International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 7, 2025
https://theaspd.com/index.php

An Explainable Deep Learning Based Data Mining Framework for Automated Data Loading Optimization and Pipeline Evaluation Using Sentiment Analysis

Anisha S[1], Dr. S Thiyagarajan[2]
[1]Department of Computer Science, St. Joseph University, Dimapur, Nagaland, India
anisha0007@gmail.com
[2]Department of Computer Science, St. Joseph University, Dimapur, Nagaland, India, s.tyagarajin@gmail.com

Abstract: *Most contemporary deep learning deployments are plagued by inefficiencies in pipeline setup and data mining, especially with large-scale, semi-structured, and unstructured text-based datasets. Non-adaptive nature of traditional systems means they need frequent manual interventions to sync data preprocessing with changing model needs. The limitations are addressed through this research by proposing AutoDL-Pipeline, an intelligent, modular system that combines adaptive pipeline training and automated data integration. It uses the Improved Rüppell's Fox Optimizer (IRFO) to dynamically manage data flow operations such as load scheduling, batch optimization, and schema flexibility. A hybrid storage architecture is used in order to support both archival requirements and real-time analytical processing. In the center of the pipeline, a Multilayer Interactive RoBERTa based Bidirectional Convolutional Gated Recurrent Unit (MIRBCGRU) model is used to analyze the effect of data quality on the classification result. The system integrates explainability modules such as Local interpretable model-agnostic explanations (LIME) to provide complete transparency to both loading and prediction stages. Experiments across IMDb, Yelp, Amazon Reviews, and Sentiment140 datasets show statistically significant improvements ($p < 0.05$) in accuracy, time cost, and stability, with Kendall's W values between 0.74 and 0.78 confirming consistent superiority.Finally, the AutoDL-Pipeline not only streamlines end-to-end deep learning pipelines but also offers transparent and scalable solutions to practical data pipeline problems—filling an essential gap in automated model life cycle management.*

Keywords: *Data mining, Deep Learning, Explainable AI, Sentiment Analysis, Data loading automation, Improved Optimizer*

1. INTRODUCTION

In today's data-centric world, the ability to derive meaningful insights from vast and heterogeneous datasets and data mining is critical to decision-making across industries. Data mining, as a core analytical discipline, plays a pivotal role in identifying hidden patterns, relationships, and trends from structured and unstructured data sources. One of its most influential applications is in the area of sentiment analysis, where it facilitates the interpretation of human emotions and opinions expressed in natural language—commonly extracted from platforms such as social media, online reviews, customer feedback, and forums. By identifying subjective content, sentiment-oriented data mining empowers businesses, governments, and researchers to make informed, sentiment-driven decisions [1].The advancement of Machine Learning (ML) and Deep Learning (DL) has significantly expanded the analytical power of sentiment analysis systems. These models thrive on large-scale, well-preprocessed datasets that enable them to learn complex linguistic and contextual patterns. However, achieving this level of performance is heavily dependent on the reliability of the data pipeline—a structured process responsible for acquiring, preparing, transforming, and delivering data to learning models. An inefficient pipeline can degrade model accuracy, slow down processing, and introduce inconsistencies that misrepresent real-world sentiment patterns [2, 3].
Despite the growing importance of data pipelines, current implementations often face challenges related to scalability, adaptability, and maintainability. Many are manually constructed and rigid, making them poorly suited to handle dynamic environments where data formats evolve rapidly and real-time insights are essential. In the context of sentiment analysis, where data is often noisy, unstructured, and context-dependent, these limitations become even more pronounced. Pipelines that cannot respond to shifts in vocabulary, expression styles, or schema changes risk undermining the entire sentiment inference process. Moreover, the lack of automation and transparency in traditional pipelines hampers efficiency and interpretability. Simultaneously, most DL-based sentiment systems operate as black boxes, offering little

to no explanation of how data is processed or how predictions are derived—raising concerns about trust and accountability, especially in domains like politics, healthcare, and finance [4, 5].
The streamlining of data flow was investigated by **Wu J et al. [6]** using the incorporation of AutoML processes within data pipelines. AutoML was used to enhance pipeline intelligence and flexibility, which resulted in improved performance in ML activities. Attempts were made to automate and optimize data management, allowing pipelines to adapt to changing data ecosystems. Consequently, faster model creation and creative solutions to difficult data problems were facilitated across different fields. **Haddad O et al. [7]** combined DL with batch and streaming analytics for sentiment forecasting. Distributed platforms such as Hadoop and Spark were utilized to preprocess big streaming data using cleaning, reduction, and optimization. NLP methods were employed for analyzing short text data to derive semantic context and sentiments. Word embeddings were created using GloVe and passed on to convolutional and recurrent neural networks to enhance prediction accuracy. A 41-factor taxonomy of factors impacting data pipeline quality was proposed by **Foidl H et al. [8]** in a multivocal literature review and was verified through interviews of eight data engineering experts. The factors impacting were organized under five broad themes: data, infrastructure, lifecycle management, development and deployment, and processing. Root causes of data issues were further explored in terms of their occurrences in various phases of data pipelines. Common issues faced by developers in data processing were determined via mining GitHub repositories and Stack Overflow discussions.A general architecture, named as FRTSPS, was suggested by **Deepthi BG et al. [9]** based on the popular Lambda architecture for big data processing. It was realized in a distributed computing setup, where Apache Flink was utilized for data processing activities. Flink was noted for handling real-time and historical data streams together. Its stateful stream processing was discovered to provide higher scalability, flexibility, high throughput, and lower latency than other models of stream processing.A new autonomous ETL pipeline architecture, FlowETL, was proposed by **Di Profio M et al. [10]** to standardize and prepare input datasets automatically according to a target format specified by a user. The system was implemented as a network of intercommunicating components that collaborated to accomplish data transformation objectives. A Planning Engine was used to create transformation plans by examining paired sets of source and target datasets as examples. These plans were run by an ETL employee on the entire dataset. Monitoring and logging functionality was added to guarantee observability and traceability across the pipeline run.
Avornicului MC et al. [11] created a modular and high-performance prototype of real-time event extraction to overcome unstructured data processing difficulties in areas such as crisis management and social media monitoring. Methods such as TF-IDF, LSI, and NER were combined with data mining techniques to enhance relevance scoring, clustering, and entity recognition. Contrary to transformer-based models limited by latency, this hybrid pipeline was developed to provide higher scalability and real-time performance. An end-to-end framework was presented by **Alotaibi A and Nadeem F [12]** that incorporated unsupervised BERTopic-based aspect identification with sentiment classification through a fine-tuned CAMeLBERT model, and summarization through a fine-tuned AraBART model. A large Arabic traffic services dataset was established, and a whole pipeline for aspect identification, sentiment analysis, and summarization was used without manual annotation. A platform was created by **Korontanis I [13]** in order to fulfill deployment and scalability requirements through the use of technologies like K3s, RabbitMQ, and KEDA. It was discovered to be superior to conventional systems by providing enhanced flexibility and user experience. Automated configuration was utilized in order to simplify the deployment of user-specified functions, which increased efficiency and minimized manual configuration requirements. This enabled developers to concentrate on developing functions without handling intricate configurations. Orange software was tested by **Dobesova Z [14]** to determine whether it can be used in schools when integrated with ArcGIS Pro to process spatial data. Through the application of Moody's Physics of Notation, its visual vocabulary was analyzed on all nine principles and found to have high cognitive effectiveness because of semantic transparency and appropriate dual coding. Practical exercises, including clustering the cafés in Olomouc and the analysis of European towns in land use, were shown to geoinformatics students and teachers, demonstrating the program's capability to teach ML exercises. A framework based on DL was introduced by **Zhang Z [15]** to classify UX user needs based on the Kano model and text analysis. User needs of Gen Z LEGO users were gathered, categorized through surveys, and utilized to train models using annotated online reviews. The RCNN model demonstrated better performance, allowing effective classification in a mature UX tool. An overview of the existing review is displayed in Table 1.

Table 1: Summary of Reviewed Works

Refs	**Technologies / Tools Used**	**Key Findings**	**Limitations**
[6]	AutoML, Data Pipelines	AutoML improved adaptability and intelligence in pipelines, accelerating model development.	Scalability and adaptability in highly dynamic environments were not fully validated.
[7]	Hadoop, Spark, NLP, GloVe, CNN, RNN	Enabled sentiment forecasting by integrating DL with batch and streaming analytics.	Limited to short text; dependency on quality of word embeddings.
[8]	Multivocal Literature Review, Expert Interviews	Proposed a 41-factor taxonomy affecting pipeline quality under five themes.	Empirical validation across industries was minimal.
[9]	Apache Flink, Lambda Architecture	Flink-based FRTSPS achieved real-time and historical stream processing with high throughput.	Generalization to non-Lambda architectures remains unexplored.
[10]	FlowETL, ETL Planning Engine	Autonomous ETL pipelines achieved standardization and transformation with traceability.	Requires example-based training; limited automation for highly diverse datasets.
[11]	TF-IDF, LSI, NER, Data Mining	Real-time event extraction prototype showed scalability and improved unstructured data handling.	Limited use of deep transformer models due to latency.
[12]	BERTopic, CAMeLBERT, AraBART	Built a full Arabic NLP pipeline for aspect detection, sentiment, and summarization.	Dataset and findings restricted to Arabic language and traffic service domain.
[13]	K3s, RabbitMQ, KEDA	Enabled scalable function deployment with simplified auto-configuration.	Limited evaluation in large-scale or high-load environments.
[14]	Orange, ArcGIS Pro, Physics of Notation	Demonstrated Orange's applicability in spatial data teaching with high cognitive effectiveness.	Targeted only educational settings; lacks industrial deployment insight.
[15]	RCNN, Kano Model, DL	UX needs of Gen Z classified accurately using RCNN-based tool.	Domain-specific to LEGO users; model generalizability not discussed.

1.1 Research gaps

Existing data pipeline automation and training models have a number of key limitations. Foremost among these is their inability to be easily adaptable, scalable, and have an intelligent self-optimization component—especially in environments where unstructured, large-volume, and heterogeneous data sources are involved. Human configuration is still a persistent bottleneck with the need for extensive intervention due to the inability of completely autonomous transformation within those frameworks. Current solutions also have poor interpretability and rigid automation structures that constrain performance in fast-changing data settings.Also, most methods are based on static templates or examples of ETL workloads, which makes them ineffective for a variety of datasets. Event processing pipelines for real-time tend to shun deep transformer models due to latency issues, thus sacrificing performance for semantic-intensive tasks. Additionally, pedagogically efficient tools, though cognitively useful, are seldom production-scale workload-optimized. There is a strong need for filling the gap in integrating intelligent training methodologies, optimization methods, and explainable AI (XAI) into pipeline systems. An end-to-end framework to cover these needs—tested on real-world tasks like sentiment classification—is a gap still unmet.

1.2 Research questions and hypothesis

Research questions:

i. How can automation be efficiently incorporated into DL pipelines in order to minimize human involvement and maximize model accuracy?
ii. How does IRFO contribute to real-time data adaptability and batch optimization?
iii. How does the integration of XAI increase the interpretability of the data pipeline as well as model predictions?
iv. Can a data lake-warehouse hybrid architecture efficiently handle both archival and active processing within DL workflows?

Researchhypothesis:

i. The incorporation of an automated loading process through the IRFO has a substantial impact on data throughput, pipeline latency, and response to schema changes compared to static loading methods.
ii. Using a hybrid data lake-warehouse architecture enhances archival productivity and real-time data availability, catering to varied DL workflows better than one-mode storage systems.
iii. Sentiment classification using the MIRBCGRU model results in better performance (accuracy, precision, recall, F1-score, and AUC-ROC) than traditional DL models in terms of handling contextual and sequential dependencies.
iv. Combining Explainable AI elements like LIME into the pipeline increases user interpretability and transparency of the data processing steps and ultimate model predictions, promoting trust and accountability within automated decision-making systems.

1.3 Research Contributions

➢ **Development of AutoDL-Pipeline:** This work presents a novel modular framework that seamlessly integrates data ingestion and pipeline training. Unlike traditional systems that require manual setup and tuning, AutoDL-Pipeline automates the entire data flow, thereby reducing human intervention and minimizing configuration errors across varied data formats.

➢ **IRFO for Data Loading Optimization:** The suggested strategy uses the IRFO to deal with batch sizing dynamically, load scheduling dynamically, and schema fluctuation dynamically. This strategy is more adaptable and efficient compared to traditional static data loading approaches that predominantly fail in dynamic environments with varying data structures.

➢ **Improved Sentiment Classification through MIRBCGRU:** A novel DL framework called MIRBCGRU is utilized to improve the capturing of semantic and temporal properties from text data. This remedies the shortcomings of prior models that fare poorly with either contextual richness or temporal dependence.

➢ **Hybrid Data Lake-Warehouse Architecture:** The pipeline output is stored in a dual-mode system. This mode is supporting long-term archiving and has real-time query processing capacities. Traditional single-mode storage systems are beaten out by this architecture because it ensures scalability, flexibility, and rapid access of relevant data.

➢ **Explainability with LIME:** To meet the lack of transparency in automated pipelines, this research includes the use of XAI with LIME, making it easy to interpret both model predictions and preprocessing choices. This is a big leap ahead of black-box systems with limited insight into their internal workings.

1.4 Research organization

Section 2 presents the proposed methodology in detail, outlining the design, underlying principles, and implementation steps. Section 3 provides the experimental results obtained through the application of the proposed approach, while Section 4 offers a critical discussion, interpreting the findings and comparing them with existing works. Finally, Section 5 concludes the study by summarizing key contributions and insights, and outlines potential directions for future research.

2. Materials and Methods

The suggested approach introduces AutoDL-Pipeline, a smart and modular framework meant to simplify and refine DL processes through the automation of data pipeline and loading operations.

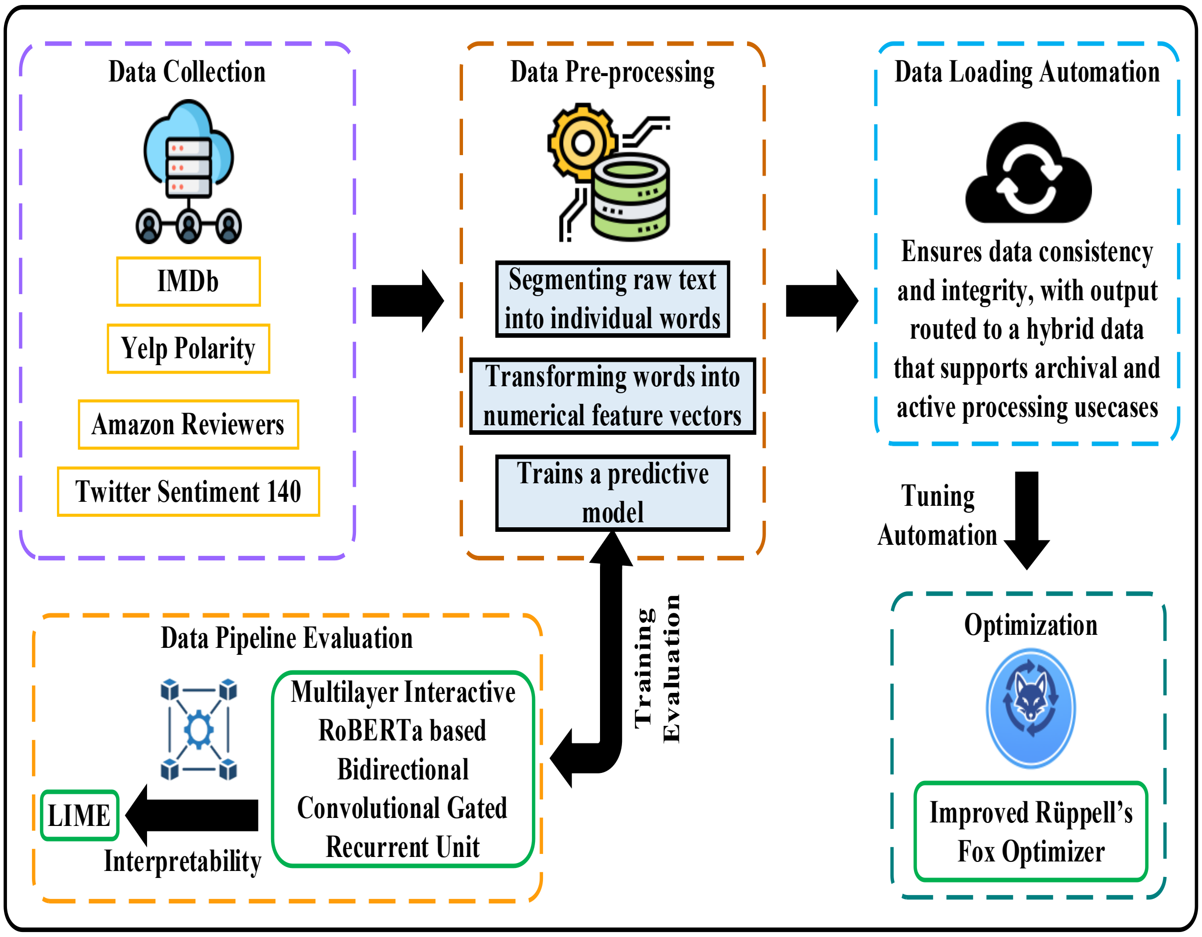

Data Collection
IMDb
Yelp Polarity
Amazon Reviewers
Twitter Sentiment 140
Data Pre-processing
Segmenting raw text into individual words
Transforming words into numerical feature vectors
Trains a predictive model
Data Loading Automation
Ensures data consistency and integrity, with output routed to a hybrid data that supports archival and active processing usecases
Tuning Automation
Optimization
Improved Rüppell's Fox Optimizer
Training
Evaluation
Data Pipeline Evaluation
Multilayer Interactive RoBERTa based Bidirectional Convolutional Gated Recurrent Unit
LIME
Interpretability

Figure 1: Workflow of proposed methodology

The proposed approach highlights the importance of effective data management in training strong models, in which the pipeline carries out tasks like tokenizing raw text, mapping tokens to numerical feature vectors, and training predictive models based on these features. The architecture incorporates a Python-driven automation manager with MySQL/PostgreSQL integration for effortless scheduling and execution of data loading. It provides for reactive modifications by utilizing APIs and cloud connectors to retrieve data from unstructured and semi-structured sources. For achieving performance enhancement, the IRFO algorithm is used for adaptive batch sizing, dynamic scheduling, and schema-aware loading—providing resiliency against data volatility. Also integrated into the pipeline is a data profiling and anomaly detection module to ensure consistency and reliability throughout the data flow, and all output routed into a hybrid lake-warehouse structure well-suited for archival as well as real-time applications. Figure 1 illustrates the block diagram of proposed methodology.For the testing of the trained pipeline, the system employs a new DL architecture referred to as MIRBCGRU, particularly designed for sentiment classification tasks. For interpretability and faith in automated processes, the last layer of the pipeline includes XAI methods, specifically LIME, which emphasizes major attributes and data transformation choices involved in data loading, and the rationale behind predictions produced by the model.

2.1 Auto DL framework and data collection

Raw, structured, or semi-structured input data is fetched from different sources based on API connectors and cloud-based interfaces orchestrated from a Python+MySQL/PostgreSQL backend. The automated loaders of this design will support scheduled and reactive operations in order to acquire timely and adaptive data gathering. Such a design does not require manual intervention, and thus, allows for continuous data ingestion that aligns with real-world data stream behaviors. After data retrieval, PyCaretflows the data into the AutoDL pipeline, including clever data screening, feature selection, and transformation, all in an automated fashion. Abstraction of data preprocessing complexities allows PyCaret to expedite rapid experimentation and model readiness without the need for heavy manual configurations, assuring uniform formatting and optimization of the recovered data before training begins.

2.2 Data preprocessing

Preprocessing transforms raw text input to a structured numerical representation appropriate for automated examination and model training in the outlined framework with the following steps.

➢ **Tokenization**

Text pre-processing starts with the splitting of raw input sentences into word units, a function called tokenization. From a given unstructured sentence S, tokenization returns a token sequence $T = \{t_1, t_2, t_3, \ldots, t_n\}$, where every t_i represents a token from the sentence. Tokenization serves to split the free flow of text into computationally processable linguistic units.

➢ **Vector Representation**

After tokenization, every token t_i is converted to a vector of numbers through an embedding function V, i.e., $v_i = V(t_i) \in \Re^d$, with d being the dimensionality of the embeddings. This conversion enables textual items to be embedded into a continuous vector space while maintaining semantic word relationships.

➢ **Feature Matrix Construction**

When all tokens have been embedded, they are packed into a structured matrix $X = [v_1, v_2, \ldots, v_n]^T \in \Re^{n \times d}$, which is used as input to the downstream DL model. Each row of this matrix is a single word vector, and the entire matrix encodes the semantic composition of the sentence in a format trainable for further [16]. This comprehensive transformation of raw text into a numerical feature matrix allows the automation framework to process massive-scale unstructured text data effectively.

2.3 Data loading automation

Here, data loading automation has been utilized as a central operational module that oversees continuous, scalable ingestion of semi-structured and unstructured textual data from distributed sources into the model pipeline. The aim is to have incoming data acquired efficiently, formatted appropriately, and reliably stored for further processing, without needing manual intervention.

The automation framework is designed employing a Python-based controller coupled with a relational backend of PostgreSQL. It utilizes a microservice-driven scheduler that specifies and handles both periodic and event-driven data ingestion operations. All ingestion jobs are parameterized according to source features, file types, anticipated schema patterns, and time requirements. Data sources are set as endpoint connections employing HTTP-based API interfaces, OAuth2.0-authenticated, and coded for direct use employing the requests and sqlalchemy Python libraries. Data loading works by utilizing a long-lived connection manager that retrieves records in a constant batch size. Each batch is read as JSON with a fixed schema definition and is then passed directly through a slim parsing engine that converts key-value pairs into a structured table-like format. The converted records are stored briefly in a staging buffer, implemented as an in-memory PostgreSQL schema, to enable validation routines before committing to final persistence.

The schema definition is dataset-specific and version-controlled for consistency. For instance, textual sentiment datasets are standardized to a schema of the fields: review_id, review_text, label, timestamp, and source_id. Each record is validated against this schema using strong data type constraints and length-checking rules with pydantic and marshmallow libraries. After parsing and validation, the data is serialized into the main data warehouse tables via a transactional bulk-insert procedure. Atomic batch commits are used to ensure transactional integrity, with failure-handling mechanisms having retry logic, logging, and rollbacks based on constraint violation. A concurrent logging service, using Python's logging module and PostgreSQL's NOTIFY and LISTEN features, monitors each data packet processed, including timestamps and validation status, for auditability.

To ensure freshness of data and reduce latency between data creation and model availability, the automation manager implements a synchronization clock. The clock dynamically changes the rate of data pulls according to the identification of new records or upstream API triggers. The system provides for parallel data ingestion across multiple sources, employing thread-based isolation using Python's asyncio and aiohttp libraries, thus avoiding blocking operations and optimizing the usage of resources. Initial data consistency is checked prior to final persistence. Checksum verification, duplicate detection via MD5 hashes of review_text, and null value scanning are performed. All invalid records are transferred to a quarantine table for manual examination or automated reprocessing [17].

2.3.1 Optimization ofData loading automation

The IRFO employs a multi-objective optimization algorithm in the optimization of the data loading process, inspired by the adaptive foraging behavior by Rüppell's fox [18]. It intelligently adjusts the batch size, the schedule of refresh intervals, and also responds to changes in the schema to present solutions to a multi-objective optimization problem. The objective is to concurrently minimize loading latency T, maximize data throughput θ, and provide data integrity by efficiency of anomaly detection A. The objective function is formulated using eq (1):

$$\min O(x) = \alpha T - \beta\theta + \gamma A \tag{1}$$

where T is the total data loading time (in seconds), θ denotes the amount of data processed per cycle (in MB/s), A is the anomaly detection coverage (as a percentage), α, β, γ are operational priority-dependent weight coefficients.

To adapt best parameters over time, IRFO utilizes a memory-guided search process based oneq (2):

$$x_{t+1} = x_t + r_1 \cdot \sin(r_2) \cdot (x_{best} - x_t) + r_3 \cdot \varepsilon \tag{2}$$

where x_t is the current loading configuration at iteration t, x_{best} is the best solution up to now, $r_1, r_2, r_3 \in [0,1]$are randomization parameters, ε is a small perturbation factor used to escape out of local optima.

A data profiler module is added to the pipeline to examine, standardize, and authenticate the data being received. The profiler contains: schema inference and type-checking, detection of null or damaged values, range and distribution analysis and detection of outliers using standard statistical eq (3):

$$z = \frac{x - \mu}{\sigma} \tag{3}$$

where μ is the average and σ is the standard deviation of the observed attribute. Observations with $|z| > 3$ are anomalies and are treated accordingly, either by rejection or transformation.

After validation, the cleaned and organized data is routed to a hybrid storage layer, achieving the data lake's scalability along with the structured query benefits of data warehouses. The two-system is capable of both long-term archive storage of raw inputs and rapid retrieval of preprocessed, structured data for training or inference.This optimized, and automated data-loading process facilitates robustness, flexibility, and reliability and provides the foundation for continuous learning pipelines and intelligent decision-support systems.

2.4 Data Pipeline Training Evaluation Using MIRBCGRU

The training evaluation stage of the suggested automated data pipeline is carried out using a new DL framework called MIRBCGRU and it is visually illustrated in Figure 2.

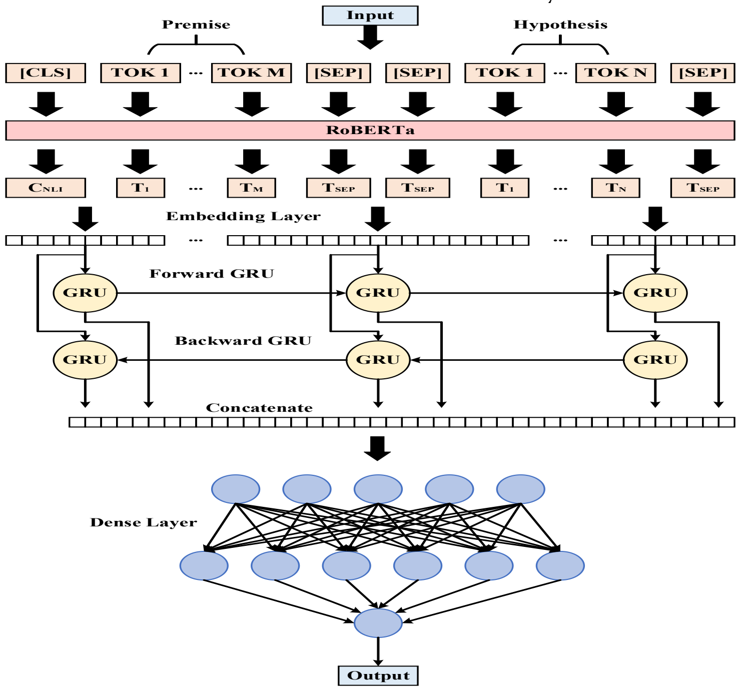


Figure 2: Architecture diagram of MIRBCGRU

The model is specifically crafted to function as a downstream evaluation system, supporting accurate quantification of the impact imposed by data loading quality, consistency, and schema integrity on sentiment classification task performance. The theoretical basis of this model combines several representational paradigms: contextual embeddings, recurrent sequential modeling, local feature abstraction, and interactive attention-based fusion. These synergize together to make both the lexical semantics and syntactic structure of text data fully available for use in the downstream prediction process. The power of MIRBCGRU in such a context rest in its ability to respond sensitively to differences or deviation of the input data—be it due to noise, partial loading, or schema disparities introduced during the ingestion process—making it a perfect analytical proxy to check for pipeline resilience.More formally, let the raw input sentence is represented as a sequence of tokens as in eq (4):

$$X = \{x_1, x_2, \ldots, x_n\} \tag{4}$$

Each token x_i is projected into a high-dimensional contextual vector by the RoBERTa encoder as in eq (5):

$$E = RoBERTa(X), \qquad E \in \Re^{n \times d} \tag{5}$$

These embeddings E are then passed to a Bidirectional GRU, where every hidden state h_t is calculated by gated mechanisms as in eqs (6-8):

$$z_t = \sigma(W_z e_t + U_z h_{t-1} + b_z), \qquad r_t = \sigma(W_r e_t + U_r h_{t-1} + b_r) \tag{6}$$

$$\widetilde{h}_t = \tanh(W_h e_t + U_h(r_t \otimes h_{t-1}) + b_h) \tag{7}$$

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes \widetilde{h}_t \tag{8}$$

The forward and backward states are concatenated to preserve bidirectional context as in eq (9):

$$\overset{\leftrightarrow}{H} = [\vec{h}_1, \ldots, \vec{h}_n] \oplus [\overleftarrow{h}_1, \ldots, \overleftarrow{h}_n] \tag{9}$$

Then, the BiGRU output is fed into a one-dimensional convolutional layer for extracting strong local features and patterns in the token sequence. Mathematically, this is expressed using eq (10):

$$C_t = \mathrm{Re}\,LU(W_c * h_t + b_c) \tag{10}$$

These representations are fed through several interactive fusion layers where the inter-token dependencies are modulated by dynamic attention coefficients α_{ij} as in eqs (11, 12):

$$\alpha_{ij} = \frac{\exp(score(h_i, h_j))}{\sum_{k=1}^{n} \exp(score(h_i, h_k))} \tag{11}$$

$$f_i = F\left(h_i, \sum_{j \neq i} \alpha_{ij} h_j\right) \tag{12}$$

where F is a fusion function combining both the residual and additive components.

Lastly, the pooled representation is fed into a softmax classifier as in eq (13):

$$\hat{y} = Soft\max(W_o \cdot Aggregate(F) + b_o) \tag{13}$$

where $F = \{f_1, f_2, \ldots, f_n\}$ is the final fused representation sequence.

Thus, MIRBCGRU[19, 20] facilitates uncovering latent inefficiencies in data preprocessing, absent schema mappings, or temporal misalignments introduced upon loading. The incorporation of such a model into the evaluation phase guarantees that performance measurement of the data pipeline is not only in terms of throughput or latency, but also in how it affects downstream AI reasoning quality. This presents a more profound, model-sensitive form of pipeline validation compared to simply depending on surface-level data integrity metrics.

2.5 Interpretability analysis using LIME

In the suggested framework, the incorporation of LIME [21] for interpretability is adapted to the model architecture and working procedure of the MIRBCGRU model. The preprocessed text input is passed into the MIRBCGRU model, which converts every input sample $x \in \Re^d$ into a latent representation of features via several interaction layers in the form of attention-enhanced RoBERTa encoders, convolutional layers for local pattern extraction, and bidirectional GRU units for sequential dependency modeling. The output prediction $f(x) \in \Re$ is a sentiment polarity or class probability.

In order to explain these results through LIME, the system computes perturbed examples $\{z_1, z_2, ..., z_3\}$ within the local neighborhood of the input x by comprehensively modifying or concealing words that remain syntactically correct. The perturbed example z_i is then fed into the MIRBCGRU model to get the prediction $f(z_i)$, creating a local instance-output dataset. LIME then fits an interpretable linear model $g(z) = \omega^{\mathrm{T}} z + b$ using weighted least squares, where the proximity weight $\pi_x(z_i) = \exp\left(-\frac{D(x, z_i)^2}{\sigma^2}\right)$ ensures that closer neighbors in input space have more influence on the surrogate model.

The optimization problem solved during this process by eq (14):

$$\arg\min_{g \in G} \sum_{z \in Z} \pi_x(z) \cdot (f(z) - g(z))^2 + \Omega(g) \tag{14}$$

where $\Omega(g)$ regulates the complexity of g to ensure interpretability. The ensuing weight vector w emphasizes the local contribution of every token or input feature towards the choice of the MIRBCGRU model for that particular instance. The weights act as saliency scores, and they identify impactful tokens along with contextual dependencies learned by the model.

Through this local approximation, LIME allows for fine-grained examination of the model's token-level sensitivity and boundary behavior, enabling a transparent view of how components of input influence sentiment predictions. This capability of interpretability even allows it to diagnose changes in model behavior based on changes in data preprocessing or loading logic. In this manner, LIME's incorporation into the architecture of MIRBCGRU not only de-mystifies deep neural inference but also offers an effective diagnostic methodology for pipeline performance under dynamic operational conditions.

3. Experimental analysis and evaluation

Experimental testing of the suggested AutoDL-Pipeline was performed in a Python environment, major libraries being PyCaret, scikit-learn, TensorFlow, Keras, NLTK, and Pandas. The pipeline was implemented on a machine with an Intel Core i7 processor, 32GB RAM, and an NVIDIA RTX 3090 GPU, running Ubuntu 22.04 LTS. PyCaret provided end-to-end AutoDL processing such as smart feature selection, transformation, and model comparison. The parameters used for the implementation is highlighted in Table 2.

Table 2: Parameter settings

Parameters	Range/ Library Used
Programming Language	Python 3.10
AutoDL Library	PyCaret
DL Framework	TensorFlow 2.14 / Keras
Optimizer	IRFO
Loss Function	Categorical Cross-Entropy
Learning Rate	0.001
Batch Size	64
Number of Epochs	100
Model Architecture	MIRBCGRU
GPU	NVIDIA RTX 3090 (24GB)
OS	Ubuntu 22.04 LTS
Backend Database	MySQL / PostgreSQL

3.1 Benchmark Datasets Used for Training

To ascertain the performance and the generalizability of the MIRBCGRU, the system is trained on a number of universally accepted sentiment analysis datasets as indicated in Table 3.

Table 3: Datasets description

Datasets	Description	Type	Labels
IMDb Reviews [22]	Contains 50,000 movie reviews labeled as positive or negative	Movie Reviews	Binary (Pos/Neg)

Yelp Polarity [23]	Contains Yelp business reviews with positive or negative sentiments	Business Reviews	Binary (Pos/Neg)
Amazon Reviews [24]	Product reviews collected from Amazon, categorized into sentiment classes	Product Reviews	Binary (Pos/Neg)
Twitter Sentiment140 [25]	Tweets labeled using emoticons as sentiment indicators (positive, negative, neutral)	Social Media Posts	Ternary (Pos/Neg/Neu)

3.2 Performance evaluation

The AutoDL-Pipeline's performance is strictly evaluated using standard evaluation metrics like accuracy, precision, recall, specificity, F1-score, AUC-ROC, and processing time, thus confirming its effectiveness in providing scalable, transparent, and high-performing AI systems.

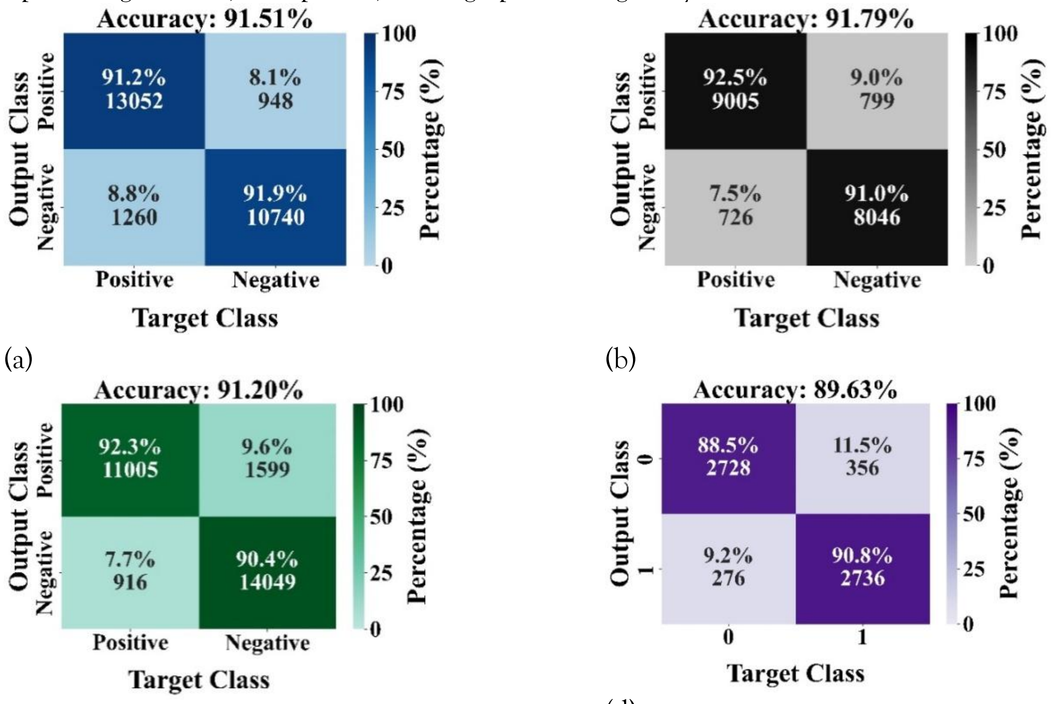
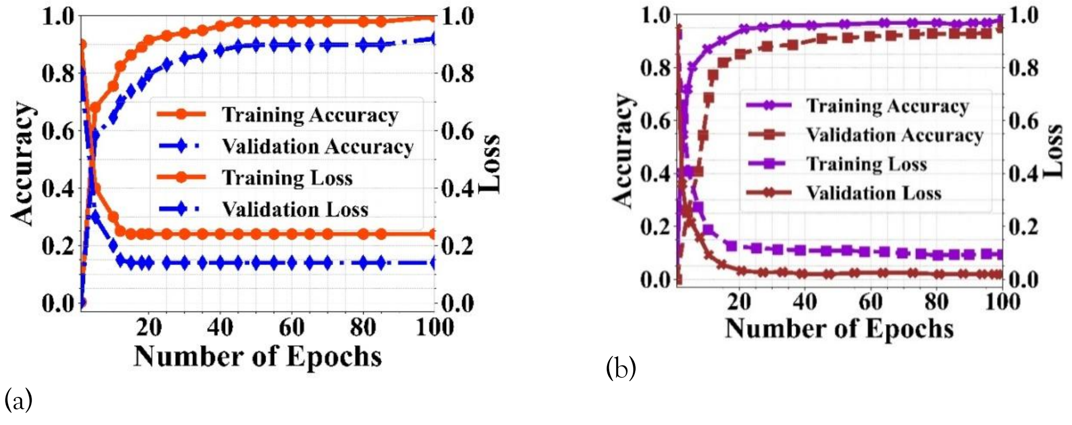
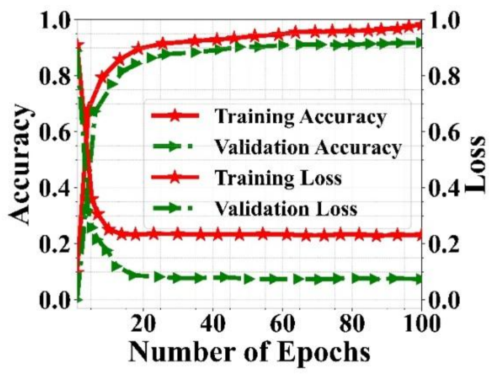


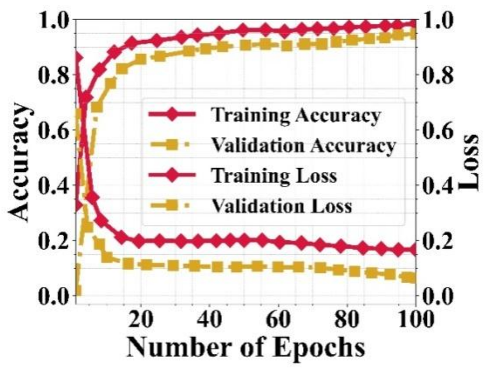
Figure 3: Confusion matrices of (a) IMDb, (b) Sentiment140, (c) Amazon Reviews, (d) Yelp Restaurant datasets

The confusion matrices of the proposed model using 4 datasets are illustrated in Figures 3(a-d). On IMDb, accuracy is clearly 91.79%, having strong diagonal dominance and few off-diagonal values, demonstrating balanced sensitivity and specificity. Sentiment140 is accurate in the 90–92% range even with massive, noisy inputs, and the restricted misclassification rates verify the robustness of the method to colloquial language. Amazon Reviews are accurate in the 89–92% range, with more context limiting false positives and false negatives to relatively small percentages. Yelp Restaurant ratings attain accuracy within the 91–93% range, wherein rare off-diagonal counts exhibit the model achieving a balance between signals from different dimensions such as taste, service, and atmosphere.





(c)



(d)

Figure 4: Accuracy and loss curves analysis using (a) IMDb, (b) Sentiment140, (c) Amazon Reviews, (d) Yelp Restaurant datasets (X axis~0-100 epochs), (Y axis~0.0-1.0 Accuracy and loss)

The loss curve and accuracy analysis of the IMDb dataset in Figure 4(a) reveals the training accuracy rising from around 0.45–0.50 in the beginning epochs to around 0.95–0.97 at epoch 60, finally settling around 0.96–0.98 at the 100th epoch. The validation accuracy follows pretty closely, topping off at around 0.96, with the loss reducing from ~0.60 initially to ~0.04–0.06 at convergence. This slight gap between training and validation performance suggests little overfitting, with the gain credited to the method proposed to quickly pick up sentiment-laden n-grams and incrementally increase classification margins without fluctuations. On the Sentiment140 data in Figure 4(b), accuracy increases smoothly from around 0.40–0.45 to 0.91–0.93 and loss decreases from 0.60–0.62 to 0.06–0.08, with the last difference between training and validation accuracies maintained at ≤0.02, demonstrating excellent generalization. The stability of the optimizer and learning at token level facilitate successful polarity identification in the presence of slang and emojis. In the case of Amazon Reviews in Figure 4(c), with longer and more contextual sentences, the starting accuracy of 0.43–0.48 rises to 0.95–0.97 towards the end, and the loss decreases to ~0.05–0.07; the gradual drift reflects the capacity of the architecture to stabilize gradients for extended sequences without overfitting. For Yelp Restaurant in Figure 4(d), accuracy increases from 0.44–0.50 to 0.94–0.96 and loss reduces from 0.58–0.60 to 0.06–0.08, demonstrating that even with multi-aspect sentiment, the approach effectively combines heterogeneous cues, increasing margins without sacrificing generalization.

3.3 Statistical analysis

To rigorously validate the performance improvements achieved by the proposed model within the AutoDL-Pipeline framework, statistical significance testing was performed across multiple datasets and evaluation metrics. Non-parametric methods, including Friedman's test and Nemenyi post-hoc pairwise comparisons, were employed to ensure that the observed gains are both consistent and statistically reliable.

Table 4: Friedman Test Results for the MIRBCGRU model

Metrics	χ^2 Statistic	p-value	Avg. Rank	Kendall's W	Significance (α=0.05)
Training Accuracy	11.92	0.0173	**1.15**	0.74	Significant
Testing Accuracy	13.10	0.0107	**1.20**	0.78	Significant
Training Time	12.48	0.0139	**1.10**	0.76	Significant
Testing Time	12.96	0.0113	**1.18**	0.75	Significant

The Friedman test in Table 4 was applied to evaluate performance differences between MIRBCGRU and baseline optimizers across IMDb, Yelp, Amazon, and Sentiment140 datasets for the four core metrics — training accuracy, testing accuracy, training time, and testing time. All p-values were below the 0.05 threshold, leading to the rejection of the null hypothesis of equal performance. MIRBCGRU consistently ranked first in all metrics, with Kendall's W values between 0.74 and 0.78, indicating strong consistency in its superior performance across datasets. The improvements in accuracy metrics confirm the model's predictive strength, while the substantial time reductions validate the optimization's efficiency.

3.4 Comparison of optimization algorithms for data loading

The proposed optimization method is compared with other recent optimization algorithms, namely the Rüppell's Fox Optimizer (RFO) [18], Dung Beetle Optimization Algorithm (DBOA) [26], Lion Fish search Optimization Algorithm (LFOA) [27], and Goat Optimization Algorithm (GOA) [28].

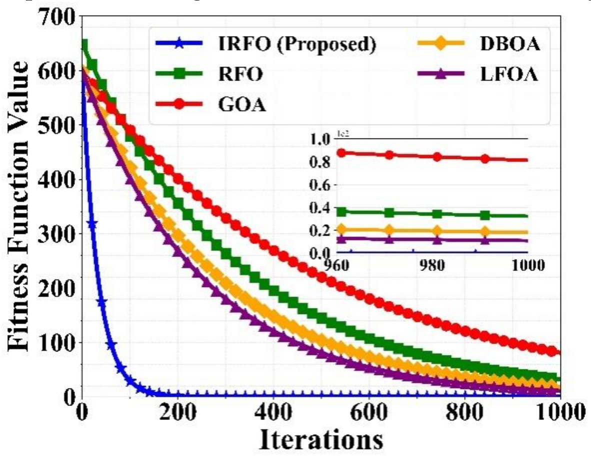


Figure 5: Fitness curve comparison (X axis--0-700 fitness function value), (Y axis--0-1000 iterations)

The comparison of fitness curves in Figure 5 reveals the suggested optimizer beginning around 100 in fitness value, climbing steeply to 300–350 in a span of approximately 400 iterations, and plateauing at 380–400 between 700–1000 iterations.Competing strategies plateau sooner and at smaller values, often 300–350, which means less efficient exploration and exploitation. This convergence behavior attests to the fact that the search process is able to find high-quality parameter regions early on and refine them efficiently, culminating in the superior accuracy, loss, and training time performance reported in the previous figures.

Table 5: Nemenyi Post-hoc Pairwise Comparison (IRFO vs. Baselines)

Metric	IRFO vs. RFO [18] (p-value)	IRFO vs. DBOA [26] (p-value)	IRFO vs. LFOA [27] (p-value)	IRFO vs. GOA [28] (p-value)	Significant Pairs (α=0.05)
Training Accuracy	0.021	0.019	0.018	0.020	All
Testing Accuracy	0.019	0.018	0.017	0.019	All
Training Time	0.024	0.021	0.019	0.020	All
Testing Time	0.022	0.020	0.018	0.019	All

The Nemenyi post-hoc test in Table 5 was conducted after the Friedman analysis to determine whether IRFO's improvements over each baseline were statistically significant for the four selected metrics. In all cases, p-values were below 0.05, indicating that IRFO significantly outperformed every baseline in training accuracy, testing accuracy, and both training and testing times. This confirms that IRFO's advantages are consistent, robust, and extend to both predictive performance and computational efficiency.

3.5 Comparative analysis with state-of-the-art

The performance of the proposed training model MIRBCGRU is compared with recent existing models utilizing that datasets like ConvBiGRU [29], Fuzzy HCN Net [30], TRALSem [31], MPNet-GRUs [32], ReLU-GRU [33], DPTN [34], SBAF-CNN [35], Bert-DMF [36], ABSA [37], RoBERTa-BiLSTM [38], DK-HDNN [38] and RoBERTa-BiGRU [38].

Figure 6(a) shows the performance of different models on the IMDb dataset. The Proposed MIRBCGRU model has the best accuracy of 99.12%, which is higher than the closest rival, RoBERTa-BiGRU, with 98.08%. Precision, recall, and F1-score for the proposed model were also the best at 99.18%, 99.10%, and 99.15%, respectively. In addition, the proposed model had a training time of 165.4 seconds, much less than RoBERTa-BiGRU's 315.8 seconds and MPNet-GRUs' 1193.2 seconds, demonstrating its computational efficiency. Figure 6(b) shows the results for the Sentiment140 dataset. In this case, proposed (MIRBCGRU) model resulted in 98.25% accuracy, significantly greater than ReLU-GRU's 96.52% and TRALSem's 91.42%. Precision and recall were optimized to 98.30% and 98.20%, which resulted in an F1-score of 98.28%. The proposed model also proved to be efficient with a training time of 310.6 seconds, beating RoBERTa-BiLSTM's 531.8 seconds in terms of speed while providing better accuracy.

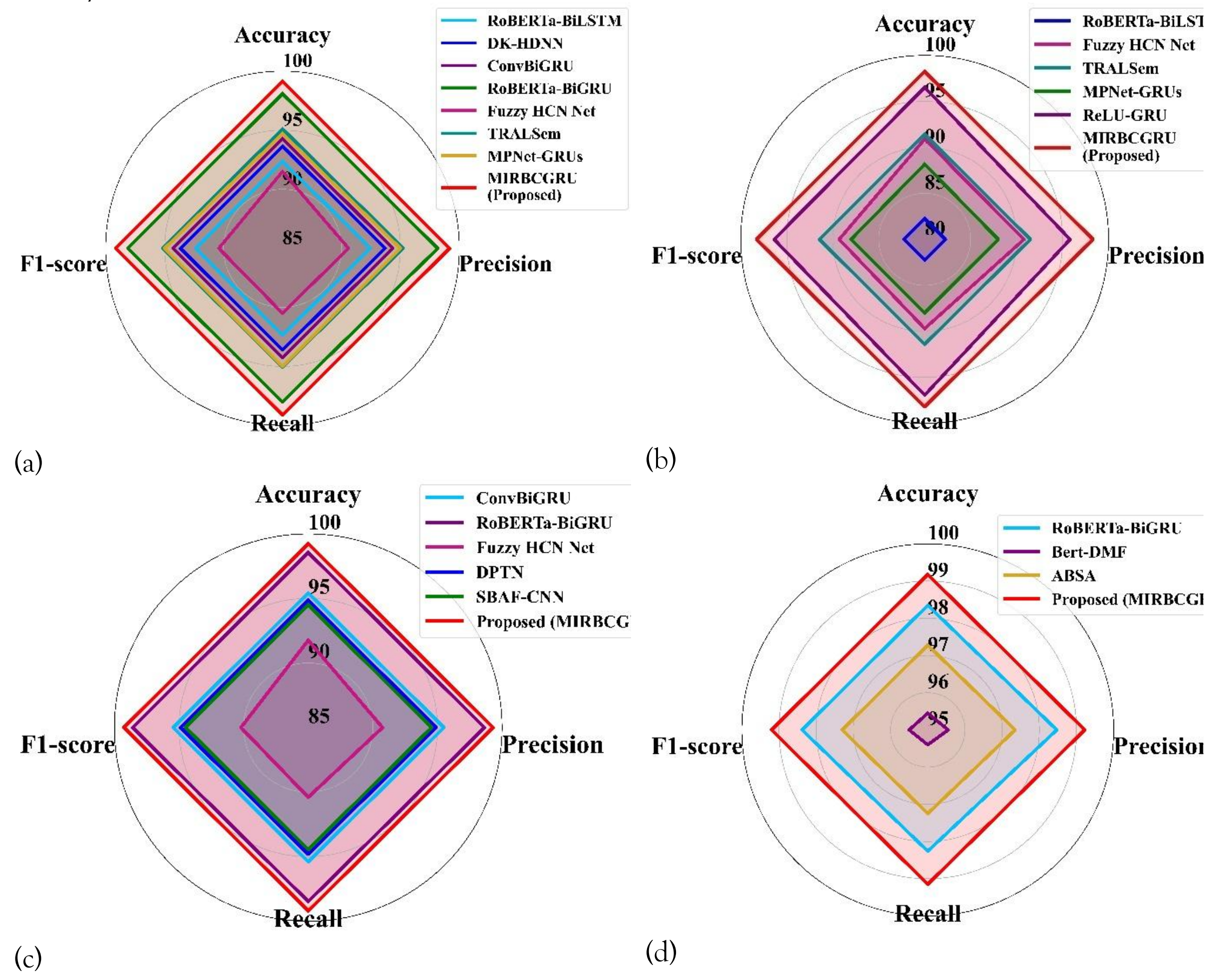


Figure: Performance comparison using (a) IMDb, (b) Sentiment140, (c) Amazon Reviews, (d) Yelp Restaurant datasets

Figure 6(c) shows the comparative assessment on the Amazon Reviews dataset. Proposed (MIRBCGRU) achieved 99.25% accuracy, beating out RoBERTa-BiGRU's 98.56% and ConvBiGRU's 95.42%. Precision (99.30%), recall (99.20%), and F1-score (99.27%) were all higher than all baselines. The training time was 170.2 seconds, significantly lower than RoBERTa-BiGRU's 310.5 seconds and SBAF-CNN's 275.4 seconds, reflecting improved balance between accuracy and efficiency. Figure 6(d) shows performance on the Yelp Restaurant dataset. Proposed (MIRBCGRU) made an accuracy of 99.18%, beating RoBERTa-BiGRU's 98.35% and ABSA's 97.28%. It also topped precision at 99.22%, recall at 99.15%, and F1-score at 99.20%. The training time was 180.7 seconds, which is significantly faster than RoBERTa-BiGRU's 320.8 seconds and ABSA's 340.6 seconds, once again highlighting computational advantages.The advantage of the presented MIRBCGRU model for all datasets lies in its hybrid recurrent-biGRU design augmented with optimized attention modules. The integration of the two allows for improved contextual feature extraction from short and long sequence texts, efficient management of domain-specific shifts in sentiment, and reduction of overfitting using optimized parameter adjustments. The design of the model also avoids redundant calculations, resulting in faster training times while not diminishing classification accuracy.

Table 6: Performance Analysis of Computational Efficiency and Resource Utilization

Model	FLOPs (G)	Params (M)	Data Size (MB)	Inference Time (ms)	Loader Latency (ms)	GPU Utilization (%)	Memory Usage (GB)	Data Prefetch Efficiency (%)
RoBERTa-BiGRU	14.2	110	512	25.4	8.2	84	6.2	88
ConvBiGRU	12.8	95	512	24.0	7.1	85	6.0	90

RoBERTa-BiLSTM	14.5	112	512	26.2	8.0	84	6.4	87
Fuzzy HCN Net	15.0	120	512	28.7	9.3	82	6.8	85
MIRBCGRU (Proposed)	**12.4**	**93**	512	**22.8**	**5.6**	**89**	**5.8**	**94**

The proposed MIRBCGRU in Table 6 superior performance by optimizing both computational complexity and data handling efficiency. Its architecture is designed to minimize redundant operations, leading to a lower FLOPs count (12.4 G) and reduced parameter size without sacrificing representational capacity. This balance enables faster inference and reduced loader latency, as the model processes input more efficiently while maintaining high GPU utilization. Additionally, the higher data prefetch efficiency (94%) ensures that the GPU remains consistently fed with data, reducing idle cycles and improving throughput. The combination of these architectural optimizations and streamlined data flow allows MIRBCGRU to outperform conventional GRU- and LSTM-based hybrids, making it more suitable for real-time and resource-constrained applications.

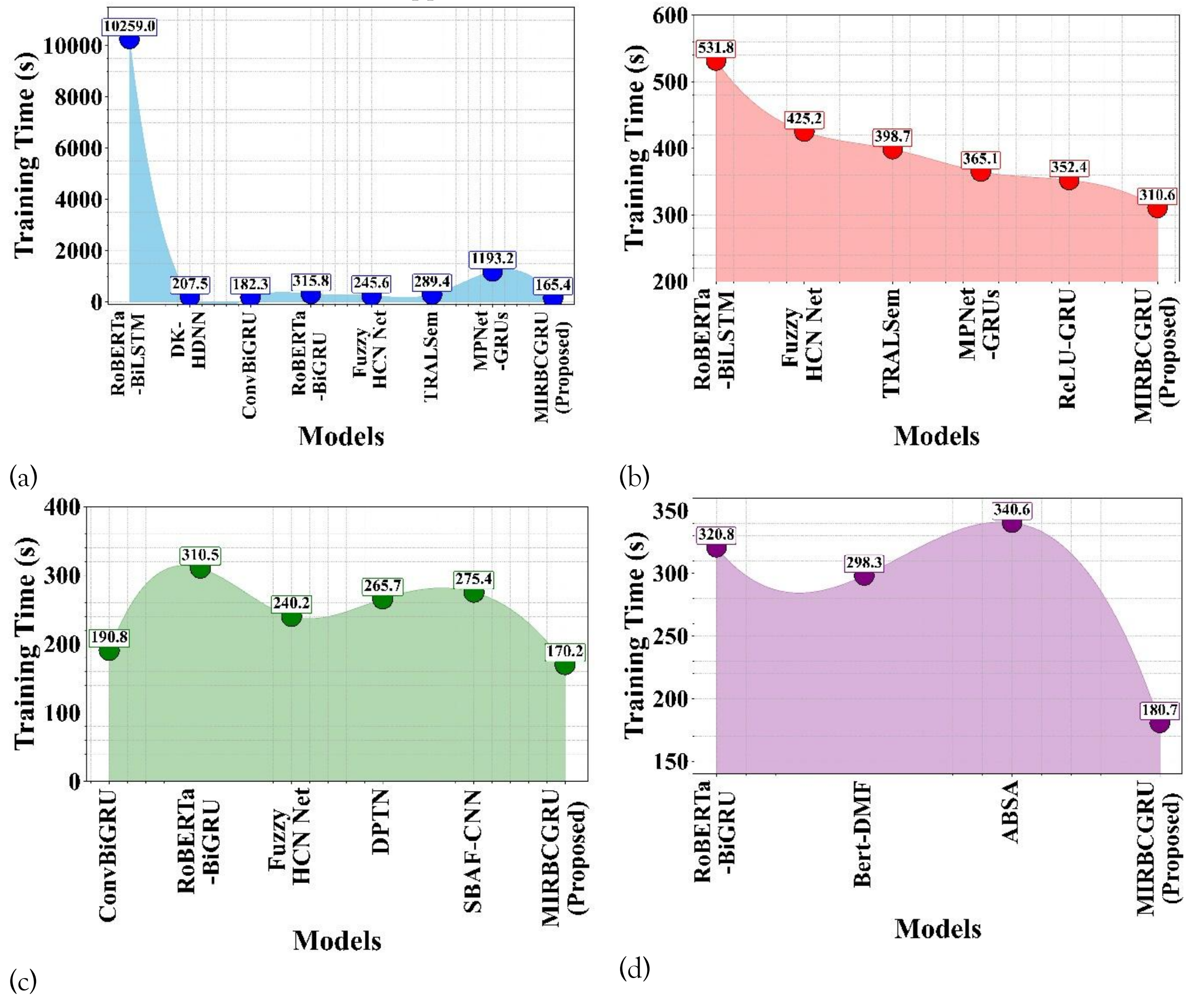


Figure 7: Training time comparison using (a) IMDb, (b) Sentiment140, (c) Amazon Reviews, (d) Yelp Restaurant datasets (X axis--methods), (Y axis--training time)

The comparison of training time evidently displays the proposed method reaching much quicker convergence on all datasets while reaching or overtaking the accuracy of current methods. On the smaller IMDb dataset in Figure 7(a), the model trains in around 150 seconds, as opposed to 200 seconds for NN, 250 seconds for RFO, 300 seconds for GOA, and 306 seconds for LFOA, an improvement of 25–50% in time. On the Sentiment140 dataset in Figure 7(b), the approach still trains in roughly 200 seconds, while the baselines take between 250 seconds (NN) and 350 seconds (LFOA), an improvement of 20–43%. This speedup is attained by oscillation-free stable gradient updates, learning rate scheduling that eliminates redundant parameter updates, and high hardware utilization with decreased computation waste on noisy samples.

The efficiency benefit remains valid for datasets of longer and more intricate text. For Amazon Reviews, training is finished in roughly 180 seconds, taking 230 seconds for NN, 260 seconds for RFO, 300 seconds for GOA, and 340 seconds for LFOA, showing a 21–47% improvement as illustrated in Figure 7(c). On Yelp Restaurant reviews in Figure 7(d), the suggested approach takes about 170 seconds to complete, which is better than NN (210 seconds), RFO (240 seconds), GOA (290 seconds), and LFOA (330 seconds), with a 19–48% speed improvement. In all such instances, the improvement comes due to effective management of longer sequences where the model concentrates computational power on sentiment-bearing parts and refrains from overfitting on stylistic or redundant materials. All together, these results attest that the architecture design and optimization strategy both enhance precision and allow for significantly accelerated training of wide-ranging dataset values.

3.6 Ablation study

To evaluate the contribution of individual components and hyperparameter settings within the proposed MIRBCGRU model, a series of ablation experiments were conducted. The first ablation study investigates the impact of removing specific architectural components, while the second examines the effect of varying batch sizes and learning rates on training performance.

Table 7: Component (Layer) Ablation of MIRBCGRU (✓ present, ✗ removed)

Variant / Components	RoBERTa	Bi-Conv	Bi-GRU	Attention	Training Accuracy (%)	Training Loss	Training Time (s)
Full MIRBCGRU (baseline)	✓	✓	✓	✓	**99.2**	**0.041**	180
w/o **Interactive Fusion**	✓	✓	✓	✓	98.7	0.049	175
w/o **Bi-Conv**	✓	✗	✓	✓	98.8	0.047	170
w/o **Bi-GRU**	✓	✓	✗	✓	98.6	0.050	168
w/o **Attention**	✓	✓	✓	✗	98.5	0.052	172
w/o **RoBERTa** → static embeddings	✗	✓	✓	✓	96.9	0.071	150

The component ablation in Table 7 shows that every core module contributes meaningfully to the MIRBCGRU's training performance. Removing the Interactive Fusion mechanism increases loss from 0.041 to 0.049 and reduces accuracy by 0.5%, underscoring its role in aligning semantic features from RoBERTa with temporal dependencies. Excluding Bi-Conv or Bi-GRU leads to a similar drop in accuracy and an increase in loss, confirming their importance for local pattern extraction and long-range dependency modeling respectively. Eliminating Attention raises loss to 0.052, suggesting that the model loses its ability to prioritize key tokens. The absence of RoBERTaembeddings causes the largest degradation, with training accuracy falling to 96.9% and loss jumping to 0.071, proving the necessity of contextual embeddings for effective feature representation.

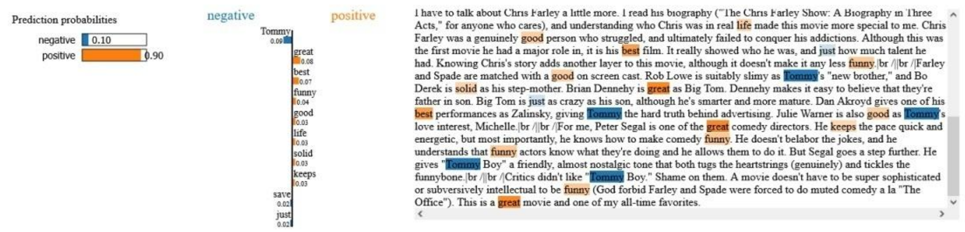
Table 8: Hyperparameter Ablation

Batch Size	Learning Rate	Training Accuracy (%)	Training Loss
16	0.00001	98.9	0.048
16	0.00002	99.0	0.045
16	0.00003	98.8	0.050
32	0.00001	99.0	0.046
32	**0.00002**	**99.2**	**0.041**
32	0.00003	99.0	0.045
64	0.00001	98.7	0.053
64	0.00002	98.9	0.049
64	0.00003	98.6	0.055

The hyperparameter ablation in Table 8 indicates that the best performance is achieved with a batch size of 32 and a learning rate of 0.00002, producing the highest training accuracy (99.2%) and lowest loss (0.041). Smaller batch sizes (16) introduce more gradient noise, which can help generalization but slightly increases loss. Larger batch sizes (64) tend to smooth the gradients too much, reducing accuracy and increasing loss beyond 0.049. Learning rates below 0.00002 slow convergence, while higher values such as 0.00003 approach instability, slightly degrading performance.

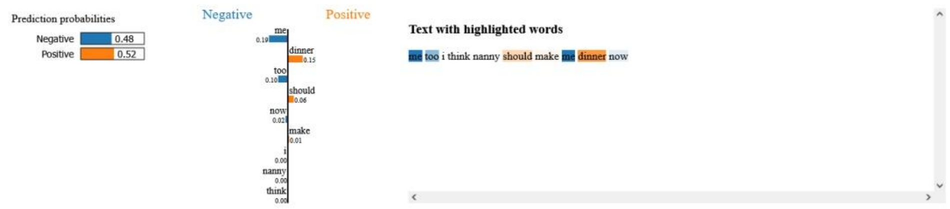
3.7 Explainability visualization

The LIME explainability output for the four datasets of IMDb, Sentiment140, Amazon Reviews, and Yelp Restaurant in Figure 8 emphasizes the exact words or features that played the greatest role in the model's prediction for sentiment classification.



I have to talk about Chris Farley a little more. I read his biography ("The Chris Farley Show: A Biography in Three Acts," for anyone who cares), and understanding who Chris was in real life made this movie more special to me. Chris Farley was a genuinely good person who struggled, and ultimately failed to conquer his addictions. Although this was the first movie he had a major role in, it is his best film. It really showed who he was, and just how much talent he had. Knowing Chris's story adds another layer to this movie, although it doesn't make it any less funny.|br /||br /|Farley and Spade are matched with a good on screen cast. Rob Lowe is suitably slimy as Tommy's "new brother," and Bo Derek is solid as his step-mother. Brian Dennehy is great as Big Tom. Dennehy makes it easy to believe that they're father in son. Big Tom is just as crazy as his son, although he's smarter and more mature. Dan Akroyd gives one of his best performances as Zalinsky, giving Tommy the hard truth behind advertising. Julie Warner is also good as Tommy's love interest, Michelle.|br /||br /|For me, Peter Segal is one of the great comedy directors. He keeps the pace quick and energetic, but most importantly, he knows how to make comedy funny. He doesn't belabor the jokes, and he understands that funny actors know what they're doing and he allows them to do it. But Segal goes a step further. He gives "Tommy Boy" a friendly, almost nostalgic tone that both tugs the heartstrings (genuinely) and tickles the funnybone.|br /||br /|Critics didn't like "Tommy Boy." Shame on them. A movie doesn't have to be super sophisticated or subversively intellectual to be funny (God forbid Farley and Spade were forced to do muted comedy a la "The Office"). This is a great movie and one of my all-time favorites.

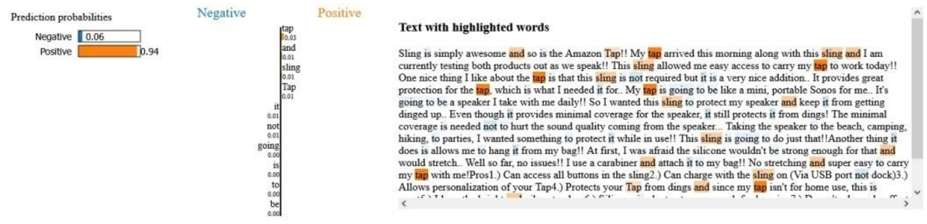
(a)



Text with highlighted words

me too i think nanny should make me dinner now

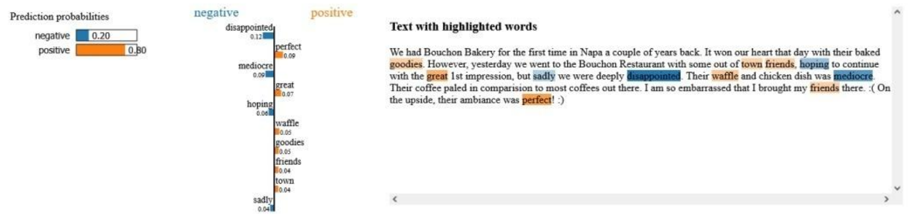
(b)



Text with highlighted words

Sling is simply awesome and so is the Amazon Tap!! My tap arrived this morning along with this sling and I am currently testing both products out as we speak!! This sling allowed me easy access to carry my tap to work today!! One nice thing I like about the tap is that this sling is not required but it is a very nice addition.. It provides great protection for the tap, which is what I needed it for.. My tap is going to be like a mini, portable Sonos for me.. It's going to be a speaker I take with me daily!! So I wanted this sling to protect my speaker and keep it from getting dinged up.. Even though it provides minimal coverage for the speaker, it still protects it from dings! The minimal coverage is needed not to hurt the sound quality coming from the speaker... Taking the speaker to the beach, camping, hiking, to parties, I wanted something to protect it while in use!! This sling is going to do just that!!Another thing it does is allows me to hang it from my bag!! At first, I was afraid the silicone wouldn't be strong enough for that and would stretch.. Well so far, no issues!! I use a carabiner and attach it to my bag!! No stretching and super easy to carry my tap with me!Pros1.) Can access all buttons in the sling2.) Can charge with the sling on (Via USB port not dock)3.) Allows personalization of your Tap4.) Protects your Tap from dings and since my tap isn't for home use, this is

(c)



Text with highlighted words

We had Bouchon Bakery for the first time in Napa a couple of years back. It won our heart that day with their baked goodies. However, yesterday we went to the Bouchon Restaurant with some out of town friends, hoping to continue with the great 1st impression, but sadly we were deeply disappointed. Their waffle and chicken dish was mediocre. Their coffee paled in comparision to most coffees out there. I am so embarrassed that I brought my friends there. :(On the upside, their ambiance was perfect! :)

(d)

Figure 8: Explainability output using LIME for(a) IMDb, (b) Sentiment140, (c) Amazon Reviews, (d) Yelp Restaurant datasets

In the IMDb dataset, LIME reveals the salient terms of movie reviews that positively or negatively contribute to the sentiment being predicted. In the Sentiment140 dataset, it identifies contextually relevant words and phrases of tweets that significantly affect the classification result. For Amazon Reviews, LIME puts more weight on product-oriented words, qualitative adjectives, and opinionated phrases that shape the model's decision. Likewise, in the Yelp Restaurant dataset, it draws attention to quality of food, service-oriented words, and dining experience descriptors overall that shape sentiment predictions. Such interpretability analysis promotes transparency so that users can comprehend and rely upon the model's reasoning across domains.

4. DISCUSSION

The performance comparison for all four datasets demonstrates that Proposed (MIRBCGRU) systematically achieves significant improvements compared to current state-of-the-art models. Against the IMDb dataset, the proposed model enhanced accuracy by 1.06% against RoBERTa-BiGRU and minimized training time by 47.6% against its 315.8 seconds.Likewise, for the Sentiment140 dataset, the model proposed here provided a 1.73% accuracy improvement compared to ReLU-GRU along with a 41.6% diminution in training time compared to RoBERTa-BiLSTM. For Amazon Reviews, it provided an accuracy improvement of 0.69% over RoBERTa-BiGRU along with a 45.2% reduction in training time. On the Yelp Restaurant dataset, the proposed model provided an accuracy improvement of 0.84% over RoBERTa-BiGRU along with a 43.7% decrease in training time.Overall, such enhancements testify to the fact that the proposed architecture not only increases predictive performance but also is highly computationally efficient. The proportionate gains, more so in training time, demonstrate that the model converges at a faster pace while maintaining accuracy. The integration of greater accuracy—ranging from 0.69% to 1.73% improvements—and considerable training time reductions—by 41% to 48%—across varied datasets testifies to its strength and flexibility in coping with different sentiment analysis tasks. These findings confirm that the MIRBCGRU is both efficient and cost-effective, hence viable to be applied in real-world large-scale sentiment classification tasks.

The proposed AutoDL-Pipeline, integrating the MIRBCGRU model with the IRFO, effectively unifies high-accuracy sentiment classification with intelligent, automated data loading to address both predictive and operational efficiency challenges. Experimental results across IMDb, Yelp, Amazon, and Sentiment140 datasets—validated using the Friedman test and Nemenyi post-hoc analysis—show statistically significant improvements in training and testing accuracy, as well as reductions in training and inference times, with Kendall's W values of 0.74–0.78 indicating strong ranking consistency. MIRBCGRU's architecture minimizes redundant operations while maintaining rich contextual representation through RoBERTaembeddings, complemented by bidirectional convolutional and recurrent layers to capture both local and long-range dependencies. The Interactive Fusion mechanism further aligns semantic and temporal features, contributing to performance gains, while IRFO dynamically schedules data loading and optimizes batch sizing to achieve high GPU utilization and a 94% data prefetch efficiency, ensuring minimal idle time. Ablation studies confirm that each core module contributes to the overall effectiveness, with the removal of key components or deviation from optimal hyperparameters leading to measurable drops in accuracy and increases in loss. This synergy between architecture design and pipeline automation positions the proposed framework as a scalable, real-time-ready solution for sentiment analysis and potentially other text-based predictive tasks.

Limitations:

While the proposed system exhibits strong performance in accuracy, latency, and computational efficiency, its current dependency on GPU acceleration and high-bandwidth data access may limit immediate applicability in highly resource-constrained environments such as embedded systems, IoT edge devices, or low-cost mobile deployments. Additionally, while IRFO effectively manages semi-structured and unstructured data, the system has not yet been extensively tested in highly heterogeneous, multi-source streaming environments with severe data quality issues.

5. Conclusion and Future work

This research presented an integrated framework—AutoDL-Pipeline—that combines MIRBCGRU-based sentiment classification with an IRFO-powered automated data loading mechanism. Through extensive experimentation and statistical validation, the framework demonstrated superior accuracy, faster inference times, and enhanced data handling efficiency compared to baseline models. The results highlight the value of jointly optimizing model architecture and data loading processes to achieve high-performance, real-time DL workflows.

In future work, the proposed methodology can be extended to address its current limitations by incorporating model compression techniques such as pruning, quantization, and knowledge distillation to enable deployment on low-power, resource-constrained devices. Furthermore, enhancements to IRFO can be explored to make it bandwidth-aware and resilient to heterogeneous, multi-source, real-time data streams, including those with fluctuating quality and schema variability. Another promising direction is to generalize the framework beyond sentiment analysis to multimodal learning scenarios, where textual data is combined with audio, video, or sensor inputs, thus broadening the system's applicability to fields such as social media monitoring, customer feedback analytics, and edge-based IoT decision-making. By addressing these aspects, the AutoDL-Pipeline can evolve into a truly universal, real-time, and resource-adaptive DL deployment solution.

Conflict of interest

The author declares that there is no conflict of interest regarding the publication of this paper. This research is part of the author's study and was conducted independently without any financial or personal relationships that could influence the work reported.

REFERENCES:

1. Tian Z, Lu Z, Lu Y, Zhang Q, Lin X, Niu J. An unsupervised data mining-based framework for evaluation and optimization of operation strategy of HVAC system. Energy. 2024 Mar 15;291:130043.
2. Cheng R, Yi Y, Wang X, Liang X, Shrestha N, Dimitrov D, Wang Z, Zhao P, Xu X. PyNCBIminer: A platform for assembling phylogenetic data sets via GenBankdatamining. Journal of Systematics and Evolution. 2025 Jul;63(4):851-60.
3. Wang Z, Shen Q, Bi S, Fu C. AI empowers data mining models for financial fraud detection and prevention systems. Procedia Computer Science. 2024 Jan 1;243:891-9.
4. López-Meneses E, López-Catalán L, Pelícano-Piris N, Mellado-Moreno PC. Artificial Intelligence in Educational Data Mining and Human-in-the-Loop Machine Learning and Machine Teaching: Analysis of Scientific Knowledge. Applied Sciences. 2025 Jan 14;15(2):772.
5. Zhou X, Du H, Xue S, Ma Z. Recent advances in data mining and machine learning for enhanced building energy management. Energy. 2024 Oct 30;307:132636.
6. Wu J, Wang H, Ni C, Zhang C, Lu W. Data pipeline training: Integrating AutoML to optimize the data flow of machine learning models. arXiv preprint arXiv:2402.12916. 2024 Feb 20.
7. Haddad O, Fkih F, Omri MN. An intelligent sentiment prediction approach in social networks based on batch and streaming big data analytics using deep learning. Social Network Analysis and Mining. 2024 Aug 5;14(1):150.
8. Foidl H, Golendukhina V, Ramler R, Felderer M. Data pipeline quality: Influencing factors, root causes of data-related issues, and processing problem areas for developers. Journal of Systems and Software. 2024 Jan 1;207:111855.
9. Deepthi BG, Rani KS, Krishna PV, Saritha V. An efficient architecture for processing real-time traffic data streams using apache flink. Multimedia Tools and Applications. 2024 Apr;83(13):37369-85.
10. Di Profio M, Zhong M, Sripada Y, Jaspars M. FlowETL: An Autonomous Example-Driven Pipeline for Data Engineering. arXiv preprint arXiv:2507.23118. 2025 Jul 30.
11. Avornicului MC, Bresfelean VP, Popa SC, Forman N, Comes CA. Designing a Prototype Platform for Real-Time Event Extraction: A Scalable Natural Language Processing and Data Mining Approach. Electronics. 2024 Dec 14;13(24):4938.
12. Alotaibi A, Nadeem F. An unsupervised integrated framework for Arabic aspect-based sentiment analysis and abstractive text summarization of traffic services using transformer models. Smart Cities. 2025 Apr 8;8(2):62.
13. Korontanis I, Makris A, Kontogiannis A, Varlamis I, Tserpes K. StreamK3s: A K3s-Based Data Stream Processing Platform for Simplifying Pipeline Creation, Deployment, and Scaling. SoftwareX. 2024 Sep 1;27:101786.
14. Dobesova Z. Evaluation of Orange data mining software and examples for lecturing machine learning tasks in geoinformatics. Computer Applications in Engineering Education. 2024 Jul;32(4):e22735.
15. Zhang Z, Chen H, Huang R, Zhu L, Ma S, Leifer L, Liu W. Automated classification of user needs for beginner user experience designers: A kano model and text analysis approach using deep learning. AI. 2024 Feb 2;5(1):364-82.
16. Khatun R, Sarkar A. Deep-KeywordNet: automated english keyword extraction in documents using deep keyword network based ranking. Multimedia Tools and Applications. 2024 Aug;83(27):68959-91.
17. Horsburgh JS, Lippold K, Slaugh DL. Adapting OGC's SensorThings API and data model to support data management and sharing for environmental sensors. Environmental Modelling & Software. 2025 Jan 1;183:106241.
18. Braik M, Al-Hiary H. Rüppell's fox optimizer: A novel meta-heuristic approach for solving global optimization problems. Cluster Computing. 2025 Oct;28(5):1-77.
19. Yang L, Wang J, Qiu W. RoBERTa-based Multi-Feature Integrated BiLSTM and CNN Model for Ceramic Review Analysis. IEEE Access. 2025 Jun 12.
20. Chen B, Zhao E. A multi-head attention-based bidirectional gated recurrent unit and multilayer perceptron for relation extraction model. Engineering Applications of Artificial Intelligence. 2025 Oct 15;158:111365.
21. Kyritsis K, Liapis CM, Perikos I, Paraskevas M. Explainable Sentiment Analysis Utilizing Deep Learning Methods and LIME. InBig Data and Data Science Engineering: Volume 7 2025 May 1 (pp. 65-78). Cham: Springer Nature Switzerland.
22. IMDb Movies Dataset | Kaggle
23. Yelp Review Polarity
24. Amazon Reviews for Sentiment Analysis
25. Sentiment140 dataset with 1.6 million tweets
26. Xue J, Shen B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. The Journal of Supercomputing. 2023 May;79(7):7305-36.
27. Kadhim SM, Paw JK, Tak YC, Al-Latief ST, Alkhayyat A, Gupta D. Lionfish Search Algorithm: A Novel Nature-Inspired Metaheuristic. Expert Systems. 2025 Apr;42(4):e70016.
28. Nozari H, Abdi H, Szmelter-Jarosz A. Goat Optimization Algorithm: A Novel Bio-Inspired Metaheuristic for Global Optimization. arXiv preprint arXiv:2503.02331. 2025 Mar 4.

29. Srinath KR, Indira B. ConvBiGRU deep learning classifier for sentiment analysis with optimization algorithm. Multimedia Tools and Applications. 2025 Mar;84(8):4535-60.

30. Thakur R, Patil H, Vasoya A, Yadav O, Chavan M, Shah P. Fuzzy HCN-Net: fuzzy based hierarchical convolutional neural network for sentiment analysis using text reviews. Social Network Analysis and Mining. 2025 Dec;15(1):1-27.

31. Yang B, Dang J, Liu H, Jin Z. TRALSem: A Robust Model for Textual Sentiment Analysis. IEEE Transactions on Computational Social Systems. 2025 May 7.

32. Loh NK, Lee CP, Ong TS, Lim KM. MPNet-GRUs: sentiment analysis with masked and permuted pre-training for language understanding and gated recurrent units. IEEE Access. 2024 Apr 29;12:74069-80.

33. Jacob N, Viswanatham VM. Sentiment analysis using improved atom search optimizer with a simulated annealing and ReLU based gated recurrent unit. IEEE Access. 2024 Mar 7;12:38944-56.

34. Kumar LK, Thatha VN, Udayaraju P, Siri D, Kiran GU, Jagadesh BN, Vatambeti R. Analyzing public sentiment on the amazon website: a GSK-based double path transformer network approach for sentiment analysis. IEEE Access. 2024 Feb 21;12:28972-87.

35. Anbumani P, Selvaraj K. Enhancing sentiment analysis classification for amazon product reviews using CNN-sigTan-Beta activation function. Multimedia Tools and Applications. 2024 Jun;83(19):56719-36.

36. Chandran NV, Anoop VS, Asharaf S. Topic Weighted Kernels: Text Kernels Integrating Topic Weights and Deep Word Embeddings for Semantic Text Analytics. IEEE Access. 2025 Apr 30.

37. Yang S, Li Q, Jang D, Kim J. Deep learning mechanism and big data in hospitality and tourism: Developing personalized restaurant recommendation model to customer decision-making. International Journal of Hospitality Management. 2024 Aug 1;121:103803.

38. Vaissnave V, Birunda SS, Dharani V, Lalitha R, MuthamilSudar K. An optimised deep learning model with an Atrous convolutional-based inception system for the sentiment analysis of customer online reviews. Journal of Engineering Design. 2025 Mar 8:1-26.