



# IJRASET

International Journal For Research in  
Applied Science and Engineering Technology



---

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

---

**Volume:** 14    **Issue:** IV    **Month of publication:** April 2026

**DOI:** <https://doi.org/10.22214/ijraset.2026.79539>

[www.ijraset.com](http://www.ijraset.com)

Call:  08813907089

E-mail ID: [ijraset@gmail.com](mailto:ijraset@gmail.com)

# Daily News Classification and Trend Analysis System Using Machine Learning

Rathnaprasad D<sup>1</sup>, Akhilesh B<sup>2</sup>, Sarathy<sup>3</sup>

Department of Artificial Intelligence and Data Science

**Abstract:** *The digital age has catalyzed an explosion in the volume and velocity of news content published every hour across hundreds of online platforms. The inability to automatically organize, categorize, and extract meaningful signal from this torrent of information represents a significant challenge for media companies, policymakers, researchers, and the general public alike. This paper presents a comprehensive Daily News Classification and Trend Analysis System that integrates multiple machine learning (ML) algorithms with Natural Language Processing (NLP) techniques to build an end-to-end, scalable news intelligence pipeline. The system ingests raw news articles from heterogeneous sources, applies a rigorous preprocessing pipeline including tokenization, stop word removal, stemming, and lemmatization, and then extracts discriminative features using TF-IDF vectorization with n-gram extensions. Three classification models—Multinomial Naive Bayes, Support Vector Machine (SVM) with linear kernel, and Logistic Regression—are trained and benchmarked on standard corpora. The best-performing classifier (SVM with TF-IDF bigrams) achieves 94.1% accuracy on the AG News corpus. A dedicated trend analysis module applies moving averages, rate-of-change measures, and Kleinberg's burst detection algorithm to identify temporally emerging and declining topic categories. Real-world experiments over a six-month news archive confirm the system's ability to detect genuine trend patterns aligned with known world events. An interactive dashboard provides real-time visualization of category-level trends, enabling actionable intelligence for media monitoring, public opinion analysis, and misinformation detection pipelines. The system is modular, language-extensible, and deployable on standard commodity hardware.*

**Keywords:** *News Classification, Machine Learning, Natural Language Processing, Trend Analysis, TF-IDF, Support Vector Machine, Naive Bayes, Logistic Regression, Text Mining, Burst Detection, Information Retrieval, Topic Modeling.*

## I. INTRODUCTION

The World Wide Web hosts billions of webpages, and thousands of new news articles are published every hour across global media outlets, aggregators, and social platforms. The Reuters news agency alone distributes approximately 2,000 news stories per day, while digital-native portals like Google News, Flipboard, and Yahoo News aggregate content from tens of thousands of sources in real time. The volume of this content is far beyond the reading capacity of any individual or editorial team, creating a critical bottleneck in how societies consume and act upon news.

Automated text classification, a branch of supervised machine learning, offers a scalable solution to the problem of news categorization. By training statistical models on labeled corpora, it is possible to assign incoming articles to semantic categories—such as Politics, Sports, Technology, Business, Health, Entertainment, and Science—with high accuracy and at machine speed. These categories serve as the foundation for downstream applications including personalized news recommendation, media monitoring, sentiment tracking, and event detection.

Trend analysis extends the value of classification by introducing a temporal dimension. Once articles are categorized, their publication frequencies over time constitute a time series that encodes the ebb and flow of public and media attention. Detecting when a category's volume deviates significantly from its baseline reveals emerging events, viral news cycles, and waning interest in previously dominant topics. Such insights are invaluable to intelligence analysts, public health agencies, political campaigns, and financial institutions. Despite the rich body of literature on both text classification and trend detection, few systems address both tasks in a unified,

production-oriented pipeline tailored for daily news. Existing approaches either focus narrowly on classification accuracy or treat trend detection as a social-media-specific problem, ignoring the structural and stylistic characteristics of professional journalism. This work bridges that gap by proposing a tightly integrated, end-to-end Daily News Classification and Trend Analysis System.

The principal contributions of this paper are as follows:

- A complete modular pipeline from raw article ingestion to trend visualization, designed for daily operational use.
- A rigorous comparative evaluation of three supervised ML classifiers under consistent experimental conditions on two standard benchmarks.
- A multi-signal trend analysis module combining moving averages, rate-of-change analysis, and burst detection to surface temporally significant topics.
- Quantitative validation of trend detection against a manually annotated ground-truth timeline of known real-world events.
- An interactive, real-time trend dashboard deployable on standard commodity hardware.

The rest of this paper is organized as follows: Section II surveys the relevant prior work. Section III describes the proposed system architecture and methodology in detail. Section IV presents the experimental setup, datasets, and results. Section V discusses the practical applications and limitations of the system. Section VI draws conclusions and outlines future research directions.

## II. LITERATURE REVIEW

### A. Early Approaches to Text Classification

Text classification has its roots in information retrieval research from the 1960s and 1970s, where Boolean keyword-matching rules were used to filter documents of interest. The introduction of probabilistic retrieval models by Robertson and Sparck Jones (1976) marked a paradigm shift toward statistical approaches. Hayes and Weinstein (1990) applied rule-based expert systems to news routing at Reuters, achieving moderate performance at the cost of massive hand-engineering effort [1].

The Naive Bayes classifier, adapted for text by McCallum and Nigam (1998), demonstrated that simple probabilistic models could outperform many rule-based alternatives on the 20 News groups dataset when trained on sufficient data [2]. Despite its strong conditional independence assumption, Naive Bayes remained competitive for decades and is still widely used in production systems due to its low computational cost and resistance to overfitting.

### B. Support Vector Machines and Feature Engineering

Joachims (1998) provided a landmark demonstration of Support Vector Machines for text categorization, showing superior performance to decision trees, k-NN, and Naive Bayes on the Reuters-21578 benchmark [3]. The key insight was that the sparse, high-dimensional nature of TF-IDF feature spaces suits the maximum-margin criterion of SVMs particularly well. Subsequent work by Yang and Liu (1999) confirmed these findings across multiple datasets and task formulations.

Salton and Buckley's (1988) TF-IDF weighting scheme, combined with n-gram extensions, became the canonical feature extraction strategy for text classification throughout the 2000s [4]. Sublinear TF scaling and L2 normalization further improved performance by reducing the influence of very frequent terms and equalizing document length effects.

### C. Deep Learning and Transformer Models

Kim (2014) introduced a simple CNN architecture over static word embeddings that achieved state-of-the-art results on multiple sentence classification benchmarks, demonstrating that local feature detectors could capture important phrasal patterns [5]. Subsequent RNN-based models with LSTM and attention mechanisms improved performance on longer documents where local context alone was insufficient.

The transformer architecture introduced by Vaswani et al. (2017) and subsequently operationalized as BERT by Devlin et al. (2019) represents the current state of the art for most NLP classification tasks [6]. Fine-tuned BERT models routinely achieve accuracy above 97% on AG News and similar benchmarks. However, their computational requirements — large GPU memory, long training times, and high inference latency — render them impractical for high-throughput real-time applications on commodity hardware. This paper therefore focuses on classical ML methods, which offer an effective and efficient solution for resource-constrained deployments.

### D. Topic Detection and Trend Analysis

The Topic Detection and Tracking (TDT) research program, initiated by DARPA in 1996 and formalized by Allan et al. (1998), established the foundational framework for identifying new events in news streams and monitoring their evolution over time [7]. Early TDT systems relied on cosine similarity between TF-IDF article vectors to cluster temporally proximate articles covering the same event.

Kleinberg (2003) proposed a computationally elegant burst detection algorithm based on an infinite-state automaton that models the emission rate of a feature as switching between low-frequency and high-frequency states. The algorithm identifies contiguous time intervals during which the emission rate is anomalously elevated, corresponding to bursts of topical activity [8]. This algorithm is particularly well suited to news stream analysis because it is parameter-free in its core formulation and robust to seasonal baseline variations.

More recent work has applied Latent Dirichlet Allocation (LDA) and nonnegative matrix factorization to unsupervised topic discovery in news archives (Blei et al., 2003). Dynamic topic models extend LDA to track topic evolution over time. While powerful, these unsupervised approaches require careful hyperparameter tuning and produce topics that are hard to map to human-interpretable categories compared to supervised classification. Our system leverages supervised classification for category assignment and reserves temporal modeling exclusively for trend detection, yielding a cleaner semantic structure.

### E. Gaps in Prior Work

While numerous works address news classification or trend detection in isolation, few provide an integrated, validated pipeline that handles both tasks and targets daily news specifically. Systems designed for social media (Twitter trend detection, Reddit topic modeling) do not account for the long text, structured format, and editorial standards of professional news articles. This work addresses that gap by combining robust supervised classification with multi-signal trend analysis in a single coherent system.

## III. PROPOSED SYSTEM ARCHITECTURE AND METHODOLOGY

The proposed system is organized into five sequential modules: (1) Data Ingestion, (2) Text Preprocessing, (3) Feature Extraction, (4) Classification, and (5) Trend Analysis and Visualization. Figure 1 illustrates the overall architecture. Each module is independently testable and replaceable, supporting future extensions such as multilingual support or deep learning classifiers.



Fig. 1. System Architecture: End-to-End News Classification and Trend Analysis Pipeline

### A. Data Ingestion

The system supports two ingestion modes. In offline mode, pre-labeled benchmark corpora (AG News, BBC News, 20 Newsgroups) are loaded from disk for training and evaluation. In online mode, articles are fetched in real time from the NewsAPI.org REST API, which aggregates content from over 80,000 sources worldwide. Each ingested article is represented as a JSON object containing the following fields: title (headline), description (lead paragraph), content (full body text), source name, category label (when available), and publishedAt (ISO 8601 timestamp).

For online articles lacking labels, the trained classifier assigns a predicted category. Articles with confidence scores below a configurable threshold (default: 0.70) are flagged for manual review. The ingestion layer also deduplicates articles using a locality-sensitive hashing scheme applied to the TF-IDF vector to prevent near-duplicate articles from inflating trend counts.

### B. Text Preprocessing Pipeline

Raw article text contains substantial noise including HTML entities, URLs, social media handles, legal disclaimers, bylines, and advertisement copy. The preprocessing module applies the following steps in sequence:

- 1) **HTML Stripping and Unicode Normalization:** BeautifulSoup removes residual HTML markup. Unicode text is normalized to NFC form and non-ASCII characters outside the Latin extended range are replaced with their ASCII equivalents or removed.
- 2) **Sentence and Word Tokenization:** The NLTK Punkt tokenizer splits text into sentences, and word tokenization is then applied. Tokenization respects abbreviations, decimal numbers, and hyphenated compounds.
- 3) **Lowercasing:** All tokens are converted to lowercase to merge surface variants (e.g., 'COVID' and 'covid').
- 4) **Stop Word Removal:** A 179-term stop word list from NLTK is augmented with 42 journalism-specific terms (e.g., 'said', 'told', 'according', 'reuters') that carry low semantic value for classification.
- 5) **Lemmatization:** WordNet lemmatization reduces tokens to their dictionary base form, respecting part-of-speech context (e.g., 'running' → 'run', 'better' → 'good' as adjective).

Named Entity Normalization: Person names and organization names identified by NLTK's Named Entity Recognition (NER) tagger are replaced with generic placeholders (PERSON, ORG, GPE) to reduce sparsity and improve generalization across topics.

- 6) Short Token Removal: Tokens of length one or two are discarded as they rarely carry semantic content after stop word removal. After preprocessing, the title and description fields receive 2x and 1.5x weight boosts respectively when concatenated with body text, reflecting the higher information density of these fields. This simple weightings scheme improved accuracy by 1.3 percentage points in ablation experiments.

### C. Feature Extraction

Two feature extraction strategies were implemented and compared. The primary strategy is TF-IDF vectorization, which computes for each term  $t$  in document  $d$  the product of its term frequency  $TF(t, d)$  and inverse document frequency  $IDF(t) = \log(N / (1 + df(t)))$ , where  $N$  is the corpus size and  $df(t)$  is the number of documents containing  $t$ . Sublinear TF scaling  $(1 + \log(TF))$  is applied to dampen the effect of high-frequency terms. Unigram and bigram features are extracted and L2-normalized per document, yielding sparse feature vectors of dimensionality up to 150,000 for large corpora.

The secondary strategy uses pre-trained 300-dimensional Word2Vec embeddings (Google News vectors, trained on 100 billion words) to represent each document as the mean of its constituent word vectors. This dense representation captures semantic similarity between articles but requires substantially more memory and loses the discriminative power of rare but highly diagnostic terms. In cross-validation experiments, TF-IDF outperformed Word2Vec by 2.8 percentage points on average, so TF-IDF was selected as the primary representation.

### D. Classification Module

Three widely used supervised classifiers are implemented using the scikit-learn library (Pedregosa et al., 2011) [9]:

**Multinomial Naive Bayes (MNB):** The MNB classifier applies Bayes' theorem with the conditional independence assumption. It models the likelihood of observing each term given a class label using a multinomial distribution over term counts. Laplace smoothing ( $\alpha=1.0$ ) prevents zero-probability estimates for unseen terms. MNB is highly efficient, requiring only a single pass over the training data, and is well suited for real-time classification of high-volume streams.

**Support Vector Machine—Linear Kernel (SVM-L):** The linear SVM finds the maximum-margin hyperplane in the TF-IDF feature space that separates training examples of each class from the rest (one-vs-rest multi-class formulation). The regularization parameter  $C$  is set to 1.0 after grid search. Linear SVMs are particularly effective for high-dimensional sparse feature spaces because the kernel computation reduces to a dot product, yielding  $O(n)$  prediction complexity per document.

**Logistic Regression (LR):** Logistic Regression models the posterior probability of each class as a softmax function of a linear combination of features. L2 regularization with  $C=5.0$  prevents overfitting. The model is trained using the LBFGS optimizer with a maximum of 1,000 iterations. LR produces calibrated probability estimates, which are used by the downstream confidence-filtering mechanism during online ingestion.

All models are trained on 80% of the data and evaluated on the held-out 20% test set. Stratified 5-fold cross-validation is used during hyperparameter search to ensure balanced class representation in each fold. Final test evaluation uses macro-averaged F1-score and per-class precision, recall, and F1 to capture performance across imbalanced category distributions.

### E. Trend Analysis and Visualization Module

The trend analysis module operates on the temporal dimension of the classified article stream. For each category  $c \in C$  and each discrete time window  $w$  (daily, weekly, or monthly), the module computes the article frequency  $f(c, w)$ —the count of articles assigned to category  $c$  during window  $w$ . This produces a multivariate time series with  $|C|$  dimensions. Three complementary trend detection signals are computed:

**Moving Average (MA):** A 7-day centered moving average  $MA(c, w) = (1/7) \sum_{k \in \{-3, \dots, 3\}} f(c, w+k)$  smooths daily fluctuations. The deviation of the raw count from the moving average highlights transient spikes and dips relative to the local baseline.

**Rate of Change (RoC):** The week-over-week percentage change  $RoC(c, w) = 100 \times (f(c, w) - f(c, w-7)) / f(c, w-7)$  measures the velocity of topic growth or decline. Categories with  $|RoC| > 50\%$  over a three-day window are flagged as rapidly trending.

**Burst Detection:** Kleinberg's infinite-state automaton is applied independently to each category's time series. The algorithm models the emission rate of the category as switching between a low-frequency background state ( $q=0$ ) and an elevated burst state ( $q=1$ ).

A burst is detected when the Viterbi path through the automaton enters state  $q=1$  for at least two consecutive time windows. The burst level, cost function, and scaling factor are set to default values ( $s=2, \gamma=1$ ) as recommended in the original work. Detected trends are persisted to a lightweight SQLite database indexed by (category, window, signal\_type). The visualization dashboard, built with Plotly Dash, renders the following views: (1) multi-line time series of normalized category frequencies, (2) a heatmap of burst alerts across categories and time, (3) per-category word cloud highlighting the most discriminative terms in trending windows, and (4) a summary table of the top trending and declining categories for the current rolling 7-day window. The dashboard auto-refreshes every 15 minutes when connected to the online ingestion API.

#### IV. EXPERIMENTAL SETUP AND RESULTS

##### A. Datasets

Two benchmark datasets were used in this study:

**AGNews Corpus:** The AGNews dataset contains 120,000 training articles and 7,600 test articles distributed equally across four categories: World, Sports, Business, and Science/Technology. Articles consist of a title and a 1–3 sentence description. The class distribution is perfectly balanced (30,000 training articles per class), making it a clean benchmark for classifier comparison.

**BBC News Dataset:** The BBC News dataset (Greene & Cunningham, 2006) contains 2,225 articles from the BBC News website published between 2004 and 2005, categorized into five classes: Business, Entertainment, Politics, Sport, and Tech. Unlike AG News, this dataset contains full article bodies of several hundred words, making it a more realistic test of classification on longer texts.

Table I: Dataset Statistics

Dataset	Categories	Train Articles	Test Articles	Avg. Words/Article
AGNews Corpus	4	120,000	7,600	43
BBCNews Dataset	5	1,780	445	389

##### B. Classification Results

Table II summarizes classification performance on the AGNews test set. All models use TF-IDF with unigrams and bigrams (max features = 100,000) as the primary feature representation. The SVM with TF-IDF bigrams achieved the highest accuracy of 94.1%, closely followed by Logistic Regression at 91.2%. Naive Bayes, while lower in accuracy, was substantially faster in both training and inference.

Table II: Classification Performance on AGNews Dataset (TF-IDF Bigrams)

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Multinomial Naive Bayes	88.4	88.7	88.4	87.9
Logistic Regression	91.2	91.5	91.2	90.8
SVM (Linear, Unigram)	92.8	93.0	92.8	92.5
SVM (Linear, Bigram)	94.1	94.3	94.1	93.8
SVM (Linear, Bigram, W-boost)	94.7	94.9	94.7	94.5

The best overall model — SVM with TF-IDF bigrams and the title/description weight boost — achieves 94.7% accuracy. Table III shows per-class results for this model on AG News, revealing that the Sports and Science/Technology categories are classified most reliably, while World news is marginally harder due to its thematic breadth.

Table III: Per-Category Results — SVM (Bigram + Weight Boost) on AG News

Category	Precision (%)	Recall (%)	F1-Score (%)	Support
World	93.2	93.8	93.5	1,900

Sports	97.1	97.4	97.3	1,900
Business	93.9	93.4	93.6	1,900
Science/Technology	95.4	94.3	94.9	1,900

Table IV presents classification results on the BBC News dataset. The longer article bodies and five-class structure make this a harder problem, but SVM still leads at 97.3% accuracy, benefiting from the richer feature space derived from full article bodies.

TableIV:ClassificationPerformanceonBBCNewsDataset

Classifier	Accuracy(%)	Precision(%)	Recall (%)	F1-Score(%)
MultinomialNaiveBayes	95.7	95.9	95.7	95.6
LogisticRegression	96.4	96.6	96.4	96.3
SVM(Linear,Bigram)	97.3	97.5	97.3	97.2

### C. AblationStudy

An ablation study was conducted to quantify the contribution of each preprocessing and feature engineering step. Table V shows the effect of progressively removing components from the full pipeline on AG News classification accuracy.

TableV:AblationStudy—AGNewsAccuracy(SVMClassifier)

Configuration	Accuracy (%)
FullPipeline(Baseline)	94.7
WithoutTitle/DescriptionWeightBoost	93.4
WithoutNamedEntityNormalization	93.8
WithoutLemmatization(StemmingOnly)	94.1
WithoutStopWordRemoval	93.0
UnigramsOnly(NoBigrams)	92.8
RawText(NoPreprocessing)	90.5

The results confirm that each preprocessing step contributes positively to the overall performance. The bigram feature extensionandtitle/descriptionweightingprovidethelargestindividualgains(+1.9%and+1.3%respectively).Namedentity normalization provides a modest but consistent benefit by reducing feature space sparsity, while the raw text baseline demonstrates that preprocessing collectively contributes 4.2 percentage points of accuracy improvement.

### D. TrendDetectionEvaluation

The trend analysis module was evaluated on a six-month archive (January–June 2024) of approximately 180,000 articles collected through the NewsAPI.org live feed. A ground-truth timeline was manually annotated by three independent annotators, identifying 34 distinct trend events — defined as periods of at least three consecutive days during which a category's article volume exceeded two standard deviations above its 30-day rolling mean. Inter-annotator agreement, measured by Fleiss' kappa, was 0.81 (strong agreement).

Table VI summarizes trend detection performance across the three detection signals and their ensemble combination. The ensemble method, which triggers a trend alert when atleasttwo of thethreesignals independently agree, achieves thebest balance of precision and recall.

TableVI:TrendDetectionPerformance(6-MonthNewsArchive,34Ground-TruthEvents)

DetectionMethod	Precision(%)	Recall(%)	F1-Score(%)	Latency(days)
MovingAverageOnly	78.3	82.4	80.3	2.1
RateofChangeOnly	81.5	76.5	78.9	1.3
BurstDetectionOnly	89.2	85.3	87.2	1.8
Ensemble(2-of-3Voting)	91.4	88.2	89.8	1.5

The ensemble method correctly identified 30 of 34 annotated trend events with a precision of 91.4%, meaning fewer than 9% of alerted trends were false positives. The averaged detection latency of 1.5 days is operationally acceptable for daily news monitoring. The four missed events were low-intensity trends in the Entertainment category with gradual onset, which fell below all three detection signal thresholds.

E. System Throughput and Scalability

Throughput benchmarks were conducted on a server with an Intel Core i7-12700K CPU (12 cores, 3.6 GHz base), 32 GB RAM, and no GPU. The system achieves the following throughput figures:

TableVII: System Throughput Benchmarks (12-core CPU, No GPU)

Pipeline Stage	Throughput/Latency
Text Preprocessing	~1,200 articles/minute
TF-IDF Vectorization	~3,500 articles/minute
SVM Classification (Inference)	~8,000 articles/minute
End-to-End Pipeline	~900 articles/minute
Trend Dashboard Refresh	Every 15 minutes (configurable)
SQLite Write (trendlog)	<5ms per batch

The end-to-end throughput of approximately 900 articles per minute comfortably exceeds the peak ingestion rate from NewsAPI.org (~400 articles per minute at scale), confirming that the system can sustain real-time operation without queuing delays. Preprocessing is the principal bottleneck, accounting for 58% of the total per-article processing time. Parallelizing preprocessing across all 12 CPU cores increases throughput to over 5,000 articles per minute.

V. APPLICATIONS, CHALLENGES, AND LIMITATIONS

A. Applications

The Daily News Classification and Trend Analysis System has broad applicability across several domains:

- 1) Media Monitoring and Competitive Intelligence: News organizations and PR agencies can deploy the system to continuously monitor how different topics are covered across competitor outlets, identify underreported stories, and measure share-of-voice for specific brands or individuals across news categories.
- 2) Public Health Surveillance: Health agencies can monitor the volume and trend of health-related news articles to detect early signals of emerging disease outbreaks, public anxiety, or misinformation campaigns before they reach mainstream awareness.
- 3) Financial Market Intelligence: Financial analysts and algorithmic trading firms monitor new trend signals as leading indicators of sector sentiment shifts and macroeconomic events that may impact asset prices. The classification module's category tags enable sector-specific monitoring.
- 4) Political Campaign Analytics: Campaigns and policy think tanks can use the trend dashboard to monitor the media salience of policy issues, track competitor messaging, and measure the impact of their own communications on news coverage patterns.

- 5) Academic Research in Computational Social Science: Longitudinal trend data derived from classified news archives enables researchers to study agenda-setting dynamics, media framing effects, and the relationship between news coverage and public opinion over time.
- 6) Misinformation Detection Pipeline: The classification system provides a structured input layer for downstream fake news detection models, enabling them to focus analysis resources on the most topically active and potentially vulnerable categories.
- 7) Automated News Summarization and Briefing: The system can be extended with an extractive summarization module to generate category-specific daily briefings, providing decision-makers with concise digests of the most trend-aligned articles.

### B. Technical Challenges

Several technical challenges were encountered during development and are worth documenting for practitioners:

- 1) Label Noise in Online Data: Articles ingested via News API lack authoritative category labels. The confidence-filtering mechanism mitigates but does not eliminate labeling errors, which can accumulate and distort trend counts over time if unchecked.
- 2) Category Ambiguity: Many news articles span multiple categories (e.g., a story about an athlete's political activism is both Sports and Politics). The current one-vs-rest SVM formulation assigns a single label, which may not capture this multi-label reality. A threshold-based multi-label extension is a natural improvement.
- 3) Baseline Drift: Long-term shifts in news volume (e.g., seasonal patterns, changes in source coverage) can confound burst detection if the baseline window is too short. Adaptive baseline estimation over longer horizons mitigates this issue.
- 4) Named Entity Sparsity: Replacing named entities with generic placeholders improves generalization but loses information that can be critical for discriminating between topically similar articles in different categories (e.g., a Business story about a tech company versus a Technology story).

### C. Limitations

- 1) The system currently supports English-language articles only. Extending to Tamil, Hindi, or other Indian languages requires language-specific preprocessing tools and labeled corpora.
- 2) Classification accuracy degrades on articles from niche sub-domains (e.g., niche sports, esoteric science topics) that are underrepresented in the AG News and BBC training corpora.
- 3) The trend detection module measures the volume of coverage but not the sentiment or stance of coverage. A highly negative spike and a highly positive spike in Technology coverage are treated identically.
- 4) The system is not designed for streaming architectures (e.g., Apache Kafka, Spark Streaming). A microservices refactoring would be required for deployment at internet scale.
- 5) The dashboard is read-only and does not support annotation or feedback workflows that would enable active learning to improve classifier accuracy over time.

## VI. CONCLUSION AND FUTURE WORK

This paper presented a comprehensive Daily News Classification and Trend Analysis System that addresses the growing challenge of making sense of the high-velocity digital news ecosystem. The system integrates a robust NLP preprocessing pipeline, TF-IDF feature extraction with n-gram extensions, and three supervised classification models into a single coherent architecture. Extensive experiments on two benchmark datasets (AG News and BBC News) demonstrated that the SVM classifier with TF-IDF bigrams and title/description weight boosting achieves 94.7% and 97.3% accuracy respectively — highly competitive with more complex deep learning approaches at a fraction of the computational cost.

The trend analysis module, combining moving averages, rate-of-change analysis, and Kleinberg's burst detection in an ensemble voting scheme, identified 30 of 34 independently annotated trend events in a six-month real-world news archive with 91.4% precision and an average detection latency of 1.5 days. The system operates at approximately 900 articles per minute on commodity hardware, satisfying the real-time requirements of news monitoring applications. An interactive Plotly Dash dashboard provides stakeholders with intuitive access to classification and trend data through time-series plots, heatmaps, and category-level word clouds.

The work confirms that well-engineered classical ML pipelines remain highly effective for news classification in production settings, particularly where computational resources are constrained or low-latency inference is critical. The modular architecture ensures that individual components can be upgraded independently as better tools become available.

Several promising directions for future work have been identified. First, integrating transformer-based classifiers such as DistilBERT or RoBERTa, potentially via model distillation to reduce inference latency, is expected to yield further accuracy gains of 2–4 percentage points on harder multi-class benchmarks. Second, extending the system to support multilingual news ingestion — particularly for Indian regional language news — would substantially broaden its social impact. Third, incorporating a sentiment analysis layer into the trend module would enable direction-aware trend detection, distinguishing between positive and negative coverage spikes. Fourth, a multi-label classification extension would better handle the pervasive category ambiguity in real-world news. Finally, an active learning feedback loop integrated into the dashboard would allow domain experts to correct misclassifications and continuously improve model performance without requiring periodic full retraining.

## VII. ACKNOWLEDGMENT

The authors express their sincere gratitude to their mentor, Ms. M. Subashani, Assistant Professor, Department of Computer Science and Engineering, for her invaluable guidance, technical insights, and continuous encouragement throughout the course of this research. Her expertise in machine learning and natural language processing significantly shaped the direction and quality of this work. The authors also thank the Department of Computer Science and Engineering for providing access to the computational infrastructure used in the experimental evaluation, and the anonymous reviewers for their constructive feedback.

## REFERENCES

- [1] P.J. Hayes and S.B. Weinstein, "CONSTRUE/TIS: A system for content-based indexing of a database of news stories," in Proc. 2nd Annual Conf. Innovative Applications of Artificial Intelligence (IAAI), 1990, pp. 49–64.
- [2] A. McCallum and K. Nigam, "A comparison of event models for Naive Bayes text classification," in Proc. AAAI-98 Workshop on Learning for Text Categorization, Madison, WI, 1998, pp. 41–48.
- [3] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in Proc. 10th European Conf. Machine Learning (ECML), Chemnitz, Germany, 1998, pp. 137–142.
- [4] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.
- [5] Y. Kim, "Convolutional neural networks for sentence classification," in Proc. 2014 Conf. Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 1746–1751.
- [6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, Minneapolis, MN, 2019, pp. 4171–4186.
- [7] J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," in Proc. 21st Annual Intl. ACM SIGIR Conf. Research and Development in Information Retrieval, Melbourne, Australia, 1998, pp. 37–45.
- [8] J. Kleinberg, "Bursty and hierarchical structure in streams," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 373–397, Oct. 2003.
- [9] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Nov. 2011.
- [10] R. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems (NIPS)*, Lake Tahoe, NV, 2013, pp. 3111–3119.
- [11] D.M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Mar. 2003.
- [12] Y. Yang and X. Liu, "A re-examination of text categorization methods," in Proc. 22nd Annual ACM SIGIR Conf. Research and Development in Information Retrieval, Berkeley, CA, 1999, pp. 42–49.
- [13] A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, Long Beach, CA, 2017, pp. 5998–6008.
- [14] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in Proc. 23rd Intl. Conf. Machine Learning (ICML), Pittsburgh, PA, 2006, pp. 377–384.
- [15] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. Cambridge University Press, 2023. [Online]. Available: <https://d2l.ai>



10.22214/IJRASET



45.98



IMPACT FACTOR:  
7.129



IMPACT FACTOR:  
7.429



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24\*7 Support on Whatsapp)