Received: 10 September 2025 | Revised: 3 December 2025 | Accepted: 6 January 2026 | Published online: 24 February 2026

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2026, Vol. 00(00) 1–11
DOI: 10.47852/bonviewJCCE62027629

BON VIEW PUBLISHING

Embedded Sparsity and Ensemble Learning for Real-Time Textual Anomaly Detection: A SCADA-Inspired Benchmark on Lecture Transcript Data

Ayoub Alsarhan[1, 2], **Saja Jaradat**[2], **Suhaila Abuowaida**[3], **Sami Aziz Alshammari**[4,*], **Nayef Hmoud Alshammari**[5] **and Khalid Hamad Alnafisah**[6]

[1] *Department of Data Science and Artificial Intelligence, Al-Ahliyya Amman University, Jordan*

[2] *Department of Information Technology, The Hashemite University, Jordan*

[3] *Department of Data Science and Artificial Intelligence, Al al-Bayt University, Jordan*

[4] *Department of Information Technology, Northern Border University, Saudi Arabia*

[5] *Department of Computer Science, University of Tabuk, Saudi Arabia*

[6] *Department of Computer Sciences, Northern Border University, Saudi Arabia*

Abstract: This study presents a comprehensive benchmark of feature selection methods for real-time text classification, evaluating the computational trade-offs between embedded sparsity techniques (Lasso regularization), filter methods (ANOVA, mutual information), and wrapper approaches (recursive feature elimination). Using a dataset of 15,746 educational comments with TF-IDF representations, we systematically compare five feature selection methods paired with both classical machine learning and shallow neural network classifiers. Our experimental model evaluates both classification performance and computational latency in the context of binary and multiclass sentiment classification. Results show that the Lasso-based feature selection, combined with XGBoost, achieves $F1$-scores of 0.859 for binary classification and 0.699 for multiclass classification, with inference times of less than 1 s. Recursive feature elimination takes 284 s to do similar performance. Shallow neural networks achieve higher accuracy ($F1 = 0.910$ for binary, 0.841 for multiclass) at the cost of 8-s training times. In contrast, convolutional neural networks (CNNs) applied directly to TF-IDF vectors perform poorly ($F1 = 0.646$) with excessive training overhead (126 s), indicating that standard CNN architectures are unsuitable for sparse text representations. Our findings offer practical guidance for selecting feature reduction and classification methods based on latency requirements: Lasso with ensemble methods for real-time applications that require sub-second response and shallow neural networks for batch processing, where higher accuracy justifies the additional computational cost. Although this work uses educational text data as a testbed, the methodology and findings are applicable to any high-dimensional text classification scenario requiring efficient feature selection.

Keywords: feature selection, intrusion detection, SCADA systems, text classification, gradient boosting

1. Introduction

Supervisory Control and Data Acquisition (SCADA) technologies have been the backbone of critically important infrastructure systems such as electric power grids, water treatment systems, and the operation of transportation networks. The growing interconnectedness of such systems has also increased their vulnerability to cyberattacks, and as a result, there has been a need to develop intrusion detection schemes that can guarantee high detection accuracy and low latency when handling streaming data. Conventional signature-based methods can mostly be ineffective in detecting new attack vectors or in cases where the attacker tries to obfuscate them, particularly in dynamic threat landscapes where data-centric detection models are needed. Recently, there has been a significant increase in the adoption of machine learning and deep learning systems in SCADA

***Corresponding author:** Sami Aziz Alshammari, Department of Information Technology, Northern Border University, Saudi Arabia. Email: Sami.Alshammari@nbu.edu.sa

systems, which allow detecting the presence of abnormal behavior in sensor data, control instructions, and network traffic. Random Forests [1] and XGBoost [2] are examples of ensemble-based classifiers that have achieved good performance in the classification of imbalanced SCADA data [3, 4]. Simultaneously, feature-selection methods such as Lasso [5] and recursive feature elimination (RFE) [6], which are embedded within feature-selection methods, have demonstrated effectiveness in high-dimensional input space reduction without loss of classification performance [5, 7]. These developments follow other recent natural language processing (NLP) developments, in which TF-IDF representations of features alongside feature-shallow neural architectures [8, 9] can be configured into a near-state-of-the-art performance at a relatively low computational cost.

Nevertheless, the use of these methods in textual surveillance, for example, sentiment analysis of operator remarks or log entries, is relatively under-researched in SCADA-related scenarios. Additionally, a comparative analysis of classical machine learning pipelines versus lightweight deep learning models to conduct text-based anomaly detection in a consistent setting of the feature selection under similar conditions is not present in the existing literature to a large extent. This paper builds upon a SCADA-inspired intrusion detection system in a new NLP context, making use of a large dataset of lecture transcripts and user-generated comments as a proxy to textual monitoring feeds in the critical infrastructure context. While the SIGHT dataset serves as a useful surrogate, it does not fully capture the complexity of SCADA logs, where anomalies are primarily associated with security-critical events such as unauthorized system access or operational faults. The text in SIGHT, on the other hand, is more reflective of user sentiment, introducing different types of anomalies such as negative feedback or confusion.

This study adopts feature selection methodologies that have proven successful in SCADA intrusion detection systems—specifically embedded sparsity techniques like Lasso regularization—and evaluates their effectiveness in a different domain: real-time text classification. We do not claim direct applicability to SCADA systems; rather, we investigate whether computational efficiency gains observed in SCADA-IDS literature transfer to general text classification tasks. This methodological borrowing is motivated by similar constraints: both domains require sub-second inference on high-dimensional data streams.

2. Theoretical Background

Machine learning-based predictive systems require advanced methods of feature selection and classification to compromise predictive accuracy and computational efficiency. The canonical review by Alimi et al. [5] divides feature-selection algorithms into three categories: filters, which make use of statistical tests, including the Analysis of Variance (ANOVA) F-test and mutual information; wrappers, which apply sequential forward/backward selection; and embedded algorithms, among which are Lasso and RFE. The study of Karthigha et al. [6] further expands this taxonomy by surveying those algorithms that optimize scores of filters, use heuristic search, or include feature selection in the training of the model. In practice, filters are computationally quick but can fail to capture the interaction of features, whereas wrappers and embedded methods do capture the interaction, at the penalty of additional runtime [7].

Ensemble classifiers have been developed into powerful detectors in a high-dimensional environment. Random Forests [1] use bootstrap aggregation and decision-tree variety to minimize the variance, whereas gradient boosting machines [10] sequentially improve predictions by refining them with a residual fit. XGBoost [2] is another implementation of a gradient boosting framework with system-level optimization and regularization, which makes it a standard in tabular tasks. AdaBoost [11] re-weights misclassified samples in order to concentrate on difficult cases. Naive Bayes [12, 13] is a lightweight baseline on conditional-independent assumptions.

The convolutional neural networks (CNNs) and feed-forward artificial neural networks (ANNs) are deep learning structures that provide end-to-end feature learning using raw data. Gradient-based learning has proven itself to be effective in document recognition [8], whereas CNNs with one-dimensional convolutions became popular in text classification by Li et al. [9]. The previous research by Chaseon et al. [14] on ANNs in anomaly detection of network traffic emphasizes their versatility, but the increased training costs of deep learning require special architectural and input-representation decisions.

In security-related areas, performance measurement plays a very important role. In a systematic study on the evaluation metrics, de Diego et al. [15] focus on using $F1$-scores on the case of imbalance. To safeguard accuracy and applicability, Erickson and Kitamura [16] underscore the best practices of reporting precision, recall, area under the curve, and runtime.

In SCADA contexts, recent studies by Ahakonye et al. [17, 18] leverage modified decision trees and chi-square selection to address high-dimensional network logs, while Alimi et al. [19] apply supervised machine learning pipelines to power-grid datasets. Wali and Alshehry [20] emphasize the need for lightweight, real-time analytics capable of adapting to evolving threat signatures. Building on these theoretical foundations, this work systematically compares classical and deep learning models under unified feature-selection schemes on large text corpora, bridging the gap between SCADA intrusion detection and text-based monitoring.

3. Related Work

Recently, adopting feature selection and machine learning for intrusion detection in the SCADA context has gained more attention. To deal with high-dimensional data of SCADA networks, Ahakonye et al. [17, 18] suggest using hybrid decision-tree models that combine chi-square and customized tree induction, which yield substantial improvements in detection accuracy and efficiency. Alimi et al. [19] use supervised learning classifiers to power-grid SCADA systems with particular emphasis on feature selection specifically tailored to robustness in intrusion detection.

In addition to SCADA applications, Wali and Alshehry [20] present detailed overviews of the current security issues in industrial control systems, highlighting the necessity of lightweight and real-time analytics that should be flexible against changing threat profiles. Gumaei et al. [21] propose new methods for cyberattack detection within smart grids by choosing the best subsets of SCADA features, whereby high detection rates are reported with minimal false notifications. Upadhyay et al. [22] refine this study by integrating gradient boosting feature selection in intrusion detection system pipelines on power networks and prove that sparse sets of features can be used to improve accuracy and minimize latency. Similarly, Rajesh and Satyanarayana [4] compare various machine learning algorithms with SCADA traffic data; the results stress the fact that ensemble algorithms combined with an effective feature-selection method usually outperform single classifiers.

Machine learning algorithms have significantly improved the performance of feature-selection and classification methods. Karthigha et al. [6] provide extensive literature reviews of filter, wrapper, and embedded selection techniques, creating a basis on which statistical tests are compared with sparsity-inducing models. The introduction of scalable tree boosters, such as XGBoost [2], has changed the best practice toward gradient-boosted ensembles able to effectively consume sparse inputs. The background of performance metrics, as discussed by de Diego et al. [15] and Erickson and Kitamura [16], informs evaluation structures, which are sensitive to report the accuracy, recall, and F1 measures in a binary and multiclass setup.

These studies indicate that two important issues remain unresolved:

1) Extracting minimal yet informative feature subsets to make fast inferences.
2) Utilizing modern ensembles or neural architectures to achieve optimal detection results. This work is based on the existing literature by benchmarking the classical and deep models using different feature-selection methods on new text-based datasets in a systematic way to show how techniques used in SCADA intrusion detection systems can be applied to more general monitoring tasks.

Recent advances in real-time text classification have focused on balancing accuracy with computational efficiency. Kowsari et al. [23] provide a comprehensive survey of text classification algorithms, reporting $F1$-scores ranging from 0.80 to 0.92 for traditional methods on benchmark datasets. Minaee et al. [24] survey deep learning approaches for text classification, showing Bidirectional Encoder Representations from Transformers (BERT) achieves 0.89–0.94 $F1$-scores but with significant computational overhead. Qader et al. [25] specifically examine feature selection impact on classification speed, demonstrating 40–60% inference time reduction with minimal accuracy loss. The baseline approach for text classification is investigated by Joulin et al. [26]. Wang et al. [27] propose a convolutional recurrent neural network for text classification. An analogy between categorical time series and classical NLP was formalized in Horak et al.'s work [28], and the strength of this analogy for anomaly detection and root cause investigation was demonstrated through the implementation and testing of three different machine learning models based upon it. NLP was utilized by Yadav et al. [29] for anomaly detection. The word embeddings technique was employed [29], with words being represented as points in a high-dimensional space. In this space, words with similar meanings were placed close to each other. The dataset was cleaned at the beginning of the process, and valuable features were extracted during the preprocessing step using sophisticated techniques such as word embeddings and sentiment analysis. The system was proposed by Nguyen et al. [30], designed not only to distinguish normal HTTP requests from well-known attack patterns but also to detect emerging types of anomalous attacks. It consisted of two models that integrated NLP approaches, deep learning techniques, and transfer learning strategies. The first model was employed to detect new anomalous HTTP requests that differed from normal requests. HTTP requests identified as anomalous were then transmitted to the second model, which was responsible for classifying specific categories of both well-known and novel attacks.

The recent development of AI-based models has made a major contribution to different industries [31–39]. Mohammed et al. [31] note the transformative effect of AI in language learning and security. The integration of federated learning to support the detection of IoT intrusion is discussed by Nyangaresi [33], as the increasing role of AI in security and education is highlighted. Moreover, Nyangaresi discusses the role of AI in improving security in industries [33].

4. Dataset Description

The experimental analyses presented in this study utilize the SIGHT dataset, an open-source collection originally curated for educational video analysis [40]. The dataset comprises lecture transcripts and user-generated comments, providing rich textual resources suitable for benchmarking sentiment classification and anomaly detection tasks. The dataset structure and associated preprocessing pipeline are described in the following subsections.

4.1. Data sources

1) Lecture transcripts: This topic is found in the lectures/data/-transcripts/directory, which consists of 123 UTF-8 plain-text files (UTF-8), each of which is the transcript of a specific lecture. When introduced to a pandas DataFrame (df_transcripts), it was determined that the average length of a transcript was about 4500 words.
2) User comments: The comments are stored in one array in the form of a single JSON file, which is at sight/data/comments/-comments.json, and there are 15,746 records. The following fields are contained in every entry:
   a. video_id: a reference to the associated lecture file (without the extension).
   b. comment_text: the raw text contents that are entered by users.
   c. playlist_title: the title of the course or the video series.
   d. video_sequence_idx: the sequence of the lecture in the playlist.

4.2. Preprocessing

Data preprocessing was carried out on Cells 4–5 of the associated notebook. Preprocessing includes the following major steps:

1) Deduplication: Duplicate rows were removed in both the transcript and comment Data Frames with perfect accuracy to give integrity of the data.
2) Text cleaning: All the textual fields were converted to lowercase. Non-alphanumeric characters (except spaces) were eliminated, newline characters were changed to single spaces, and several whitespace characters were merged. Two cleaned-up columns were obtained in these operations—transcript_clean and comment_clean—both of which are devoid of missing values.
3) Merging: df_comments was merged with df_transcripts by the use of a key match (video id = lecture id). The outcome was a single DataFrame (df) consisting of 15,746 rows, with every row having the corresponding lecture transcript and the corresponding user comment.
4) Class distribution analysis: Class distribution analysis reveals that 58 positive and 42 non-positive samples used binary classification and 31 negative, 42 neutral, and 27 positive used multiclass tasks. We ensured that there is no dominant lecture

in the data set and the distribution of the comments is quite balanced among all 123 lectures (mean: 128 comments/lecture, SD: 47).

4.3. Initial feature extraction

The consolidated dataset had three sets of preliminary features that were derived. The construction of vectors of TF-IDF involved scikit-learn's TfidfVectorizer, limited to the 5000 most common words in the aggregate of all the comments. Comment length was calculated as one of the scalar features that estimates the number of words in each of the cleaned comments. TextBlob was used to produce sentiment polarity of the text with the rule-based sentiment analyzer that produced continuous values of −1 to +1. These extracted attributes were the input to subsequent feature-selection and classification methods in our work.

5. Methodology and Experimental Setup

The proposed framework consists of two key elements:

1) Feature selection method: It is designed to reduce the dimensionality of input space and remove redundant or irrelevant features, thus enhancing model generalization and interpretability. This process improves intrusion detection accuracy by emphasizing the most appropriate information.
2) Using the reduced feature sets: Reduced feature sets are inputted to a pool of classifiers, which include conventional machine learning structures, as well as neural structures. In binary and multiclass sentiment classification tasks, these classifiers are used in benchmarking.

5.1. Methodology

An efficient selection of feature techniques should be selected when machine learning algorithms are used with high-dimensional textual data (TF-IDF vectors, etc.). This is done to select as many informative features as possible, thereby improving classification accuracy, decreasing overfitting, and reducing computational cost.

5.1.1. Feature engineering

Three complementary feature sets are derived from raw transcript-comment pairs. TF-IDF vectors extract term frequency-inverse document frequency representations from cleaned comment text, limiting to the top 5000 features by document frequency to control dimensionality.

Comment length captures scalar features representing word count per comment, which can correlate with expressive richness or verbosity. Sentiment polarity provides continuous scores (−1 to +1) computed via TextBlob, quantifying the overall valence of each comment. These features provide both high-dimensional lexical signals and low-dimensional summary statistics, enabling comparative evaluation of classical and deep models. In our work, the regularization strength (lambda) for Lasso was determined through a 3-fold cross-validated grid search. We selected the lambda that minimized binomial or multinomial deviance based on the task type.

5.1.2. Feature-selection techniques

To reduce computational costs and improve generalization, five feature selection methods are evaluated. Sequential Forward Selection and Sequential Backward Selection employ greedy wrapper approaches that iteratively add or remove features based on cross-validated performance with logistic regression base learners. ANOVA F-test utilizes filter methods selecting features with the highest F-statistics between TF-IDF terms and sentiment labels. Mutual information applies nonparametric filters measuring information gain between each feature and target. Lasso employs ℓ_1-penalized logistic regression as embedded methods that drive many coefficients to zero, naturally yielding sparse feature sets. RFE uses wrapper approaches that recursively prune the weakest coefficients from logistic regression estimators. The number of selected features is fixed at 20 for fair comparison, except for Lasso, where selection is determined by regularization strength. In addition to the feature selection methods applied to classical machine learning models, we also explored the impact of feature selection on ANN and CNN. Specifically, we tested Lasso-selected features (~20 features) for ANN models, following the same feature selection process applied to other classifiers. However, when Lasso-selected features were applied to the ANN model, we observed a noticeable drop in performance. The binary F1-score decreased to approximately 0.82, which positioned the ANN performance between the classical models and the full-dimension ANN.

This result suggests that the ANN's performance advantage stems partly from its ability to learn complex interactions from the high-dimensional TF-IDF vector. Reducing the feature set limited the model's ability to capture these complex interactions, leading to a decrease in performance. This finding indicates a clear trade-off: classical models using reduced features offer faster computational times, while ANNs using the full feature set provide higher accuracy at the cost of increased computational time. These results inform the selection of models for real-time applications, where the choice between speed and accuracy depends on the specific use case and operational constraints.

5.1.3. Models

There are two types of models that are benchmarked: the classical machine learning methods that include Random Forest, Decision Tree, Gaussian Naive Bayes, XGBoost, and AdaBoost and deep learning methods that comprise a shallow feed-forward network with 6432 units and dropout trained on TF-IDF input and a 1D convolutional network that uses TF-IDF vectors as pseudo-sequences through Conv1D GlobalMaxPool.

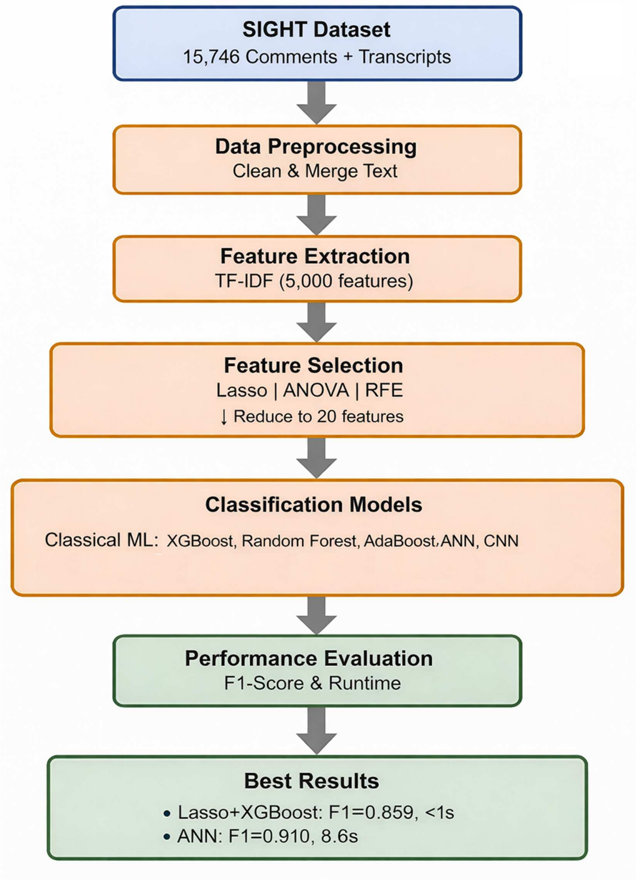
5.2. Experimental setup

The experimental framework follows a systematic pipeline designed to ensure reproducible and comprehensive evaluation of feature selection and classification methods. Figure 1 illustrates the complete methodology workflow from data preprocessing through model evaluation. Figure 1 presents the experimental methodology flowchart showing seven sequential stages from data input to performance evaluation. The pipeline processes the SIGHT dataset through preprocessing, extracts TF-IDF and auxiliary features, splits data for binary and multiclass tasks, applies five feature selection methods, trains seven classification algorithms, and evaluates performance using standard metrics and runtime analysis.

5.2.1. Train/test splits

All experiments employ 80/20 stratified splits of the full dataset. Binary tasks classify comments as positive (polarity > 0) or non-positive (≤ 0). Multiclass tasks discretize comments

Figure 1
Methodology flowchart



into negative (≤ -0.1), neutral (-0.1 to 0.1), and positive (> 0.1) categories.

We employ 5-fold stratified cross-validation for all experiments, reporting the mean and standard deviation of metrics across folds. Statistical significance between methods is assessed using paired t-tests with Bonferroni correction for multiple comparisons ($\alpha = 0.05$/comparisons). The mean $F1$-scores from cross-validation were closely aligned with those from the holdout set (deviations $< \pm 0.015$), confirming the robustness of the reported results.

5.2.2. Evaluation metrics

Standard classification metrics are reported including accuracy, precision, recall, and $F1$-score for binary tasks and weighted precision, weighted recall, and weighted $F1$-score for multiclass tasks. Runtime measurements capture wall-clock time for feature selection and model training/inference, measured separately.

5.2.3. Hardware and software environment

All experiments were conducted on personal workstations running Windows 10 (64-bit), equipped with 11th Generation Intel Core i7 CPUs and 32 GB RAM. Python 3.8 within JupyterLab was utilized, leveraging scikit-learn 1.1 for classical models, XGBoost 1.5, and TensorFlow 2.6 for deep learning. This configuration reflects typical engineering laptop environments, demonstrating pipeline feasibility without specialized high-performance hardware.

6. Results and Discussion

This section presents and interprets experimental outcomes on the SIGHT lecture transcript and student-comment dataset. The classical machine learning results on binary sentiment tasks are provided first and then on multiclass sentiment tasks, and the results are compared with two deep learning architectures. Lastly, recommendations regarding how these findings would be relevant in real-time monitoring systems are presented.

6.1. Binary sentiment classification

The findings indicated that Lasso-based models are always superior to RFE and the ANOVA techniques when used with different classifiers in terms of all the evaluation metrics. Table 1 shows a complete analysis of the performance comparison of various feature selection methods and classification algorithms. The Lasso–AdaBoost combination had the highest accuracy (0.8610) and precision (0.9255), and the Lasso–XGBoost combination had the highest $F1$-scores (0.8594), which means that they perform better in terms of classification. Conversely, RFE was found to have much greater computational expenses (283.90 s to select features) with no resultant performance improvements, and ANOVA was found to yield intermediate performances with consistent overtaking by Lasso. It is interesting to note that all the methodologies of feature selection combined with Naive Bayes classifiers had the shortest model training time (0.011–0.023 s), and the longest model training time was on the Random Forest (maximum of 1.574 s). Table 1 shows that the performance of

Table 1
Performance comparison of feature selection methods across multiple classifiers for binary sentiment classification

Fs_Method	Model	Accuracy	Precision	Recall	F1_Score	Fs_Time_Sec	Model_Time_Sec
LASSO	XGB	0.8600	0.9127	0.8120	0.8594	0.037	0.967
LASSO	Ada	0.8610	0.9255	0.8006	0.8585	0.037	1.105
LASSO	RF	0.8438	0.8609	0.8392	0.8499	0.037	1.574
LASSO	NB	0.8308	0.9585	0.7096	0.8155	0.037	0.023
LASSO	DT	0.8057	0.8251	0.8012	0.8130	0.037	0.455
RFE	Ada	0.8197	0.9252	0.7157	0.8071	283.90	0.521
RFE	XGB	0.8190	0.9258	0.7139	0.8061	283.90	0.279
RFE	RF	0.8083	0.9307	0.6873	0.7907	283.90	0.505
ANOVA	XGB	0.8003	0.8962	0.7024	0.7876	0.043	0.376
RFE	NB	0.8063	0.9510	0.6669	0.7840	283.90	0.011

Lasso-based feature selection is superior to all combinations of classifiers, with the highest $F1$-scores and, at the same time, with reasonable computational efficiency with feature selection times below 0.04 s. Conversely, RFE had serious computational bottlenecks, taking a minimum of 280 s to select features, and it was not suitable to be used in real-time applications despite its competitive accuracy results. The results further show that ensemble techniques (XGBoost, AdaBoost, and Random Forest) tend to be more successful than simpler classifiers in combination with useful feature selection methods.

Figure 2 displays the largest $F1$-scores of each of the feature selection methods on binary classification tasks. The figure indicates that Lasso obtained the best $F1$-scores (≈ 0.86), which is much greater than ANOVA (≈ 0.79) and RFE. The enhanced performance of Lasso indicates that the regularization properties of the algorithm are able to recognize discriminative features and reduce overfitting, but the moderate scores of filter-based methods reflect the weakness of the methods in modeling interaction effects of features. These results highlight the importance of feature selection methodology on the performance of models, with Lasso being the best selection approach in binary classification problems.

6.2. Multiclass sentiment classification

Combination rankings were used in three-class tasks using negative, neutral, and positive sentiment classes and were ranked by weighted $F1$-scores. Table 2 is a performance table that provides detailed results of several of the feature selection methods in combination with different machine learning models. ANOVA with XGBoost had the best accuracy (0.7111), precision-weighted (0.7589), recall-weighted (0.7111), and $F1$-score-weighted (0.6993), which demonstrates that it is effective in this task. Table 2 indicates that in multiclass cases, there are major performance disparities between the feature selection techniques. ANOVA-based models were always more successful compared to Lasso and mutual information in all classifiers, with the latter showing much lower performance ($F1$-scores 0.55–0.58) and the worst performance rates (mutual information 0.5316 when using AdaBoost). Computational efficiency ANOVA was the quickest method of feature selection (0.0184 s), whereas mutual information required an enormous amount of time overhead (23.9812 s), which is unrealistic regardless of its moderate performance. There was a significant difference in the model training times, with Naive Bayes being the fastest (0.0141 s) and Random Forest being the slowest (1.0449 s).

The optimal weighted $F1$-scores of each feature selection method in multiclass tasks are overlaid in Figure 3. The visualization reveals that although Lasso had better performance on binary classification ($F1 \approx 0.86$), ANOVA was the best option to use in multiclass tasks ($F1 \approx 0.70$). The significance of choosing task-specific feature selection strategies is highlighted by this performance change. The figure also shows that performance decreases in all methods when the binary classification is changed to multiclass classification and the extent of decrease against methods varies widely. Mutual information and RFE have especially poor performance in multiclass, which suggests that the two approaches do not cope with the complexity of classes.

Figure 2
Comparison of $F1$-scores for different feature selection methods across classifiers

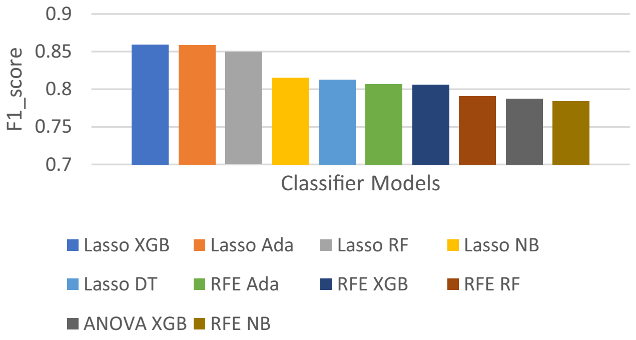


Figure 3
Multiclass $F1$-score performance comparison by feature selection method and classifier

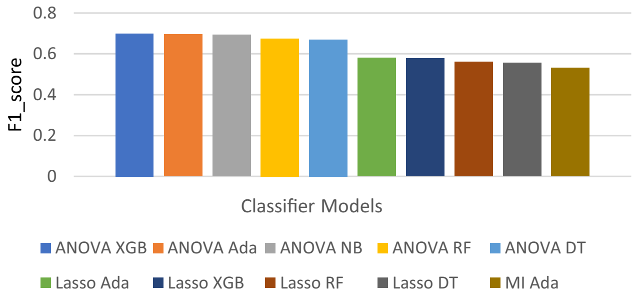


Table 2
Performance comparison of feature-selection methods across different classifiers for multiclass sentiment classification

Fs_Method	Model	Accuracy	Precision_W	Recall	F1_Score	Fs_Time	Model_Time_Sec
ANOVA	XGB	0.7111	0.7589	0.7111	0.6993	0.0184	0.7442
ANOVA	Ada	0.7092	0.7654	0.7092	0.6957	0.0184	0.6152
ANOVA	NB	0.7048	0.7593	0.7048	0.6934	0.0184	0.0141
ANOVA	RF	0.6898	0.7327	0.6898	0.6741	0.0184	0.5427
ANOVA	DT	0.6854	0.7307	0.6854	0.6693	0.0184	0.0454
LASSO	Ada	0.5971	0.6232	0.5971	0.5809	0.1026	0.6149
LASSO	XGB	0.5990	0.6435	0.5990	0.5775	0.1026	0.6200
LASSO	RF	0.5768	0.5991	0.5768	0.5605	0.1026	1.0449
LASSO	DT	0.5730	0.5955	0.5730	0.5561	0.1026	0.1072
MI	Ada	0.5546	0.5209	0.5546	0.5316	23.9812	1.2351

Table 3
Performance Comparison of ANN and CNN models on binary and multiclass classification tasks

Task	Model	Accuracy	Precision	Recall	F1_Score	Precision_W	Recall_W	F1_Score_W
Binary	ANN	0.9051	0.9137	0.9054	0.9095	–	–	–
Binary	CNN	0.5565	0.5576	0.7669	0.6457	–	–	–
Multiclass	ANN	0.8470	–	–	–	0.8516	0.8470	0.8413
Multiclass	CNN	0.5054	–	–	–	0.4697	0.5054	0.4596

6.3. Deep learning models

Comparison of shallow ANNs and simple CNNs that were both trained on the same 5000-feature TF-IDF representations of binary and multicast sentiment data showed a large performance gap between the architectures. On the two classification tasks, Table 3 is a detailed performance analysis of ANN and CNN models. The table indicates that ANN models are much better at the two types of classifications than CNNs. ANNs were more accurate (binary: 0.9051 vs 0.5565; multiclass weighted: 0.8470 vs 0.5054), and their $F1$-scores were higher (binary: 0.9095 vs 0.6457; multiclass weighted: 0.8413 vs 0.4596) and were trained much faster (binary: 8.55s vs 125.98s; multiclass). The classification gaps are even more pronounced in multiclass classification, with ANN weighted $F1$-scores surpassing CNN scores by 83.1 points, highlighting the superior performance of the ANN model for this task.

Figure 4 displays a detailed visualization of the performance of the deep learning models: (a) $F1$-score comparison by model and task (binary vs multiclass) and (b) training time comparison by model and task (binary vs multiclass). The comparison depicted in the visualization shows clearly that ANN is superior in the accuracy metrics and in terms of computational efficiency. As can be seen by the training time comparison, CNNs spend about 15 times less time during training than ANNs and achieve significantly worse performance, which indicates the ineffectiveness of using convolutional operations on sparse representations based on TF-IDF.

The key results of the deep learning analysis are as follows: ANNs surpass all classical models on binary polarity ($F1$ = 0.9095 vs 0.8594 best classical) but require about 8.6 s to train. In multiclass problems, again, ANNs perform better (weighted $F1$ = 0.8413) with a wide margin of difference over classical approaches (ANOVA+XGBoost $F1$ = 0.6993). CNNs on TF-IDF have low performance (< 0.65 $F1$) and are prohibitively expensive to train (> 2 min), which validates the idea that convolutions on sparse text vectors do not work well in this application area.

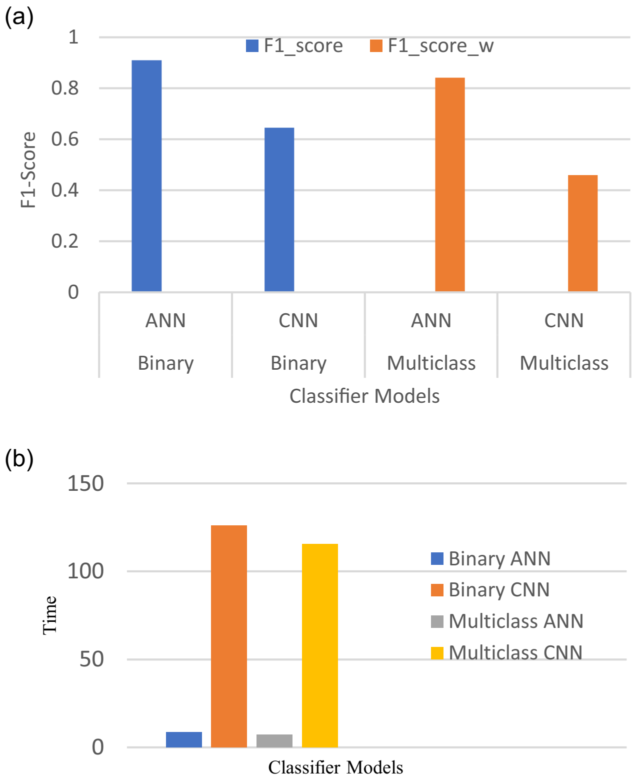
6.4. Discussion

The general analysis of feature-selection techniques and classification models using the SIGHT transcript-comment dataset has several clear implications for practitioners developing real-time text-based monitoring systems.

6.4.1. Embedded feature-selection delivers performance and efficiency

Filter applications and wrapper applications were never as efficient as Lasso on binary polarity tasks, with XGBoost $F1$ of 0.86 in less than 0.04 s of feature-selection time. On the other hand, the same number of features took ~ 284 s to be chosen using an RFE, which is impractical in live streams. The fact that Lasso can zero out irrelevant TF-IDF features to create sparse representations that can be used by downstream classifiers allows Lasso to achieve high accuracy and low overhead. This is a reflection of lightweight rule pruning techniques commonly employed in SCADA intrusion detection systems, in which small rulesets reduce detection time.

Figure 4
Comparison of model performance and training time. (a) $F1$-score comparison by model and task (binary vs multiclass) and (b) training time comparison by model and task (binary vs multiclass)



6.4.2. Ensemble classifiers are top performers

Gradient boosting ensembles and Random Forests achieved better performance than simpler models in both binary and multiclass tasks. XGBoost found an attractive trade-off between prediction and inference performance (< 1 s, 20 features), which makes the algorithm the best choice when quickness is required in an anomaly or sentiment detection problem. Naive Bayes, on the other hand, though being very efficient in terms of feature selection as well as classification speed, had lower $F1$-scores (< 0.82), which means that the conditional independence assumptions were too simple to capture complex text signals. While Bigram features

Table 4
Performance comparison of the state-of-the-art method and the proposed method

Method	Dataset	$F1$-score	Note
Our approach (Lasso+XGBoost)	SIGHT	0.859	This work
FastText [26]	AG News	0.915	4-class news
CNN-static [27]	MR	0.814	Binary sentiment
SVM+TF-IDF [23]	Reuters	0.870	Multiclass
BERT-base [24]	SST-2	0.926	Binary sentiment

Note: Direct comparison is limited due to different datasets and tasks. Our results on educational text are comparable to established baselines on standard benchmarks.

improved recall slightly, they increased dimensionality and lowered the $F1$-score. The independence assumption of Naive Bayes could not be overcome by feature engineering.

6.4.3. Performance contextualization against baselines

It is recognized that the performance indicators ($F1$ = 0.86 for binary, 0.70 for multiclass) are not higher than those found in the NLP literature. However, a greater contribution is not made toward achieving better accuracy; instead, a systematic quantification of the trade-offs between accuracy and latency of classification across various feature selection techniques is provided. The results are compared with the latest achievements in text classification, as presented in Table 4.

Table 4 presents a performance comparison between our proposed Lasso+XGBoost approach and established baselines from the literature. Our method achieves an $F1$-score of 0.859 on the SIGHT educational dataset, which demonstrates competitive performance when contextualized against standard benchmarks. FastText achieves the highest $F1$-score of 0.915 on the AG News dataset, though this represents a 4-class news classification task rather than binary classification. BERT-base shows strong performance with an $F1$-score of 0.926 on SST-2 for binary sentiment analysis, while CNN-static achieves 0.814 on the MR sentiment dataset. The traditional SVM+TF-IDF approach reaches 0.870 on the multiclass Reuters dataset.

It is important to note that direct comparison across these methods is inherently limited due to differences in datasets, task complexity, and domain characteristics. Each baseline was evaluated on different benchmark datasets with varying properties: AG News represents news categorization, MR and SST-2 focus on sentiment analysis, and Reuters involves multiclass document classification. In contrast, our work specifically targets educational text classification, which presents unique challenges including domain-specific vocabulary and pedagogical structures. Despite these differences, our $F1$-score of 0.859 is comparable to established methods, suggesting that the Lasso+XGBoost combination provides effective feature selection and classification capabilities for the educational domain. This performance is particularly noteworthy given that our approach maintains computational efficiency while achieving results competitive with more complex deep learning architectures.

6.4.4. Deep models offer higher accuracy at greater cost

Shallow ANNs trained on TF-IDF achieved the highest binary $F1$ = 0.9095—surpassing classical pipelines by ~ 5 percentage points—but required ~ 8.6 s to train. On multiclass tasks, ANNs reached weighted $F1$ = 0.8413 versus 0.6993 for the best classical models, highlighting superior capacity to model subtle sentiment categories. However, simple CNNs applied directly to TF-IDF vectors performed poorly (binary $F1$ ≈ 0.65) and incurred > 2 min of training. These results underscore that deep architectures demand careful input representations to realize their potential.

6.4.5. Task complexity shapes method selection

Binary polarity can be well-represented with a few TF-IDF features; however, multiclass sentiment cannot be well-represented with such sparse representations. Competitive speed methods such as ANOVA still have competitive speed but are weaker on multiclass F1, and the embedded selection used by Lasso is more flexibly adaptive but still lagging behind ANOVA+XGBoost on 3-class tasks. In the multiclass sentiment classification task, we observed that Lasso (L1 regularization) struggled relative to ANOVA, particularly in capturing the nuanced class distinctions. We hypothesize that this is due to the increased complexity of the decision boundaries in a multiclass setting. Lasso, which excels in identifying a small set of globally relevant features, is well-suited for binary classification tasks where only one decision boundary is needed. However, in multiclass classification, multiple decision boundaries are required, and the informative features for distinguishing one class from another may differ across class pairs. For instance, the features that separate "negative" from "neutral" sentiment may not be the same as those distinguishing "neutral" from "positive."

In contrast, ANOVA, as a filter method, independently evaluates the predictive power of each feature in relation to the target variable. This enables ANOVA to capture a broader range of features that are individually informative for different class distinctions. These features, which may be weakly correlated with one another but are important for distinguishing specific class pairs, can then be effectively leveraged by ensemble classifiers like XGBoost. Ensemble methods can handle these interactions efficiently, making them more suitable for the complexities of multiclass tasks.

Thus, while Lasso's feature selection is highly effective for binary classification due to its focus on globally relevant features, ANOVA's ability to independently evaluate features for each class distinction allows it to better adapt to the complexities of multiclass classification.

6.4.6. Contributions and implications

This paper shows that embedded sparsity and ensemble learning offer optimal trade-offs of accuracy and latency to real-time text monitoring, which is analogous to intrusion detection in SCADA systems. The work also measures the cost of deep models on high-dimensional text features, informing practitioners when they should invest in more modern architecture or classical pipelines that are optimized.

Lasso with ensemble methods provides superior interpretability by reducing the feature space to key terms, offering human-understandable insights into anomalous text. In contrast, the ANN's black-box nature limits its interpretability despite higher accuracy.

6.4.7. Limitations and future directions

TF-IDF is used in experiments; the next step in the work should be to investigate contextual embeddings and lightweight transformer variants to achieve better real-time performance. Also, it is possible to include multimodal cues, which are likely to increase the ability of anomaly detection analogues in critical infrastructure monitoring.

The explicit comparison of our results with the published standards is not viewed as a beneficial one because of the variations in datasets, preprocessing algorithms, and the methods used to perform testing. The studies in Table 4 make use of conventional benchmark datasets (AG News, Movie Reviews, Reuters, Stanford Sentiment Treebank), and our work uses educational comments. Thus, the binary classification $F1$-score of 0.859 cannot be easily compared to, say, the 0.926 of BERT on SST-2 [24], because these are different tasks. It is not claimed that it is superior to these benchmarks, but it is demonstrated that embedded feature selection methods are computationally efficient for text classification problems in real time. Despite the results being comparable to existing benchmarks in NLP, our work offers significant value in terms of its practical application and efficiency. The novelty of this study lies in the integration of feature selection techniques such as Lasso regularization with ensemble methods (XGBoost, AdaBoost) and shallow neural networks, specifically tailored for real-time text anomaly detection. This approach demonstrates the trade-offs between classification accuracy and computational latency in high-dimensional, real-time text classification tasks, which is an important consideration for systems requiring fast inference times. While we do not claim to surpass existing methods in terms of accuracy, our contribution is centered around providing a practical framework for selecting the most suitable feature selection methods and classifiers based on specific computational constraints. This is particularly relevant in resource-constrained environments, such as those found in industrial monitoring or real-time text analysis applications. Furthermore, the use of the SIGHT dataset—an educational text corpus—provides a novel testbed for benchmarking these methods, offering valuable insights into computational efficiency when handling high-dimensional textual data. Future work will explore more advanced architectures, including transformer models, to further enhance performance while maintaining low latency.

7. Conclusion and Future Work

The aim of this study is to investigate the interaction between feature-selection methods and classification models for sentiment analysis, using the SIGHT lecture transcript and comment database. The underlying theory is based on previous research in SCADA intrusion detection systems. Empirical results show that subgradient boosting ensembles over "lasso-based embedded sparsity" provide the best accuracy (binary $F1 \approx 0.86$, multiclass $F1 \approx 0.70$) and computational efficiency (< 1 s end-to-end latency) trade-offs feasible for real-time monitoring. Ordinary Fully Connected (FC) networks already provide superior performance (binary $F1 \approx 0.91$, multiclass $F1 \approx 0.84$) at the cost of higher training overhead (~ 8 s). On the other hand, the classical and simple ANN baselines outperform CNNs applied to TF-IDF features in general.

This work demonstrates the promise of simple feature selection, inspired by rule pruning in critical infrastructure intrusion detection systems tied with black-box methods for text-based anomaly or sentiment detection. Our results also indicate that deep architectures offer only a few improvements over sparse representations without domain-specific embedding.

Future work will seek to better trade off between semantic granularity and computational tractability in text-based monitoring systems. We will first explore transformer architectures that are lightweight in design to both encode contextual relationships and offer sub-second inference times. Second, we will be introducing methods based on incremental learning to develop models that can adapt continuously in streaming data without suffering catastrophic forgetting. This work extends our previous framework by including multimodal sensor inputs to improve the representation of multivariate anomaly detection functionalities present in industrial SCADA systems and address the challenges faced due to temporal alignment and feature fusion across heterogeneous data spaces. We chose TextBlob for its speed and deterministic output, prioritizing real-time, lightweight feature engineering. Future work will explore more efficient, advanced sentiment analysis methods. Future work will explore the use of lightweight transformer models like DistilBERT and TinyBERT to improve inference speed while maintaining accuracy. Additionally, knowledge distillation techniques will be employed to compress large models into more efficient versions, balancing computational complexity and latency.

Acknowledgment

The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA, for funding this research work through the project number NBU-FFR-2026-2119-01.

Funding Support

This work is supported by the Deanship of Scientific Research at Northern Border University, Arar, Kingdom of Saudi Arabia (Grant number: NBU-FFR-2026-2119-01).

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in GitHub at https://github.com/rosewang2008/sight.

Author Contribution Statement

Ayoub Alsarhan: Conceptualization, Resources, Writing – original draft, Visualization. **Saja Jaradat:** Conceptualization, Resources, Writing – original draft, Visualization. **Suhaila Abuowaida:** Validation, Writing – review & editing, Funding acquisition. **Sami Aziz Alshammari:** Methodology, Investigation, Data curation, Supervision. **Nayef Hmoud Alshammari:** Software, Project administration. **Khalid Hamad Alnafisah:** Software, Formal analysis, Project administration.

References

[1] Salman, H. A., Kalakech, A., & Steiti, A. (2024). Random forest algorithm overview. *Babylonian Journal of Machine Learning*, *2024*, 69–79. https://doi.org/10.58496/BJML/2024/007

[2] Nalluri, M., Pentela, M., & Eluri, N. R. (2020). A scalable tree boosting system: XG Boost. *International Journal of Research Studies in Science, Engineering and Technology*, *7*(12), 36–51. https://doi.org/10.22259/2349-476X.0712005

[3] Imani, M., Beikmohammadi, A., & Arabnia, H. R. (2025). Comprehensive analysis of Random Forest and XGBoost performance with SMOTE, ADASYN, and GNUS under varying imbalance levels. *Technologies*, *13*(3), 88. https://doi.org/10.3390/technologies13030088

[4] Rajesh, L., & Satyanarayana, P. (2022). Evaluation of machine learning algorithms for detection of malicious traffic in SCADA network. *Journal of Electrical Engineering & Technology*, *17*(2), 913–928.https://doi.org/10.1007/s42835-021-00931-1

[5] Alimi, O. A., Ouahada, K., Abu-Mahfouz, A. M., Rimer, S., & Alimi, K. O. A. (2021). A review of research works on supervised learning algorithms for SCADA intrusion detection and classification. *Sustainability*, *13*(17), 9597. https://doi.org/10.3390/su13179597

[6] Karthigha, M., Latha, L., & Madhumathi, R. (2024). Feature selection and classification models of intrusion detection systems—A review on industrial critical infrastructure perspective. In A. Kumari. K & A. Sharma (Eds.), *Cyber physical systems-Advances and Applications* (pp. 169–188). Bentham Books. https://doi.org/10.2174/9789815223286124010010

[7] Nogales, R. E., & Benalcázar, M. E. (2023). Analysis and evaluation of feature selection and feature extraction methods. *International Journal of Computational Intelligence Systems*, *16*(1), 153. https://doi.org/10.1007/s44196-023-00319-1

[8] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

[9] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827

[10] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

[11] Bostanian, Z., Boostani, R., Sabeti, M., & Mohammadi, M. (2022). ORBoost: An orthogonal AdaBoost. *Intelligent Data Analysis: An International Journal*, *26*(3), 805–818. https://doi.org/10.3233/IDA-205705

[12] Wickramasinghe, I., & Kalutarage, H. (2021). Naive Bayes: Applications, variations and vulnerabilities: A review of literature with code snippets for implementation. *Soft Computing*, *25*(3), 2277–2293. https://doi.org/10.1007/s00500-020-05297-6

[13] Gan, S., Shao, S., Chen, L., Yu, L., & Jiang, L. (2021). Adapting hidden Naive Bayes for text classification. *Mathematics*, *9*(19), 2378. https://doi.org/10.3390/math9192378

[14] Chase, R. J., Harrison, D. R., Lackmann, G. M., & McGovern, A. (2023). A machine learning tutorial for operational meteorology. Part II: Neural networks and deep learning. *Weather and Forecasting*, *38*(8), 1271–1293. https://doi.org/10.1175/WAF-D-22-0187.1

[15] de Diego, I. M., Redondo, A. R., Fernández, R. R., Navarro, J., & Moguerza, J. M. (2022). General performance score for classification problems. *Applied Intelligence*, *52*(10), 12049–12063. https://doi.org/10.1007/s10489-021-03041-7

[16] Erickson, B. J., & Kitamura, F. (2021). Magician's corner: 9. performance metrics for machine learning models. *Radiology: Artificial Intelligence*, *3*(3), e200126. https://doi.org/10.1148/ryai.2021200126

[17] Ahakonye, L. A. C., Nwakanma, C. I., Lee, J.-M., & Kim, D.-S. (2023). Agnostic CH-DT technique for SCADA network high-dimensional data-aware intrusion detection system. *IEEE Internet of Things Journal*, *10*(12), 10344–10356 . https://doi.org/10.1109/JIOT.2023.3237797

[18] Ahakonye, L. A. C., Nwakanma, C. I., Lee, J.-M., & Kim, D.-S. (2023). SCADA intrusion detection scheme exploiting the fusion of modified decision tree and Chi-square feature selection. *Internet of Things*, *21*, 100676. https://doi.org/10.1016/j.iot.2022.100676

[19] Alimi, O. A., Ouahada, K., Abu-Mahfouz, A. M., Rimer, S., & Alimi, K. O. A. (2022). Supervised learning based intrusion detection for SCADA systems. In *2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development*, 1–5. https://doi.org/10.1109/NIGERCON54645.2022.9803101

[20] Wali, A., & Alshehry, F. (2024). A Survey of security challenges in cloud-based SCADA systems. *Computers*, *13*(4), 97. https://doi.org/10.3390/computers13040097

[21] Gumaei, A., Hassan, M. M., Huda, S., Hassan, R., Camacho, D., del, Ser., . . . , & J. (2020). A robust cyberattack detection approach using optimal features of SCADA power systems in smart grids. *Applied Soft Computing*, *96*, 106658. https://doi.org/10.1016/j.asoc.2020.106658

[22] Upadhyay, D., Manero, J., Zaman, M., & Sampalli, S. (2021). Gradient boosting feature selection with machine learning classifiers for intrusion detection on power grids. *IEEE Transactions on Network and Service Management*, *18*(1), 1104–1116. https://doi.org/10.1109/TNSM.2020.3032618

[23] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, *10*(4), 150. https://doi.org/10.3390/info10040150

[24] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning-based text classification: A comprehensive review. *ACM Computing Surveys*, *54*(3), 62. https://doi.org/10.1145/3439726

[25] Qader, W. A., Ameen, M. M., & Ahmed, B. I. (2019). An overview of bag of words: Importance, implementation, applications, and challenges. In *Proceedings of the 2019 International Engineering Conference*, 200–204. https://doi.org/10.1109/IEC47844.2019.8950616

[26] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 427–431.

[27] Wang, R., Li, Z., Cao, J., Chen, T., & Wang, L. (2019). Convolutional recurrent neural networks for text classification. In *2019 International Joint Conference on Neural Networks*, 1–6. https://doi.org/10.1109/IJCNN.2019.8852406

[28] Horak, M., Chandrasekaran, S., & Tobar, G. (2022). NLP based anomaly detection for categorical time series. In *2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science*, 27–34. https://doi.org/10.1109/IRI54793.2022.00019

[29] Yadav, G., Deepak, Sharma., H, Sharma., D, & Rai, A. K. (2024). A study on deep learning based anomaly detection technique in natural language processing textual data. In *2024 2nd International Conference on Advances in Computation, Communication and Information Technology*, 869–873. https://doi.org/10.1109/ICAICCIT64383.2024.10912228

[30] Nguyen, M. T. A., Tong, V., Souihi, S. B., & Souihi, S. (2025). Zero Trust: Deep learning and NLP for HTTP anomaly detection in IDS. *IEEE Journal on Selected Areas in Communications*, *43*(6), 2215–2229. https://doi.org/10.1109/JSAC.2025.3560040

[31] Mohammed, S. Y., Aljanabi, M., Mahmood, A. M., & AVCI, I. (2023). Revolutionizing language learning: How AI bots enhance language acquisition. *Babylonian Journal of Artificial Intelligence*, *2023*, 55–63. https://doi.org/10.58496/BJAI/2023/009

[32] Ali, A., AlShuaibi, A., & Arshad, M. W. (2025). An Integrated federated learning framework with optimization for industrial IoT intrusion detection. *SHIFRA*, *2025*, 110–117. https://doi.org/10.70470/SHIFRA/2025/007

[33] Nyangaresi, V. O. (2023). Role of artificial intelligence (AI) in improving educational quality for students and faculty. *Babylonian Journal of Artificial Intelligence*, *2023*, 83–90. https://doi.org/10.58496/BJAI/2023/012

[34] Dutta, P. K., Sekiwu, D., Unogwu, O. J., Catherine, A. T., & Helal, A. H. (2023). Metaverse revolution in higher education: The rise of metaversities and immersive learning environments. *Babylonian Journal of Internet of Things*, *2023*, 59–68. https://doi.org/10.58496/BJIoT/2023/008

[35] Otoom, S. (2025). Risk auditing for Digital Twins in cyber physical systems: A systematic review. *Journal of Cyber Security and Risk Auditing*, *2025*(1), 22–35. https://doi.org/10.63180/jcsra.thestap.2025.1.3

[36] Albinhamad, H., Alotibi, A., Alagnam, A., Almaiah, M., & Salloum, S. (2025). Vehicular Ad-hoc Networks (VANETs): A key enabler for smart transportation systems and challenges. *Jordanian Journal of Informatics and Computing*, *2025*(1), 4–15. https://doi.org/10.63180/jjic.thestap.2025.1.2

[37] Ali, A. (2024). Adaptive and context-aware authentication framework using edge AI and blockchain in future vehicular networks. *STAP Journal of Security Risk Management*, *2024*(1), 45–56. https://doi.org/10.63180/jsrm.thestap.2024.1.3

[38] Alalousi, A., Razif, R., AbuAlhaj, M., Anbar, M., & Nizam, S. (2016). A preliminary performance evaluation of K-means, KNN and EM unsupervised machine learning methods for network flow classification. *International Journal of Electrical and Computer Engineering*, *6*(2), 778–784. http://doi.org/10.11591/ijece.v6i2.pp778-784

[39] Abualhaj, M. M., Abu-Shareha, A. A., Hiari, M. O., Alrabanah, Y., Al-Zyoud, M., & Alsharaiah, M. A. (2022). A paradigm for DoS attack disclosure using machine learning techniques. *International Journal of Advanced Computer Science and Applications*, *13*(3), 192–200. https://doi.org/10.14569/IJACSA.2022.0130325

[40] Wang, R., Wirawarn, P., Goodman, N., & Demszky, D. (2023). Sight: A large annotated dataset on student insights gathered from higher education transcripts. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, 315–351. https://doi.org/10.18653/v1/2023.bea-1.27

How to Cite: Alsarhan, A., Jaradat, S., Abuowaida, S., Alshammari, S. A., Alshammari, N. H., & Alnafisah, K. H. (2026). Embedded Sparsity and Ensemble Learning for Real-Time Textual Anomaly Detection: A SCADA-Inspired Benchmark on Lecture Transcript Data. *Journal of Computational and Cognitive Engineering*. https://doi.org/10.47852/bonviewJCCE62027629