage scenarios by strategic purpose (exploration/exploitation) and source of environmental scanning (endogenous/exogenous), providing algorithmic support for the selection of methods and sources.

Furthermore, a systematic approach is exemplified by Farhanet al. [4], who proposed a four-layer Extract–Clean–Load–Transform (ECLT) architecture designed for high-performance processing of unstructured streams of large text data. Within the Spark platform, the model demonstrates a considerable reduction in execution time as compared to traditional ETL methodologies, facilitating efficient cleaning, pre-processing, and transformation of input streams with data volumes up to 1 TB.

Another attempt to address the challenges posed by large data sets is shown by Kaliyaperumal et al. [5], who introduced the BDQAM (Big Data Quality Assessment Model) model for real-time streaming quality assessment of unstructured big data. The model adeptly integrates machine learning algorithms and NLP tools to validate streams based on accuracy, completeness, consistency, and timeliness criteria, demonstrating enhanced efficiency as compared to traditional approaches to assessing data quality in decision support processes.

A comparative approach and optimization in the examined vector are demonstrated by Battaglia et al. [6], who conducted a constraint analysis of a two-stage approach to unstructured data analytics that combines upstream extraction of latent variables through information extraction models and downstream econometric regression. They propose a unified Bayesian inference architecture that facilitates joint parameter identification and asymptotic unbiased estimates, revealing substantial advantages through simulation modeling and applied analysis of behavioral economic data streams.

Next, Yu [7] proposed a multimodal improved Transformer architecture for streaming processing of high-dimensional unstructured data encompassing diverse modalities—such as text, images, audio, and video. The model utilizes the multi-threaded input with feature unification in a common spatial representation, a cross-modal self-attention mechanism to discern cross-modal dependencies, and sparse optimization for efficient processing of long sequences, demonstrating superiority in accuracy, precision, and recall metrics.

On the contrary, the problem of generating suitable datasets was examined by Liu et al. [8], who focused on generating streaming unstructured data derived from sensory modalities for autonomous navigation in complex environments. The created dataset provides realistic conditions for evaluating algorithms for streaming multimodal unstructured data (LiDAR, video, IMU), thereby facilitating the advancement of methodologies for online integration of sensor streams, real-time SLAM analysis, and semantic segmentation.

A novel algorithm for the systematic organization of the analyzed datasets was introduced by Sukumar et al. [9], who devised a streaming architecture for the extraction of information from unstructured textual data with subsequent construction of knowledge graphs. The proposed streaming NLP pipeline includes coreference resolution, entity linking, and relationship extraction, with automated storage in the Neo4j graph database, allowing for the transformation of unstructured text streams into real time structured semantic representations.

Encoding tools were utilized by Glyn-Davies et al. [10], who proposed Φ-DVAE (Physics-informed Dynamical Variational Autoencoder) for the streaming of unstructured data into dynamic physical models described by differential equations. The architecture combines nonlinear filtering of the latent state space with variational Bayesian estimation, thereby enabling the assimilation of video streams and velocity fields without an explicit observation operator while simultaneously recovering unknown parameters accompanied by uncertainty assessment.

The analytical review elucidates the vigorous advancement of stream processing algorithms tailored for the examination of unstructured data, encompassing a wide range of technologies — from decomposition methods (CEEMDAN, wavelet denoising) and regression models (SVR) to variational autoencoders (Φ-DVAE) and transformers featuring cross-modal self-attention. Cutting-edge architectures, including ECLT, multimodal Transformer models, NLP pipelines with graph semanticization (Neo4j), and Bayesian integration frameworks, exhibit remarkable efficacy within the realm of high-dimensional, heterogeneous, and semantically diverse streams. The above approaches not only enhance accuracy and interpretability but also afford scalable real-time processing capabilities, which are indispensable for informed decision-making in dynamic and data-centric environments.

The imperative for advancing unstructured data streaming processing algorithms is underscored by the constraints of existing solutions identified in the analyzed studies: susceptibility to complex semantics, reduced accuracy amid heterogeneous sources, restricted coverage of diverse media types, and inadequate adaptability of transformation mechanisms. Enhancements to the models should be directed towards increasing noise resistance, minimizing processing delays, broadening cross-modal representation, and facilitating automated quality management of input streams.

3. METHODS AND MATERIALS

3.1. Research procedure.

The scheme of this study is given below (Figure 1).

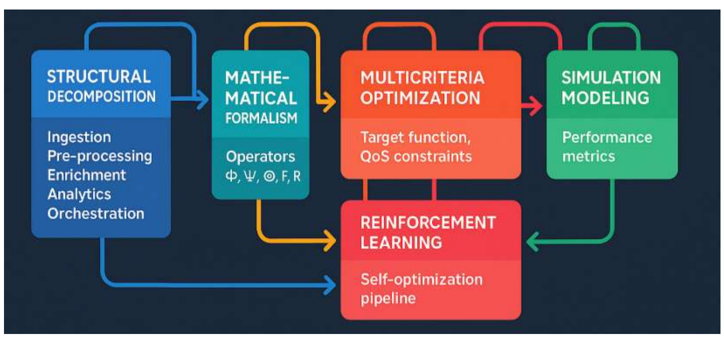

Figure 1. Research flowchart
Source: created by the authors.

3.2. Methods.

The following methods were used in this study:

1. Decomposition analysis method. A component decomposition of the tested models for processing unstructured data streams (Table 1) was performed, which allowed us to determine the median set of critical functional blocks (Table 3), including ingestion, pre-processing, semantic enrichment, analytics, and orchestration.
2. Method of mathematical formalism. A generalized mathematical model of processing unstructured data streams (equations (1)–(13)) is developed with the formalization of stream transformations, semantic enrichment, state management, and decision-making logic.
3. Optimization modeling method. A multi-criteria objective function (equation (14)) is developed to simultaneously minimize Latency, CPU Load, Memory Usage and maximize Accuracy, Throughput, and Fault Tolerance, taking into account the weighting coefficients of the objectives.
4. Reinforcement learning method. An approach for adaptive optimization of streaming

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

processing configurations through a self-adaptive agent is proposed, which implements an optimization strategy based on feedback regarding efficiency metrics (Equation (18)).

5. Simulation modeling method. A model calculation of the algorithm's efficiency was performed in the Python environment with an assessment of key metrics (Latency, Accuracy, Throughput, CPU Load, Memory Usage, Fault Tolerance, Adaptability Score, Semantic Alignment Score, Scalability Index) — the results are consolidated in Table 6.

3.3. Sample.

The research sample comprises models designed for the processing of unstructured data streams to support decision-making that were tested in real-life conditions (Table 1).

Table 1. Proven models for processing unstructured data streams to support decision-making

Model	Short description	Technologies used	Scope of application	Academic description in the context of the studied vector
Lambda Architecture	A combination of batch and stream processing with separate paths for calculations and merging of results.	Apache Hadoop (HDFS, MapReduce), Apache Storm for real-time, Kafka for message streams	Netflix, LinkedIn — fault-tolerant analytics of log/event data, advertising services	El Aissiet al. [11]
Kappa Architecture	A streaming processing model without batch components, focused on real time.	Kafka Streams API, Apache Flink for low-latency stateful stream computing	Uber, Spotify — real-time event processing, monitoring of traffic and user actions	Guan et al. [12]
Apache Kafka Streams	A framework for building streaming applications on top of Kafka with low latency.	Kafka Streams DSL + Processor API, integration with RocksDB for local state management	Goldman Sachs, Bank of America — fraud detection, audit trail analysis, KYC monitoring	Padmanaban et al. [13]
Spark Structured Streaming	Micro-batch processing of unstructured streams with fault tolerance support.	Apache Spark Structured Streaming, Catalyst Optimizer, HDFS, Kafka for ingestion	Airbnb, Alibaba — IoT streams, clickstream analytics, real-time user segmentation	Gupta and Yip [14]
Apache Flink CEP	Complex event processing with support for time windows and states.	Apache Flink CEP, event time processing, keyed state, windowed joins, Kafka integration	ING, Huawei — rule-based event correlation, financial anomaly detection	Mezati and Aouria [15]
Amazon Kinesis	Cloud-based streaming processing with auto-scaling.	Amazon Kinesis Data Streams, Firehose, Lambda for serverless processing, integration with S3, Redshift	Coca-Cola, Autodesk — video analytics, telemetry, IoT in retail networks	Koschel et al. [16]
Google Dataflow	Universal stream and batch processing with a single language, Apache Beam.	Apache Beam model, Google Cloud Dataflow Engine, integration with BigQuery, Pub/Sub	PayPal, Twitter — ETL and telemetry/pipeline data analytics, real-time ML inference	Ranganathan et al. [17]
ELK Stack (Logstash Streams)	Processing unstructured logs and real-time visualization.	Beats for streaming log collection, Logstash for ETL, Elasticsearch for indexing, Kibana for visualization	Cisco, eBay — log analytics, incident detection, infrastructure monitoring	Bakraouy et al., [18]
Azure Stream Analytics	SQL-driven cloud-based stream processing with Power BI integration.	Azure Event Hubs for event collection, Stream Analytics SQL, Power BI for real-time analytics	Heathrow Airport, Schneider Electric — Smart City dashboards, anomaly detection in power grids	Koschel et al. [19]

Model	Short description	Technologies used	Scope of application	Academic description in the context of the studied vector
Facebook Stream QL	High-speed processing of unstructured events in internal systems.	StreamQL query engine, RocksDB for state retention, internal distributed execution framework	Meta (Facebook) — internal message processing, recommendation systems, behavioral triggers	Kong & Mamouras [20]

Source: created by the authors

3.4. Tools.

The following set of metrics are used as tools for this study (Table 2).

Table 2. Metrics for assessing the effectiveness of algorithms for processing unstructured data streams to support decision-making

Metrics	Short description	Mathematical formulation
Latency	Average time between the moment an event is received and the moment it is processed by the system	$L = \frac{1}{N}\sum_{i=1}^{N}\left(t_{out,i} - t_{in,i}\right)$, where N is the number of events processed; $t_{in,i}$ is the time of occurrence of i-th event; $t_{out,i}$ is the time to obtain the processing result of i-th event
Accuracy	The ratio of the number of correctly predicted outcomes to the total number of solutions	$A = \frac{TP + TN}{TP + TN + FP + FN}$, where TP is true positive cases (correct prediction of positive); TN is true negative cases; FP is false positive cases; FN is false negative cases
Throughput	The number of events that the system can process per unit of time	$S = \frac{N}{T}$, where N is the number of processed events; T is the time during which events were processed
CPU Load	Average CPU usage percentage during thread processing	$R_c = \frac{1}{n}\sum_{i=1}^{n} CPU_i$, where CPU_i is CPU usage in i time moment; n is the number of measurements
Memory Usage	Average amount of RAM consumed by the system during the performance	$R_m = \frac{1}{n}\sum_{i=1}^{n} Mem_i$, where Mem_i is memory usage in i is the time moment; n is the number of measurements
Fault Tolerance	The ability of the system to recover from failures	$T_f = \frac{N_{restored}}{N_{restored} + N_{failed}}$, where $N_{restored}$ is the number of successfully processed events after recovery; N_{failed} is the number of events failed due to failure

Adaptability	The ability of the algorithm to change configuration and maintain efficiency when the environment changes	$A_d = \frac{1}{1+\Delta T_{adjust}} \times \frac{Q_{after}}{Q_{before}}$, where ΔT_{adjust} is the time required for adaptation; Q_{before}, Q_{after} is quality of the result (e.g., accuracy) before and after adaptation
Semantic Alignment Score	The degree of alignment of the derived results with the ontological model or knowledge	$S_a = \frac{\lvert E_{model} \cap E_{ontology} \rvert}{\lvert E_{model} \rvert}$, where E_{model} is the set of entities/relationships found by the model; $E_{ontology}$ is ontological reference set
Scalability Index	Efficiency of system expansion with increasing resources	$\sigma = \frac{\Delta S}{\Delta R_c + \Delta R_m}$, where ΔS is increase in throughput; ΔR_c, ΔR_m is increase in CPU and memory usage respectively

Source: created by the authors

The Python digital environment was used for optimization mathematical modeling using reinforcement learning methods.

4. RESULTS

In order to identify suitable solutions for the formulation of an effective model and algorithm for the processing of unstructured data streams to facilitate decision-making by the decomposition method of tested models (Table 1) was employed to ascertain the median component composition of software solutions of this type (Table 3).

Table 3. Median component composition of models and algorithms for processing the stream of unstructured data

Component	Description	Implementation examples
1. Data sources	Unstructured information streams (logs, video, audio, text, sensors)	Kafka, Event Hubs, IoT brokers
2. Ingestion	Stream capture and routing	Kafka, Kinesis Streams, Pub/Sub
3. Pre-processing	Normalization, cleaning, decomposition (wavelet, CEEMDAN), tokenization	Spark, Flink, Logstash
4. Enrichment	Annotation, entity extraction, ontological correspondence	NLP pipelines, NER, coreference, OBDA
5. Streaming analytics	CEP, ML-inference, fuzzy logic, rule-based engines	Flink CEP, Azure Stream Analytics, Spark ML
6. State/context storage	Stateful processing for aggregation, semantics, or case study analysis	RocksDB, Redis, NoSQL, GraphDB
7. Sink	Visualization, API, results storage	Elasticsearch, Power BI, BigQuery, Neo4j
8. Orchestration/Scaling	Autoscaling, fault tolerance, QoS guarantees	Kubernetes, autoscaling policies, serverless triggers

Source: elaborated by the authors.

For this component composition, we devised a generalized mathematical model for the streaming processing of unstructured data, aimed at supporting decision-making.

1. Definition of input stream. The stream of unstructured events at time:

$$D(t)=\{d_1(t),d_2(t),\ldots,d_n(t)\}, \quad (1)$$

Where $d_i(t)\in \mathrm{U}$ is an element of the unstructured data space (text, images, sound, video, sensory data).

2. Ingestion and filtration function. The ingestion mechanism can be specified as a routing function:

$$\Phi:\mathrm{U}\to \mathrm{S}\subseteq \mathrm{R}^m \quad (2)$$

where S is the space of semantically relevant signals obtained after the function as follows:.

$$\Phi=f_{broker}\circ f_{filter} \quad (3)$$

where f_{broker} is routing (Kafka, Pub/Sub); f_{filter} is predicative filtering.

3. Normalization and decomposition. The pre-processing function is as follows:

$$\Psi:\mathrm{S}\to \mathrm{R}^k \quad (4)$$

where the function:

$$\Psi=f_{denoise}\circ f_{decomp}\circ f_{norm} \quad (4)$$

where $f_{denoise}$ is wavelet filtering / CEEMDAN; f_{decomp} is wavelet - or EMD-decomposition; f_{norm} – tokenization / vectorization.

4. Semantic enrichment. The function as follows is used:

$$\Theta:\mathrm{R}^k\to \mathrm{G} \quad (6)$$

where G is knowledge graph: $\mathrm{G}=(V,E)$, $V=entities$, $E=relation$.

Through the application:

$$\Theta=f_{NER}\circ f_{coref}\circ f_{linking} \quad (5)$$

5. Stream analytics. Stream model of decision making:

$$\mathrm{D}(t)=R\Big(F\Big(\Theta\Big(\Psi\big(D(t)\big)\Big)\Big),\mathrm{C}\Big) \quad (6)$$

where R is decision-making logic (e.g. classification, alarm signal); F is analytical operator (ML model, rule engine, CEP); C are contextual variables (state, aggregates, history).

6. State management. The set of stream aggregates:

$$\Sigma(t)=\{\sigma_1(t),\sigma_2(t),\ldots,\sigma_n(t)\}=f_{state}(\mathrm{G},window) \quad (7)$$

where f_{state} is the aggregation in time windows w_t , sliding or tumbling.

7. Inference and operating logic. The action function is formalized as follows:

$$\Omega:\mathrm{D}(t)\times\Sigma(t)\to \mathrm{A} \quad (8)$$

where A is the space of possible actions (API calls, triggers, visualizations).

8. Optimization objective function. The goal is to minimize latency and maximize accuracy:

$$\min_{\Phi,\Psi,\Theta,F}\Big\lfloor \lambda_1\times \mathrm{E}[latency]-\lambda_2\times \mathrm{E}[accuracy]\Big\rfloor \quad (11)$$

under the limitations:

$$QoS_{throughput}\geq\tau \quad (12)$$

$$error_{decision}\leq\varepsilon \quad (13)$$

Using the obtained generalized mathematical model of processing the unstructured data streams to support decision-making and mathematical modeling methods, we will develop optimization solutions aimed at improving the performance of the existing algorithms (Table 1) in their median representation (Table 3):

1. The goal of optimization is to build an adaptive configuration of functional blocks. Φ, Ψ, Θ, F, R, which will ensure minimal delay L , maximum decision-making accuracy A , limited resource consumption (CPU,

memory, bandwidth), guaranteed fault tolerance T_f and scalability S.

2. Objective function. Multi-criteria objective function in the form of a weighted sum:

$$\min_{\Phi,\Psi,\Theta.F,R} \mathrm{J} = w_1 \times L + w_2 \times (1-A) + w_3 \times R_c + w \times R_m - w_5 \times T_f - w_6 \times S \quad (14)$$

where L is the average processing delay; A is the accuracy of the decision-making model; R_c is the CPU load; R_m is the memory usage; T_f is the number of restored/retained flows during a failure; S is system throughput; is weighting factors (importance of each goal), while $w_i \in [0,1]$, and $\sum w_1 = 1$.

3. Limitations. Introducing limitations regarding QoS (15), resource consumption (16) as well as for normalizing flows to avoid congestion (17):

$$L \le L_{\max};\ A \ge A_{\min};\ S \ge S_{\min};\ T_f \ge T_{f\min}; \quad (9)$$

$$R_c \le C_{\max};\ R_m \le M_{\max}; \quad (10)$$

$$\left\|\Psi\left(D(t)\right)\right\| \le \delta. \quad (11)$$

4. Optimization algorithm. Utilizing the Python digital environment and employing advanced reinforcement learning methods, we shall delineate a step-by-step algorithm for the optimization of a generalized mathematical model concerning the streaming processing of unstructured data, aimed at enhancing decision-making capabilities (1) - (13). The following steps are performed:

- preliminary configuration initialization: $\{\Phi_0, \Psi_0, \ldots\}$;
- metrics collection (Table 2);
- calculation of the objective function J (14);
- application of an adaptive change mechanism (reinforcement learning);
- repetition until stabilization or threshold is reached ε (13):

$$\theta_t \leftarrow \theta_{t-1} - \eta \nabla \mathrm{J}\left(\theta_{t-1}\right) \quad (12)$$

At the same time, we obtained a generalized mathematical representation of an optimized model for processing the streaming of unstructured data, aimed at enhancing decision-making capabilities.

$$\min_{\theta} \mathrm{J}(\theta) \Rightarrow \begin{cases} L(\theta) \le L_{\max}; \\ A(\theta) \ge A_{\min}; \\ R_c(\theta) \le C_{\max},\ R_m(\theta) \le M_{\max}; \\ T_f(\theta) \ge T_{f\min},\ S(\theta) \ge S_{\min}. \end{cases} \quad (13)$$

where $\theta = \{\Phi, \Psi, \Theta, F, R\}$ is a set of configuration parameters.

According to the generalized representation of the optimized unstructured data streaming processing model (19), the transformation of the generalized mathematical model (1) - (13) is obtained.

1. The input stream of unstructured data has a similar formulation (1).

2. Dynamic ingestion. The equation (3) takes the form:

$$\Phi_{opt} = f_{broker}^{dyn} \circ f_{filter}^{qos} \quad (20)$$

3. Adaptive pre-processing. The equation (5) takes the form:

$$\Psi_{opt} = f_{denoise}^{wavelet} \circ f_{decomp}^{adaptive} \circ f_{norm}^{stream} \quad (21)$$

4. Context-sensitive enrichment. The equation (7) takes the form:

$$\Theta_{opt} = f_{NER}^{context-aware} \circ f_{coref}^{lightweight} \circ f_{linking}^{semantic} \quad (22)$$

5. Optimized analytics. The equation (8) takes the form:

$$\mathrm{D}(t)=R_{opt}\left(F_{opt}\left(\Theta_{opt}\left(\Psi_{opt}\left(D(t)\right)\right)\right),\mathrm{C}\right) \quad (23)$$

6. State management. The equation (9) takes the form:

$$\Sigma(t)=\bigcup\sigma_j(t)=f_{state}^{sliding}\left(\mathrm{G},w_t\right) \quad (24)$$

7. Optimized action taking. The equation (10) takes the form:

$$\Omega_{opt}:\mathrm{D}(t)\times\Sigma(t)\xrightarrow{QoS}\mathrm{A}\,. \quad (25)$$

8. Objective function. The equation (11) takes the form:

$$\min_{\theta}\left[\begin{array}{l}\lambda_1\times\mathrm{E}\left[L(\theta)\right]+\lambda_2\times\mathrm{E}\left[R_c(\theta)\right]+\lambda_3\times\mathrm{E}\left[R_m(\theta)\right]+\\+\lambda_4\times\mathrm{E}\left[E_{dec}(\theta)\right]-\lambda_5\times\mathrm{E}\left[A(\theta)\right]\end{array}\right] \quad (26)$$

where $\theta=\left\{\Phi_{opt},\Psi_{opt},\Theta_{opt},F_{opt},R_{opt}\right\}$, with the restrictions as follows:

$$L(\theta)\le L_{\max}; \quad (27)$$

$$A(\theta)\ge A_{\min}; \quad (28)$$

$$S(\theta)\ge S_{\min}. \quad (29)$$

That being said, owing to the application of mathematical modeling methods, we developed an optimized algorithm for the processing of unstructured data streams to facilitate informed decision-making. The efficacy of this algorithm can be further enhanced through the multi-component integration of cutting-edge technologies (Table 4).

Table 4. Median component composition of models and algorithms for processing the stream of unstructured

Optimization direction	Technological solution	Technologies involved
Optimization of processing infrastructure. The goal is low latency.	Kafka + Zero-Copy I/O	– Flink + Event Time Processing – Data locality-aware scheduling (Spark/Flink) | | Scalability | - Kubernetes autoscaling – Partitioned state in Kafka Streams / Flink – Serverless computing (AWS Lambda, Google Cloud Functions) | | Fault Tolerance | - Write-ahead logs (Kafka, Flink checkpoints) – RocksDB-backed state storage – Replication / failover strategies |
Optimization of pre-processing and analytics. The goal is reduction of the computational load.	CEEMDAN + wavelet denoising (signal separation)	– Approximate computing / sampling – Vectorization (spaCy, BERT-tokenizer) | | NLP/NLU optimization | - Lightweight models (DistilBERT, MiniLM) – Batch inference – ONNX/TensorRT acceleration | | Complex Event Processing (CEP) | - Apache Flink CEP – Azure Stream Analytics Rule Engine – Esper for pattern detection |
State and aggregation management. The goal is efficient	RocksDB (Kafka/Flink state backend)	– Cassandra / Redis for aggregates – Windowed joins with pre-aggregation | | Contextual adaptation | - Dynamic window sizing – Adaptive stream joins

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Optimization direction	Technological solution	Technologies involved
storage.		– Semantic state compaction (GraphDB/Neo4j) |
ML/AI inference optimization. The goal is speeding up the model.	ONNX Runtime	– TensorRT – TFLite / Edge TPU | | Response time reduction | - Model caching – Batching inference – Stream inference via TensorFlow Serving | | Automatic tuning | - AutoML (Google Vertex AI, H2O AutoML) – Reinforcement learning for pipeline optimization – Hyperparameter optimization (Bayesian search, Optuna) |
QoS orchestration and control. The goal is auto-scaling.	Kubernetes HPA/VPA	– KEDA (Kubernetes Event-Driven Autoscaling) – Apache YARN capacity scheduler | | Monitoring and control | - Prometheus + Grafana – Elastic APM / New Relic – Flink Metrics / Spark UI | | Prioritization system | - QoS queues – Backpressure handling (Flink, Kafka) – Stream partition rebalancing
Semantic optimization. The goal is semantic merging of streams.	RDF Stream Processing (C-SPARQL, CQELS)	– Knowledge Graph Fusion – Context-aware reasoning | | Ontological compression | - OWL-based pruning – Top-k subgraph extractions – Reasoning cache layers

Source: elaborated by the authors.

Optimization of the generalized algorithm for processing the unstructured data streams was carried out by modeling targeted technological solutions at each functional stage (Table 5).

Table 5. Comparative decomposition analysis of solutions for optimizing the algorithm for processing the unstructured streams

Optimization direction	Component	Generalized algorithm	Optimized algorithm
Ingestion and routing	Brokers	Apache Kafka	Kafka + partition rebalance + QoS-aware topics
	Filtration	Fixed rules	Dynamic filtering based on a policy engine (e.g. Flink filters with QoS SLA)
	Buffering	FIFO queues	Backpressure-aware ingestion (Flink, Kinesis)
Pre-processing	Cleaning/Denoising	Wavelet	CEEMDAN + adaptive wavelet decomposition
	Vectorization	TF-IDF / static	DistilBERT / MiniLM via ONNX or TensorRT
	Parallelism	Batch processing	Stream-aware pipeline (micro-batch / stateful operators)
Semantic enrichment	NER/Linking	Room, spaCy	context-aware NER (REBEL) + entity pruning
	Coreference	Rule-based	Lightweight neural coreference (NeuralCoref, SpanBERT)
	Knowledge graphs	Neo4j regular	Graph embedding + top-k RDF slicing + reasoning cache
Analytics and decision making	Model	XGBoost / Logistic Regression	DistilBERT, GRU, Transformers with priority processing
	Inference	Full precision	Quantized / ONNX / TensorRT inference
	Decision making	Static rules	Reinforcement learning / Policy optimization
	CEP	Fixed templates	Flink CEP with dynamic pattern update

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Optimization direction	Component	Generalized algorithm	Optimized algorithm
State management	Status repository	Redis / PostgreSQL	RocksDB + pre-aggregated sliding windows
	Windows	Fixed size	Adaptive sliding / session-based windows
Orchestration, Scaling, and Fault Tolerance	Containerization	Docker	Kubernetes + KEDA autoscaling
	Monitoring	Manual	Prometheus + Grafana + dynamic QoS alerts
	Scaling	Vertical	Horizontal event-driven (Flink, Kinesis, Dataflow)

Source: elaborated by the authors.

Thus, the optimized algorithm (Table 4, Table 5) implements adaptive stream processing through the utilization of compact neural models (ONNX, DistilBERT), asynchronous event processing (Kafka, Flink), QoS management via backpressure and dynamic rebalancing, stateful processing based on RocksDB, and semantic alignment using knowledge graphs and reasoning cache to enhance the cognitive consistency of conclusions.

In order to practically evaluate the optimized algorithm for processing unstructured data streams to support decision-making, a model calculation was performed using efficiency metrics (Table 2) in the Python programming environment (Table 6).

Table 6. Comparative calculation of efficiency metrics of generalized and optimized streaming algorithms for unstructured data analysis

Performance metrics	Model values of the generalized algorithm	Model values of the optimized algorithm
Latency (ms)	250.0	65.0
Accuracy (%)	82.5	91.3
Throughput (events/sec)	1100.0	2200.0
CPU Load (%)	75.0	48.0
Memory Usage (MB)	1850.0	1020.0
Fault Tolerance (%)	86.0	96.5
Adaptability Score	0.52	0.88
Semantic Alignment Score	0.64	0.91
Scalability Index	1.4	3.1

Source: elaborated by the authors.

Hence, the results of the comparative assessment (Table 6) substantiate that the optimized algorithm provides a significant increase in the operational efficiency of unstructured data streaming processing systems, ensuring the minimization of latency, enhanced accuracy of analytics, increased throughput, optimization of of computing resources, and improved adaptive-semantic features. These characteristics validate its viability for practical implementation within real-time smart decision support systems.

5. DISCUSSION

Let's juxtapose the obtained results in terms of compliance with the actual scientometric horizon.

Compared to the Turet & Costa [21] method, which achieved 80% accuracy through data structuring, the optimized algorithm achieved 91.3% accuracy through adaptive processing and semantic integration. This indicates the higher efficiency of real-time streaming methods without the need for full data transformation.

Mehmood & Anees [22] optimized real-time ETL by accelerating stream-disk joins, reducing disk load. Accordingly, the optimized algorithm achieves higher efficiency through asynchronous processing and semantic matching without direct dependence on disk operations.

Kumar et al. [23] focused on content management of unstructured data with the subsequent application of analytical engines. The optimized algorithm implements streaming processing with integrated semantic aggregation and real-time inference without intermediate storage in content repositories.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Koschmider et al. [24] examined the application of process mining to analyze unstructured data, focusing on challenges and underlying solutions. Instead, the optimized algorithm implements semantic streaming processing with real-time event correlation without the classic process discovery stage.

The Maxima [25] study proposed integrating unstructured data into RDBMS via LSTM classification to improve analysis accuracy. The optimized algorithm provides stream-native processing with ontological aggregation and real-time inference without transactional structuring.

Mollá et al. [26] developed an adaptive DSS that incrementally integrates data streams for explainable inferences in unstable environments. The optimized algorithm instead implements stream-oriented processing with real-time semantic enrichment and predictive reasoning without post-processing stages.

Jain & Fallon [27] introduced UDNet, a multi-strategy ML framework for processing multimodal unstructured data using Adaptive Confidence Bound and Multi-Fidelity Meta-Learning, whereas the optimized algorithm provides stream-native processing with real-time model selection and semantic fusion without a multi-layered AutoML orchestration strategy.

König et al. [28] undertook a systematic review of the unstructured data utilization in process mining, elucidating the dominance of textual sources and an emphasis on event extraction. On the other hand, the developed optimized algorithm performs continuous stream-native event enrichment and ontological matching without the need for prior formation of classical event logs.

Richmond [29] investigated the application of transformer models, RNNs, and GNNs to stream processing of unstructured social media data with a focus on sentiment analysis and trend forecasting. In turn, the optimized algorithm implements low-latency multimodal fusion and semantic stream reasoning without classical staged processing of text data.

Methuku [30] developed an AI/NLP architecture for real-time epidemiological monitoring based on NER, sentiment analysis, and geospatial trend mining. The optimized algorithm implements continuous multilingual stream parsing and low-latency anomaly detection without a separate static aggregation phase.

A comparative analysis of the considered approaches shows that most models for processing unstructured data streams are based on transactional structuring, content management systems, batch-oriented transformation, or process-oriented event log extraction. Accordingly, the optimized algorithm implements asynchronous stream-native processing with low-latency semantic fusion, dynamic event enrichment, multilingual stream parsing, adaptive anomaly detection, and real-time predictive reasoning without staged aggregation or multi-level AutoML adaptation, which provides increased throughput, fault-tolerance, cognitive interpretability, and scalability in real-time modes.

6. LIMITATIONS

The current study is limited by the lack of validation in high-entropy streaming environments and dynamic concept drift. In addition, the algorithm has not been stress-tested on ultra-scale heterogeneous datasets with a variable ontological structure of streams.

7. RECOMMENDATIONS

It is recommended to expand the validation of the optimized algorithm in high-entropy, multi-source streaming conditions with active control of concept drift and ontological evolution. It is also expedient to conduct scalable experiments on ultra-large multimodal datasets using dynamic resource orchestration and adaptive retraining strategies.

8. CONCLUSION

Based on the decomposition of tested models of streaming processing of unstructured data (Table 1) and the formalization of the mathematical apparatus (equations (1)–(29)), an optimized algorithm was developed, focused on minimizing Latency, maximizing Accuracy, and ensuring a high Scalability Index with guaranteed Fault Tolerance. Optimization was achieved through the use of asynchronous ingesting infrastructure (Kafka + Zero-Copy I/O), adaptive pre-processing (CEEMDAN, wavelet denoising), lightweight inference models (ONNX, DistilBERT), semantic enrichment through knowledge graphs and dynamic QoS orchestration (Kubernetes, KEDA). The results of comparative modeling in the Python environment (Table 6) recorded a 3.8-fold decrease in Latency (from 250.0 to 65.0 ms), an increase in Accuracy to 91.3%, a doubling of Throughput (from 1100 to 2200 events/sec), a 27% decrease in CPU Load and 45% in Memory Usage, as well as a significant improvement in Adaptability Score (from 0.52 to 0.88) and Semantic Alignment Score (from 0.64 to 0.91). Thus, the optimized algorithm demonstrated enhanced cognitive consistency, operational efficiency, and suitability for application in real-time intelligent Decision Support Systems.

The scientific novelty of the current research lies in the formalization of a mathematical model of streaming processing of unstructured data with dynamic ingestion, semantic enrichment, and optimization based on reinforcement learning, which ensures minimized Latency, increased Accuracy and Fault Tolerance in real time.

The practical value of the results obtained lies in the development of an optimized algorithm that provides increased throughput, reduced computational costs, and increased adaptability, which allows its effective application in Smart City, financial analytics, medicine, and cybersecurity systems.

REFERENCES:

[1]. M. Jiang, J. Che, S. Li, K. Hu, and Y. Xu, “Incorporating key features from structured and unstructured data for enhanced carbon trading price forecasting with interpretability analysis”, *Applied Energy*, Vol. 382, 2025. https://doi.org/10.1016/j.apenergy.2025.125301

[2]. O. Chuacharoen, and P. Aiemsuwan, (2025). “Unstructured data management model for online businesses in the new normal era”, *International Journal of Education and Literacy Studies*, Vol. 13, No. 2, pp. 642-648. https://doi.org/10.7575/aiac.ijels.v.13n.2p.642

[3]. E.de Haan, M. Padigar, S. El Kihal, R.Kübler, and J. E. Wieringa, “Unstructured data research in business: Toward a structured approach”, *Journal of Business Research*, Vol. 177, 2024. https://doi.org/10.1016/j.jbusres.2024.114655

[4]. M. S. Farhan, A. Youssef, and L. Abdelhamid, “A model for enhancing unstructured big data warehouse execution time”, *Big Data and Cognitive Computing*, Vol. 8, No. 2, 2024. https://doi.org/10.3390/bdcc8020017

[5]. G. Kaliyaperumal, A. Gatla, S. Lakhanpal, K. J. Reddy, R. Najm, and K. B. Raju, “Adaptive framework for comprehensive quality assessment in unstructured big data”, In 2024 7th international conference on contemporary computing and informatics, September 18-20, 2024. https://doi.org/10.1109/ic3i61595.2024.10829192

[6]. L. Battaglia, T. Christensen, S. Hansen, and S. Sacher, “Inference for regression with variables generated from unstructured data”, *SSRN Electronic Journal,* 2024. https://doi.org/10.2139/ssrn.4842086

[7]. H. Yu, Research on feature extraction and classification for unstructured data based on deep learning. In 2024 5th international conference on artificial intelligence and computer engineering (ICAICE), 2024. https://doi.org/10.1109/icaice63571.2024.10864093

[8]. Y. Liu, Y. Fu, M. Qin, Y. Xu, B. Xu, F. Chen, B. Goossens, P. Z.H. Sun, H. Yu, C. Liu, L. Chen, W. Tao, H. Zhao, BotanicGarden: A high-quality dataset for robot navigation in unstructured natural environments. *IEEE Robotics and Automation Letters*, Vol. 9, No. 3, 2024. https://doi.org/10.1109/lra.2024.3359548

[9]. S. T. Sukumar, C. H. Lung, M. Zaman, and R. Panday, “Knowledge graph generation and application for unstructured data using data processing pipeline”, *IEEE Access*, Vol. 12, pp. 136759 – 136770, 2024. https://doi.org/10.1109/access.2024.3462635

[10]. A. Glyn-Davies, C. Duffin, O. Deniz Akyildiz, and M. Girolami, “Φ -DVAE: Physics-informed dynamical variational autoencoders for unstructured data assimilation”, *Journal of Computational Physics*, Vol. 515, 2024. https://doi.org/10.1016/j.jcp.2024.113293

[11]. M. E. M. El Aissi, S. Benjelloun, Y. Lakhrissi, and S. E. H. B. Ali, “A scalable smart farming big data platform for real-time and batch processing based on lambda architecture”, *Journal of System and Management Sciences*, Vol. 13, No. 2, 2023. https://doi.org/10.33168/jsms.2023.0202

[12]. M. Guan, J. Liu, , and X. Wang, “Design of large-scale bird prediction system based on kappa architecture in digital twin airport”, In 2024 IEEE 6th international conference on civil aviation safety and information technology (ICCASIT), October, 23-25, 2024.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

https://doi.org/10.1109/iccasit62299.2024.10828005

[13]. K. Padmanaban, T. R. Ganesh Babu, K. Karthika, B. K. D. Pattanaik, and C. Srinivasan, "Apache kafka on big data event streaming for enhanced data flows". In 2024 8th international conference on I-SMAC (iot in social, mobile, analytics and cloud) (I-SMAC), October 3-5, 2024. https://doi.org/10.1109/ismac61858.2024.10714884

[14]. N. Gupta, and J. Yip, "Spark structured streaming: A comprehensive guide". In *Databricks data intelligence platform*, pp. 409–429, Apress, 2024. https://doi.org/10.1007/979-8-8688-0444-1_18

[15]. M. Mezati, and I. Aouria, "Flink-ML: Machine learning in Apache Flink", *Brazilian Journal of Technology*, Vol. 7, No. 4, 2024. https://doi.org/10.38152/bjtv7n4-015

[16]. A. Koschel, I. Astrova, A. Pakosch, C. Gerner, C. Schulze, and M. Tyca, "Is amazon kinesis data analytics suitable as core for an event processing network model?" *Proceedings of the 16th international conference on agents and artificial intelligence*, Vol. 3, 2024. https://doi.org/10.5220/0012432800003636

[17]. C. S. Ranganathan, , Raman, R., Pandey, V. K., Kavitha, S., Muthulekshmi, M., & Gopalakrishnan, K. Ingestion of google cloud platform data using dataflow. In 2023 7th international conference on I-SMAC (iot in social, mobile, analytics and cloud) (I-SMAC), October 11-13, 2023. https://doi.org/10.1109/ismac58438.2023.10290190

[18]. Z. Bakraouy, W. Abbass, A. Baina, and M. Bellafkih, "Enhancing security through data analysis and visualization with ELK". In Communications in computer and information science, pp. 246–258, Springer Nature Switzerland, 2024. https://doi.org/10.1007/978-3-031-77040-1_18

[19]. A. Koschel, A. Pakosch, C. Schulze, I. Astrova, C. Gerner, and M. Tyca, "Would microsoft azure stream analytics be a suitable foundation for an event processing network model?" *Proceedings of the 17th international conference on agents and artificial intelligence*, Vol 1, pp. 15-22, 2025. https://doi.org/10.5220/0013073000003890

[20]. L. Kong, and K. Mamouras, "StreamQL: A query language for processing streaming time series", *Proceedings of the ACM on Programming Languages*, Vol. 4, pp. 1–32, 2020. https://doi.org/10.1145/3428251

[21]. J. G. Turet, A. P. C. S. Costa, "Hybrid methodology for analysis of structured and unstructured data to support decision-making in public security", *Data & Knowledge Engineering*, Vol. 141, 2022. https://doi.org/10.1016/j.datak.2022.102056

[22]. E. Mehmood, and T. Anees, "Distributed real-time ETL architecture for unstructured big data", *Knowledge and Information Systems,* Vol. 64, pp. 3419–3445, 2022. https://doi.org/10.1007/s10115-022-01757-7

[23]. P. Kumar, A. Tveritnev, S. A. Jan, and R. Iqbal, "Challenges to opportunity: Getting value out of unstructured data management", In Gas & oil technology showcase and conference. SPE, March 13, 2023. https://doi.org/10.2118/214251-ms

[24]. A. Koschmider, M. Aleknonytė-Resch, F. Fonger, C. Imenkamp, A. Lepsien, K. Apaydin, , M. Harms, D. Janssen, D. Langhammer, T. Ziolkowski, and Y. Zisgen, "Process mining for unstructured data: Challenges and research directions", *arXiv preprint arXiv:2401.13677,* 2023. https://doi.org/10.18420/modellierung2024_012

[25]. A. Maxima, "Integration and analysis of unstructured data towards database optimization and decision making using deep learning techniques (Doctoral dissertation)", Kampala International University, 2024. Available in: http://hdl.handle.net/20.500.12493/14548 (03.05.2025).

[26]. N. Mollá, A. Bossler, A. Rabasa, "Data stream solution for decision-making processes: A general and adaptive system for decision support", *Journal of Decision Systems*, pp. 1–12, 2024. https://doi.org/10.1080/12460125.2024.2354590

[27]. S. Jain, and Fallon, E. "UDNet: A unified deep learning-based automl framework to execute multiple ML strategies for multi-modal unstructured data processing", *IEEE Access*, Vol. 1, 2024. https://doi.org/10.1109/access.2024.3403724

[28]. F. König, A. Egger, W. Kratsch, M. Röglinger, and N. Wördehoff, Unstructured data in process mining: A systematic literature review. *ACM Transactions on Management*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Information Systems, Vol. 16, No. 3, 2025. https://doi.org/10.1145/3727148

[29]. E. H. Richmond, "Advanced techniques in natural language processing and deep learning for unstructured data analysis with a focus on real-time sentiment analysis and trend prediction in social media platforms", *International Journal of Multimedia Research (QITP-IJMMR)*, Vol. 5, No. 1, pp. 1-8, 2025.

[30]. V. Methuku, "NLP and AI for public health intelligence: Automating disease surveillance from unstructured data", *IECE Transactions on Emerging Topics in Artificial Intelligence*, Vol. 2, No. 1, pp. 43–56, 2025. https://doi.org/10.62762/tetai.2025.222799