Towards Mobile-deployable Incident Report Classification: A Statistically Validated Benchmark of Classical Machine Learning Algorithms

Aaron Paul M. Dela Rosa*, Donna Q. Fernando, Rosalyn P. Reyes, and Evelyn C. Samson

College of Information and Communications Technology, Bulacan State University, City of Malolos, Philippines
Email: aaronpaul.delarosa@bulsu.edu.ph (A.P.M.D.R.); donna.fernando@bulsu.edu.ph (D.Q.F.); rosalyn.reyes@bulsu.edu.ph (R.P.R.); evelyn.samson@bulsu.edu.ph (E.C.S.)
*Corresponding author

Manuscript received October 20, 2025; revised January 30, 2026; accepted May 15, 2026; published September 11, 2026

***Abstract*—Efficient and accurate classification of incident reports is essential for improving decision-making in local governance systems. In the Philippine context, a barangay, the smallest administrative unit, serves as the frontline in handling community-level incidents. While many studies apply machine learning to text classification, limited attention has been given to computational efficiency and deployment feasibility in resource-constrained environments. This study presents a comparative evaluation of classical machine learning algorithms for classifying real-world barangay incident reports using Term Frequency-Inverse Document Frequency (TF-IDF) features. A lightweight embedding-based baseline (FastText) was also included to assess whether semantic representations improve performance under deployment constraints. The evaluated models span probabilistic, linear, margin-based, online, and ensemble learning paradigms, enabling analysis of predictive performance and computational trade-offs. Linear Support Vector Machine (SVM) achieved the highest performance, with a mean accuracy of 0.6932 ± 0.0408 and a weighted F1-score of 0.6890 ± 0.0425. FastText achieved competitive results (accuracy = 0.6633 ± 0.0388) but did not outperform SVM. Wilcoxon signed-rank testing confirmed that SVM significantly outperformed all models except Stochastic Gradient Descent (SGD) ($p < 0.05$). Results indicate that linear classifiers provide the best balance between predictive performance and computational efficiency, making them suitable for mobile and resource-constrained deployment.**

***Keywords*—incident report classification, Term Frequency-Inverse Document Frequency (TF-IDF), FastText, machine learning benchmarking, resource-constrained deployment, statistical validation**

I. Introduction

Incident reports are essential for documenting events in communities, institutions, and organizations, serving as critical tools for accountability, risk management, and policy formulation [1]. Traditionally, the processing and classification of such reports have been performed manually, making them time-consuming, labor-intensive, and prone to human error [2]. With the increasing volume of textual incident data, there is a growing need for automated approaches that can efficiently and accurately classify reports. Advances in Machine Learning (ML) and Natural Language Processing (NLP) provide viable solutions to address these challenges [3].

Text classification, a core task in NLP, involves assigning predefined labels to textual data and has been widely applied in domains such as sentiment analysis, spam detection, and document categorization [4]. While deep learning models, particularly transformer-based architectures, have achieved state-of-the-art performance in many text processing tasks, classical machine learning approaches remain competitive in domain-specific applications, especially when data and computational resources are limited [5]. These methods are generally more interpretable, require fewer computational resources, and perform effectively when combined with feature extraction techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) [6].

In the context of local governance, particularly at the barangay level, deployment constraints play a critical role in model selection. Systems must operate efficiently in environments with limited computational capacity, such as mobile devices or low-resource servers, while still providing accurate and timely classification. Although many existing studies focus on improving classification accuracy using complex models, they often overlook practical considerations such as computational efficiency, interpretability, and deployment feasibility in real-world settings [7, 8].

This study addresses this gap by evaluating classical machine learning algorithms for incident report classification under deployment-oriented constraints. Specifically, the study compares Logistic Regression, Naïve Bayes, Support Vector Machine, Random Forest, Stochastic Gradient Descent (SGD), Passive Aggressive, and Gradient Boosting classifiers. In addition, a lightweight embedding-based baseline (FastText) is included to assess whether semantic representations improve performance under the same conditions.

The study aims to: (1) evaluate and compare the predictive performance of classical machine learning models for incident report classification; (2) validate performance differences using statistical testing; and (3) analyze computational efficiency and deployment feasibility in resource-constrained environments.

Unlike prior studies that focus primarily on accuracy or single-model evaluation, this work provides a systematic comparison across multiple learning paradigms and incorporates both statistical validation and deployment-oriented analysis. By integrating performance evaluation with practical deployment considerations, the study offers insights into selecting models that balance accuracy and efficiency for mobile and resource-constrained governance systems.

II. Literature Review

A. Introduction to Incident Report Analysis

Incident reporting systems play a critical role in documenting, tracking, and resolving community-level issues, particularly in local governance contexts such as the Philippine barangay system. As the lowest administrative unit, barangays serve as the frontline in handling diverse incidents ranging from minor disputes to public safety concerns [9]. However, manual processing of incident reports presents challenges, including inconsistent classification, delayed response, and limited support for data-driven decision-making.

Machine learning-based approaches offer a scalable solution for automating the classification of incident reports. While prior work has explored such approaches in domains such as healthcare, workplace safety, and information security [10–12], limited studies have focused on small-scale, multilingual, and resource-constrained governance settings. Existing systems often assume access to large datasets and computational resources, which may not be feasible in barangay environments. This highlights the need for lightweight and interpretable models that can operate effectively under real-world constraints [13].

B. Text Representation in NLP for Local Governance Data

Text representation is a critical step in transforming unstructured incident reports into machine-processable formats. In barangay contexts, reports are often written in Filipino/Tagalog, English, or code-switched language, introducing linguistic variability that complicates preprocessing and modeling.

Traditional feature representation techniques, such as Bag-of-Words (BoW) and Term Frequency (TF), have been widely used but are limited in their ability to capture term importance across documents. TF-IDF addresses this limitation by weighting terms based on their relative importance, making it effective for sparse textual data and low-resource environments [14]. Prior studies have shown that TF-IDF remains competitive in governance-related text classification tasks while maintaining low computational cost [15, 16].

In contrast, embedding-based approaches such as Word2Vec, GloVe, and transformer-based models provide richer semantic representations but typically require larger datasets and higher computational resources [17, 18]. These requirements pose challenges in deployment scenarios such as barangay systems, where computational infrastructure is limited.

While TF-IDF is effective for capturing term importance, it does not encode contextual semantics or word order. To address this limitation, recent studies have explored lightweight embedding-based approaches. However, there remains limited empirical evidence comparing these methods under constrained, multilingual governance settings. This gap motivates the inclusion of a lightweight embedding-based baseline (FastText) in the present study.

C. Classical Machine Learning Algorithms for Text Classification

Classical machine learning algorithms remain widely used for text classification due to their balance between efficiency, interpretability, and performance in high-dimensional feature spaces. Models such as Logistic Regression, Naïve Bayes, Support Vector Machines (SVM), Random Forest, and Gradient Boosting have demonstrated effectiveness in handling sparse representations such as TF-IDF [19–23].

Prior studies in the Philippine context have applied these methods to tasks such as judicial decision prediction, tweet classification, and customer feedback analysis, highlighting their robustness in multilingual and noisy text environments [20–22]. Lightweight algorithms such as Stochastic Gradient Descent (SGD) and Passive Aggressive classifiers further support real-time and incremental learning scenarios, making them suitable for streaming data in governance systems.

While ensemble methods such as Random Forest and Gradient Boosting can improve predictive performance, they often introduce higher computational cost, which may limit deployment in resource-constrained environments. In contrast, linear and margin-based classifiers offer efficient training and inference, making them attractive for mobile-based applications.

Despite the growing use of deep learning approaches, many existing studies rely on large-scale datasets and do not evaluate computational efficiency or deployment feasibility. Furthermore, limited work provides statistically validated comparisons across multiple classical algorithms in governance-specific contexts. This highlights the need for systematic benchmarking that considers both performance and practical deployment constraints.

D. Evaluation Metrics for Governance-oriented Text Classification

Evaluating text classification models in governance contexts requires metrics that capture both overall performance and class-level behavior. While accuracy provides a general measure, it may be misleading in imbalanced datasets commonly observed in incident reporting [24].

Precision, recall, and F1-score offer more informative evaluation by capturing trade-offs between false positives and false negatives, which are critical in governance scenarios where misclassification may affect prioritization and response [25, 26]. Confusion matrices further provide insights into class-specific performance and systematic misclassification patterns, enabling targeted improvements in data collection and model design [27].

To address class imbalance, macro and weighted averaging are commonly employed to ensure fair evaluation across both frequent and rare categories [28]. In addition to these metrics, recent studies emphasize the importance of statistical validation to ensure that observed performance differences are meaningful rather than due to random variation.

However, most prior work focuses primarily on predictive accuracy without considering computational efficiency or deployment constraints. This study extends existing approaches by incorporating statistical testing and deployment-oriented analysis to provide a more comprehensive evaluation framework.

Despite the extensive application of machine learning in text classification, prior studies exhibit several limitations when applied to local governance contexts. Most existing

work focuses on maximizing predictive accuracy using either single-model approaches or computationally intensive deep learning architectures, often evaluated on large-scale and well-structured datasets. However, these approaches may not be directly applicable to barangay-level systems, which are characterized by small datasets, multilingual and code-switched text, and limited computational resources.

Furthermore, existing studies rarely provide comprehensive comparisons across multiple classical machine learning algorithms under consistent experimental settings, nor do they incorporate statistical validation to assess the significance of observed performance differences. Deployment considerations, such as computational efficiency, model size, and suitability for mobile or edge environments, are also often overlooked.

To address these gaps, this study provides a systematic benchmarking of classical machine learning models alongside a lightweight embedding-based baseline (FastText), incorporating statistical validation and deployment-oriented analysis. This enables a more holistic evaluation of performance-efficiency trade-offs, which is essential for real-world implementation in resource-constrained governance systems.

III. Materials and Methods

A. Dataset Preparation

The dataset was gathered from the City of Malolos, Bulacan, Philippines, across five barangays: Longos, Pinagbakahan, Look 1st, Bulihan, and Mojon. A barangay refers to the smallest administrative unit in the Philippines and serves as the primary level of local governance. A total of 623 incident reports were collected, all manually labeled into 15 categories, including crime-related cases, sanitation complaints, domestic conflicts, curfew violations, and other governance-related incidents. This variety ensures that the dataset captured the ordinary problems experienced at the barangay level. The dataset exhibits significant class imbalance, with some categories containing fewer than 10 samples. While this reflects real-world reporting distributions, it introduces challenges in model generalization and minority class prediction. No explicit class imbalance mitigation techniques, e.g., Synthetic Minority Oversampling Technique (SMOTE), oversampling, or class weighting, were applied, as the study aims to evaluate model performance under real-world data distributions.

B. Data Preprocessing and Feature Representation

Text preprocessing was conducted in order to cleanse and normalize the incident reports. It involved converting all text to lowercase, removing punctuation, stopwords, and non-alphanumeric characters, and then tokenizing words from the sentences. Normalization was finally conducted through stemming and lemmatization in order to reduce words to their root forms. The preprocessing pipeline can be summarized as follows:

- Translation of reports from Filipino/Tagalog to English
- Conversion to lowercase
- Removal of punctuation, stopwords, and non-alphanumeric characters
- Tokenization
- Stemming and lemmatization
- TF-IDF vectorization

Because the original incident reports were primarily written in Filipino/Tagalog and often contained code-switched expressions, all entries were translated into English prior to preprocessing to ensure compatibility with the NLP tools and consistency in feature representation. Initial translation was performed using an artificial intelligence-assisted tool (Gemini), followed by manual review and correction by the researchers to preserve contextual meaning, barangay-specific terminology, and class semantics. After the researchers reviewed the translation, the translated reports were validated by the *Sentro ng Wika at Kultura*, a center at the researchers' university specializing in translating English to Filipino/Tagalog and vice versa, to ensure that all translations used were correct and accurate. Particular attention was given to incident-specific expressions such as disputes, threats, and fraud-related complaints, where direct translation may alter interpretation. TF-IDF vectorization was applied after translation to maintain a unified vocabulary space across all reports. While this process improved consistency for model training, translation may still introduce semantic distortion or reduce linguistic nuance, particularly in informal or context-dependent narratives. This limitation is acknowledged, and future work will evaluate models directly on multilingual and code-switched text. While translation ensured consistency in preprocessing and compatibility with NLP tools, it may introduce semantic distortion or reduce linguistic nuance, particularly in code-switched expressions. This limitation is acknowledged, and future work will evaluate models directly on multilingual and code-switched text to better reflect real-world barangay reporting conditions. While TF-IDF effectively captures the discriminative importance of terms, it does not encode semantic relationships, word order, or contextual meaning. As a result, it may struggle with polysemy and semantically similar expressions across categories. This limitation motivates the inclusion of embedding-based approaches such as FastText in the experimental evaluation.

C. Classification Algorithms

Seven classical machine learning algorithms were selected to reflect a balanced spectrum of baseline, robust, lightweight, and ensemble-based classifiers commonly used in text classification. The selection was guided by three criteria: (1) proven effectiveness on high-dimensional sparse text features, (2) computational efficiency suitable for mobile or low-resource deployment, and (3) representativeness of distinct learning paradigms. This mixture enables meaningful comparisons between tree-based and non-tree-based models, as well as between batch and online learning approaches. Table 1 presents the characteristics and advantages for text classification of the models selected for comparison.

The selection of algorithms in this study was guided by their established effectiveness in text classification and their suitability for resource-constrained environments such as barangay-level government systems. Logistic Regression and Naïve Bayes were included as baseline models due to their simplicity, computational efficiency, and interpretability, making them well-suited for mobile-deployable applications.

Naïve Bayes, in particular, has been widely used as a baseline in text classification due to its strong performance in handling sparse, high-dimensional data representations [21, 29].

Table 1. Selected classical machine learning algorithms for text classification of incident reports

Algorithm	Key Characteristics	Advantages for Text Classification	Supporting References
Logistic Regression (LR)	Linear model that estimates class probabilities using a logistic function.	Efficient, interpretable, and performs well with TF-IDF features.	[20]
Naïve Bayes (NB)	Probabilistic model assuming conditional independence among features.	Fast, handles sparse data well, strong recall performance.	[21, 29]
Support Vector Machine (SVM)	Maximizes margins between classes in high-dimensional space.	Robust, effective in text classification tasks, high accuracy.	[21, 30, 31]
Random Forest (RF)	Ensemble of decision trees using bagging.	Handles noisy and imbalanced data, interpretable feature importance.	[32, 33]
Stochastic Gradient Descent (SGD) Classifier	Linear model optimized with gradient descent on large-scale data.	Scalable, efficient with high-dimensional sparse data.	[34, 35]
Passive Aggressive (PA) Classifier	An online learning algorithm that updates only on misclassified samples.	Lightweight, suited for streaming or incremental data.	[36]
Gradient Boosting (GB)	An ensemble method that builds sequential models to correct prior errors.	High accuracy, effective for complex decision boundaries.	[37, 38]

Support Vector Machine (SVM) and Random Forest (RF) were selected for their robust classification capabilities. SVM is widely regarded as one of the most effective models for text classification, particularly due to its ability to construct optimal separating hyperplanes in high-dimensional feature spaces [30, 31]. Random Forest, as an ensemble of decision trees, can capture nonlinear relationships and handle noisy data, making it suitable for heterogeneous incident reports [32, 33].

Stochastic Gradient Descent (SGD) and Passive Aggressive (PA) classifiers were included due to their scalability and efficiency in handling large, high-dimensional datasets. SGD enables efficient optimization for linear models using iterative updates, making it particularly suitable for sparse textual data [34, 35]. The Passive Aggressive algorithm, as an online learning method, is well-suited for real-time and streaming classification scenarios, where models must adapt incrementally to new data [36].

Gradient Boosting (GB) was incorporated to evaluate the performance of boosting-based ensemble methods. By iteratively correcting errors from previous models, Gradient Boosting can capture complex feature interactions and improve predictive accuracy, albeit at a higher computational cost [37, 38].

Collectively, these models represent a diverse set of approaches, including baseline probabilistic methods, linear margin-based classifiers, online learning algorithms, and ensemble techniques. This selection enables a comprehensive comparison across dimensions of predictive performance, computational efficiency, and deployment feasibility for mobile-deployable incident report classification systems.

From a computational perspective, linear and margin-based classifiers exhibit near-linear time complexity with respect to the number of features, making them well-suited for high-dimensional sparse text representations. In contrast, tree-based ensemble methods incur a higher computational cost due to multiple tree constructions and iterative boosting processes. This complexity analysis further supports the deployment-oriented recommendations of this study.

D. FastText Baseline Model

To complement the evaluation of classical machine learning algorithms and address limitations of TF-IDF-based representations, a lightweight embedding-based baseline using FastText was incorporated. FastText is an efficient word embedding model that represents text by averaging word vectors and incorporating subword information, allowing it to capture semantic relationships and handle morphological variations effectively.

FastText is particularly suitable for low-resource and multilingual environments due to its computational efficiency and ability to generalize across word variations. Unlike TF-IDF, which treats words as independent features, FastText captures contextual similarity through distributed representations, potentially improving classification performance in semantically overlapping categories.

In this study, FastText was configured with an embedding dimension of 100, balancing representational capacity and computational efficiency. This baseline enables a more comprehensive evaluation by comparing sparse-feature models with lightweight semantic embedding approaches.

E. Model Training and Experimental Setup

Model evaluation was conducted using repeated stratified 5 × 5 cross-validation (25 runs), eliminating the need for a fixed train–test split and providing more robust performance estimates. Due to the comparatively small dataset, repeated stratified 5 × 5 cross-validation (25 runs) was employed to ensure robust and stable performance estimation while preserving class distribution across folds. All experiments were conducted in a standard cloud-based environment (Google Colab) to ensure reproducibility and accessibility. While this does not directly replicate mobile hardware, it provides a consistent baseline for comparing computational efficiency across models.

Model performance during the training phase was monitored to ensure convergence and detect signs of overfitting. However, this study emphasizes test-set performance, as generalization capability is more relevant for real-world deployment. Given the moderate dataset size and the deployment-oriented objective, reporting test performance provides a more realistic assessment of operational effectiveness.

A clear distinction emerges between tree-based ensemble models and non-tree-based classifiers. Random Forests and Gradient Boosting exhibit stable, robust performance due to their ability to model nonlinear feature interactions and reduce variance. However, this robustness comes at the cost

of significantly higher computational time and memory usage. In contrast, non-tree-based models such as SVM, SGD, and Passive Aggressive achieve comparable accuracy with substantially lower latency, making them more suitable for real-time and mobile applications. From a practical perspective, tree-based models are better suited for offline batch analytics, while margin-based and linear models are preferable for deployment-constrained environments.

No explicit class imbalance mitigation techniques, e.g., oversampling, undersampling, or class weighting, were applied. Instead, macro and weighted evaluation metrics were used to account for imbalance during performance assessment. This approach reflects real-world deployment conditions but may affect performance on minority classes.

F. Evaluation Metrics

To analyze performance, several metrics were used. Accuracy was calculated overall as the percentage of correctly classified reports. Precision, recall, and F1-scores were further calculated per class and reported as macro and weighted averages, allowing for unbiased measurement across both common and uncommon categories. Confusion matrices were also computed to provide a clear indication of misclassifications across the 15 categories and which incident types were most error-prone. Aside from predictive accuracy, computational efficiency was also considered. Both training and inference time were considered in the runtime of each model, a key factor in the feasibility of mobile-deployable systems, where limited processing capability will be a key constraint [13].

G. Mobile-deployment Considerations

Because the initial goal of this work is the exploration of data that can be incidentally classified by methods suitable for mobile use, model testing went beyond predictive capability. Criteria governed the selection of predictive accuracy, computational time, and the model's weight to ensure compatibility with resource-limited devices. They align with the growing focus on digital government and the practical demand at the barangay level for scalable, real-time tools. The ultimate test, then, was models that balanced capability and efficiency, and these were the viable contenders that could be put into the barangay information system for use.

IV. Results and Discussion

A. Data Preprocessing and Feature Representation

The dataset used in this study comprises 623 incident reports from five barangays in the City of Malolos, Bulacan, Philippines. A barangay refers to the smallest administrative unit in the Philippines and serves as the primary level of local governance. The reports were categorized into 15 incident types. Table 2 presents the different categories and their frequencies in the dataset. Several random examples were also included.

Property Damage/Incident has the highest number of reports (95), followed by Accident (81), Theft (66), and Debt/Unpaid Wages (64). These top the dataset and indicate that property issues and vehicular or traffic accidents are widespread in the reporting area. On the other hand, like Drug Addiction (2) and Others (7) have significantly fewer reports, implying low rates of incidents or underreporting.

Table 2. Dataset summary by category, frequency, and sample reports

Category	Frequency	Sample Reports
Theft	66	Theft of Property; Cell Phone Theft; Robbery in Vape Shop
Reports/Agreement	27	Internet Application Report; Business Partnership Agreement in Store; Road Accident Settlement Agreement
Accident	81	Road Accident; Motorcycle Collision with Vehicle; Vehicle Crash into Broken Utility Pole
Debt/Unpaid Wages Report	64	Failure to Pay Debt; Unpaid Rent; Unpaid Electricity Bill
Defamation Complaint	20	Defamation in Facebook Post; Spreading False Information; Social Media Post
Assault/Harassment	53	Physical Assault; Harassment; Machete Attack on Arm
Property Damage/Incident	95	Vehicle Damage; Illegal Construction; Trespassing in Restaurant
Animal Incident	13	Animal Bite Report; Neighbor's Noisy Dog; Missing Ducks
Verbal Abuse and Threats	63	Threatening with a Knife; Profanity; Verbal Argument
Alarm and Scandal	33	Drunk Driving; Unauthorized Video Recording; Loud Music at Night
Lost Items	41	Lost Phone; Missing Identification; Lost Wallet
Scam/Fraud	48	Online Scam; Bounced Check; Hacked Facebook Account
Drugs Addiction	2	Illegal Drug Usage; Health and Safety Concern (Sensitive Case)
Missing Person	10	Missing Child; Missing Sibling; Failure to Return Home
Others	7	Burning Trash; Internet Disconnection; Unclaimed Parcel

Providing sample reports for each category provides qualitative evidence of the lexical and semantic variability within each class. For instance, the Theft category contains general and specific terms such as "Theft of Property", "Cell Phone Theft", and "Robbery in Vape Shop", illustrating lexical variability that the model needs to learn and generalize from. In the same way, Accident reports demonstrate variability, such as "Motorcycle Collision with Vehicle" and "Road Accident", which illustrate the domain-specific terms present.

Dataset composition is critical in developing successful text classifiers. Unequal distribution of category sizes introduces class imbalance, which can bias models toward majority classes and degrade performance on minority categories. To address this, weighted evaluation metrics, such as the weighted F1-score, are emphasized alongside macro metrics to ensure balanced assessment across all classes [39, 40]. In addition, semantic disparities across categories such as Scam/Fraud and Verbal Abuse and Threats can make classification difficult, underscoring the importance of robust preprocessing and feature extraction.

Following recent works, TF-IDF vectorization was used to transform textual data into format-matched numerical representations, enabling classical machine learning algorithms to learn from term significance rather than raw word frequencies [1, 6]. This step is vital, as discriminative

terms across categories (e.g., "robbery", "agreement", "accident") are retained while the effects of frequently occurring, non-informative words are diminished.

Most of the incident reports were originally written in Filipino/Tagalog or code-switched language and were translated into English prior to preprocessing. While this ensures consistency in feature representation, translation may introduce semantic distortion or reduce linguistic nuance. This limitation is acknowledged and further discussed in the limitations section. Fig. 1 illustrates the preprocessing pipeline utilized to preprocess the dataset.

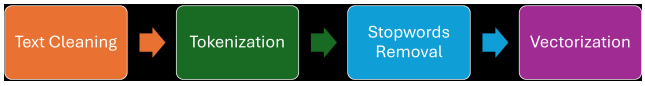


Fig. 1. Preprocessing pipeline.

The TF-IDF representation inherently emphasizes discriminative keywords by assigning higher weights to terms that are frequent within a category but rare across the corpus. Analysis of high-weight features revealed domain-relevant keywords such as "robbery", "accident", "curfew", and "agreement", which align with the semantic structure of incident categories. While this study focuses on classification performance, dimensionality reduction and visualization techniques such as Principal Component Analysis (PCA) can further enhance interpretability by revealing latent keyword groupings and category separability. This is identified as a valuable extension for future analytical work.

The preprocessing pipeline in Fig. 1 describes the sequential process for converting raw barangay incident reports into structured textual data ready for machine learning-based classification. It starts with text cleaning, which normalizes the reports by converting all characters to lowercase and deleting punctuation, special characters, and non-alphanumeric tokens. This is done to ensure uniformity and aggregate noise, which can negatively affect feature extraction.

After that, tokenization splits the text into distinct words, or tokens, so the model can process each word as a separate feature. This is vital in converting narrative-type reports into tabular inputs. Finally, stopword elimination is carried out in the removal of commonly occurring yet semantically weak words (e.g., "the", "and", "of") that do not provide meaningful classification. Dimensionality is lowered with the retention of words that are the highest information-containing words in the differentiation between incident categories.

The last step in the pipeline is vectorization, in which the tokenized and cleaned text is converted to a numerical representation using the TF-IDF method. TF-IDF vectorization was configured with a maximum feature size of 5000 terms, representing the most informative unique keywords after preprocessing. This upper bound was selected to balance representational richness with computational efficiency. Least frequent terms were implicitly filtered through TF-IDF weighting and feature capping, ensuring that extremely rare and non-informative tokens did not contribute to model complexity or noise. It gives more weight to words that occur frequently in a specific document and are relatively rare across the whole corpus, enabling the model to grasp the unique properties of each incident report. While TF-IDF is effective for capturing discriminative keyword importance, it does not encode semantic relationships, word order, or contextual meaning. As a result, it may struggle with polysemy and semantically similar expressions across categories. This limitation motivates the inclusion of embedding-based baselines such as FastText in the experimental evaluation.

By fulfilling this pipeline, textual data is converted into computationally tractable, semantically formatted data, enabling classical machine learning models to be trained efficiently. In this study, a limit of up to 5000 features was set to balance information richness and computational efficiency. This reduction in features is key to achieving efficiency and lightweight operation, which are requirements for mobile-deployable programs.

This study used a structured benchmarking framework for incident report classification designed to ensure statistical robustness and deployment feasibility. The framework consists of five primary stages: label encoding, TF-IDF feature extraction, repeated stratified cross-validation, classifier training, and statistical evaluation. To prevent data leakage and ensure fair comparison, TF-IDF vectorization is embedded within the cross-validation pipeline, allowing feature extraction to be performed independently within each training fold. The overall workflow is illustrated in Algorithm 1.

Algorithm 1: Incident Report Classification Framework

INPUT: Text dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$

OUTPUT: Trained classifier M and aggregated performance metrics

1. Encode categorical labels into numerical form
2. Initialize repeated stratified 5 × 5 cross-validation
3. **For** each fold:
   - Split the dataset into training and validation sets
   - Fit TF-IDF vectorizer on training data
   - Transform validation data using fitted vectorizer
   - Train classifier M
   - Evaluate performance using accuracy and weighted F1-score
4. Aggregate results across 25 runs
5. Perform Wilcoxon signed-rank statistical testing
6. Compute effect size (Cohen's d)
7. Rank classifiers based on mean performance

The incident report classification framework presented in Algorithm 1 follows a structured, statistically robust procedure that ensures reproducibility, fairness, and deployment feasibility. Let the dataset be represented as $D = \{(x_i, y_i)\}_{i=1}^{n}$, where x_i denotes the textual content of the i-th incident report and y_i corresponds to its categorical label. The algorithm integrates feature extraction, repeated stratified cross-validation, supervised learning, and statistical validation within a unified evaluation pipeline.

Initially, categorical labels are transformed into numerical representations to ensure compatibility with machine learning algorithms. To obtain reliable performance estimates and mitigate the effects of class imbalance, repeated stratified 5 × 5 cross-validation is employed. Stratification preserves the original class distribution within each fold, whereas repetition increases the total number of evaluation runs to 25, thereby enhancing statistical stability and reducing variance in performance estimates.

Within each cross-validation iteration, the dataset is partitioned into training and validation subsets. Textual features are extracted using TF-IDF vectorization. Importantly, the vectorizer is fitted exclusively on the training subset and subsequently applied to the validation subset. This fold-wise feature transformation prevents data leakage and ensures unbiased evaluation. The selected classifier is then trained on the transformed training data and used to generate predictions for the validation set. Performance metrics, including accuracy and weighted F1-score, are computed for each fold. To account for class imbalance, the weighted F1-score is emphasized alongside macro metrics.

Upon completion of all cross-validation runs, performance scores are aggregated to obtain mean accuracy, standard deviation, and mean weighted F1-score for each classifier. To assess whether observed differences between models are statistically meaningful, the Wilcoxon signed-rank test is conducted for pairwise comparison against the highest-performing model. Given the repeated cross-validation setting and potential non-normality of score distributions, this non-parametric test provides an appropriate and conservative evaluation of statistical significance. In addition, Cohen's *d* effect size is calculated to quantify the magnitude of performance differences and enable interpretation beyond *p*-values alone.

Lastly, classifiers are ranked according to their mean performance across all evaluation runs. The combined consideration of predictive accuracy, statistical significance, effect size, and computational efficiency provides a comprehensive basis for identifying the most suitable model for deployment. By integrating repeated stratified validation, strict leakage control, and statistical comparison, the proposed algorithmic framework ensures rigorous benchmarking while maintaining alignment with the practical objective of mobile-deployable incident classification.

To ensure reproducibility, the preprocessing pipeline, model training procedures, and evaluation scripts will be made publicly available through an online repository upon publication. The link is accessible at https://bit.ly/ML-IncidentReports.

B. Comparison of the Performance of Machine Learning Algorithms

The comparative evaluation of machine learning algorithms in the table reveals critical insights into their predictive performance, as assessed using Precision, Recall, and F1-score metrics at both macro and weighted-average levels. The macro average treats all classes equally, thereby providing a balanced overview across different categories. In contrast, the weighted average considers class imbalance by assigning weights proportional to the number of instances per class. This distinction is essential in interpreting classifier performance, especially when datasets are imbalanced [41, 42]. Table 3 presents the performance metrics for each model, with a support of 125.

Among the evaluated models, GB was the best-performing, with the highest macro-average precision (0.81) and a high F1-score (0.73 macro, 0.71 weighted), indicating the highest predictive performance and generalization. Random Forest showed very stable performance, with macro precision of 0.78 and an F1-score of 0.69, again establishing its credibility as a robust ensemble method suitable for handling high-dimensional, non-linear data. SVM showed strong buckling performance across all metrics, with very high recall and F1-scores (0.71–0.73), indicating its ability to correctly detect both the majority and minority classes. These results align with Islam *et al.*'s work [43], which stresses the credibility of SVMs and ensemble methods for varied classification tasks.

Table 3. Performance metrics of machine learning algorithms (support = 125)

Algorithm	Precision	Recall	F1-score
LR	0.73*	0.59*	0.62*
	0.75**	0.68**	0.68**
NB	0.66*	0.52*	0.54*
	0.70**	0.64**	0.62**
SVM	0.73*	0.71*	0.71*
	0.76**	0.74**	0.73**
RF	0.78*	0.67*	0.69*
	0.76**	0.70**	0.71**
SGD	0.72*	0.68*	0.69*
	0.73**	0.70**	0.70**
PA	0.72*	0.69*	0.70*
	0.74**	0.71**	0.72**
GB	0.81*	0.70*	0.73*
	0.79**	0.70**	0.71**
FastText	0.69*	0.62*	0.64*
	0.72**	0.66**	0.66**

Note: *macro average; **weighted average

On the other hand, Linear models like Logistic Regression and Naïve Bayes had lower recall and F1-scores than ensemble-based methods, indicating their over-reliance on class-dominated data and their inability to capture nonlinearity in feature relationships. Naïve Bayes, in particular, demonstrated lower recall (0.52 macro, 0.64 weighted), likely due to the assumption that features are independent, which is typically not the case in real-world data [29]. Conversely, SGD and Passive-Aggressive classifiers achieved decent performance, achieving F1-scores comparable to those of SVMs, making them viable alternatives for large-scale and streaming datasets, given that computational intensity is a priority [44].

Including the FastText baseline provides additional insight into the role of semantic feature representations in incident classification. FastText achieved a mean accuracy of 0.66 and a weighted F1-score of 0.66, demonstrating competitive performance relative to classical models. The use of subword-level embeddings allows FastText to better handle lexical variability and informal language patterns commonly present in incident reports.

However, despite its semantic advantages, FastText did not outperform the Linear SVM model. The SVM achieved a higher mean accuracy of 0.69. This suggests that, for moderate-sized and domain-specific datasets with sparse textual features, linear classifiers remain highly effective. The relatively small performance gap indicates that TF-IDF representations are sufficient to capture discriminative patterns in structured incident report data.

Furthermore, FastText exhibited lower macro recall (0.62) compared to SVM (0.71), indicating reduced sensitivity to minority classes. This may be attributed to the limited dataset size and class imbalance, which can constrain the effectiveness of embedding-based methods.

The distribution of keywords across incident categories

significantly affects model performance. Classes with richer and more distinctive vocabularies benefit from stronger feature separability, whereas minority classes with limited lexical diversity exhibit higher misclassification rates. Models such as SVM and Gradient Boosting demonstrate greater robustness to skewed keyword distributions, while simpler linear models are more sensitive to dominant terms associated with majority classes.

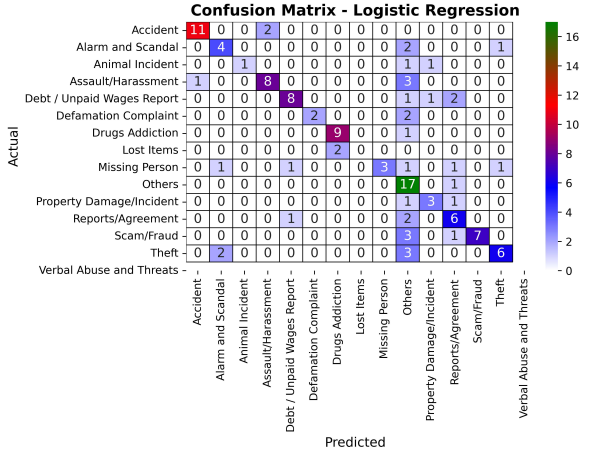
This shows that ensemble-based algorithms, particularly Gradient Boosting and Random Forest, contain the highest trade-off between precision and recall, corroborating more recent work by Shahani *et al*. [37] and Yao *et al.* [45]. These methods effectively reduce variance and bias through sequential model correction, providing better predictive stability. Also, a high-weighted-average score across most models suggests class imbalance in the dataset. Reporting both macro and weighted metrics ensures clear evaluation of classifier fairness and performance across different data distributions [42, 46].

Recent literature demonstrates the growing dominance of deep learning and transformer-based architectures in text classification tasks. Comprehensive surveys by Rady *et al.* [4] and Taha *et al.* [35] indicate that models such as BERT [17] and deep ensemble frameworks [38] frequently achieve superior predictive performance across benchmark datasets. In domain-specific applications, transformer-based approaches have shown strong results in multilingual and contextual settings, including Filipino political sentiment analysis [18]. Similarly, hybrid architectures integrating TF-IDF with LSTM networks have demonstrated performance gains in structured classification problems [15].

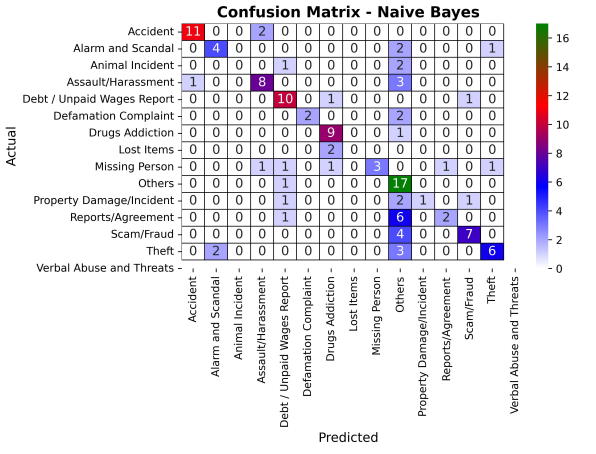
However, prior work also highlights that model effectiveness is strongly influenced by dataset size, feature sparsity, and domain variability [23, 32]. Deep neural models typically require large-scale corpora and substantial computational resources to fully realize their advantages. In contrast, classical linear classifiers, particularly Support Vector Machines, remain competitive in sparse, moderate-sized datasets [14, 47], and have demonstrated reliable performance in Filipino-language classification tasks [20, 21].

C. *Analysis of Confusion Matrices on Misclassifications*

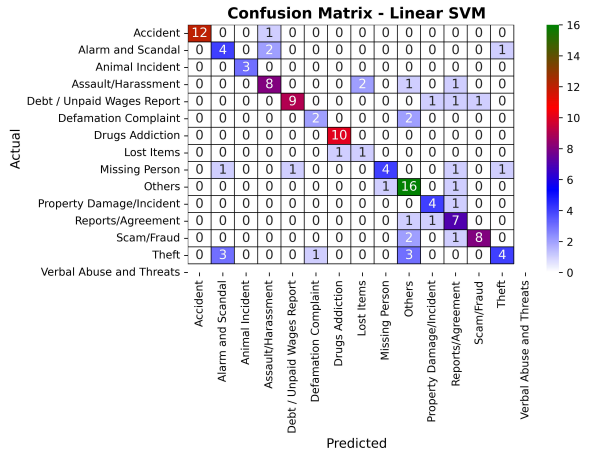
To analyze misclassifications in trained algorithms, confusion matrices based on the test subset were visualized to determine the distribution of actual and predicted values. Fig. 2 presents the confusion matrices of all evaluated classifiers, providing detailed insights into class-level prediction behavior and misclassification patterns. Each matrix illustrates the distribution of predicted labels versus actual labels, with diagonal elements representing correct classifications and off-diagonal elements indicating misclassifications. This analysis enables a deeper understanding of model behavior beyond aggregate performance metrics.



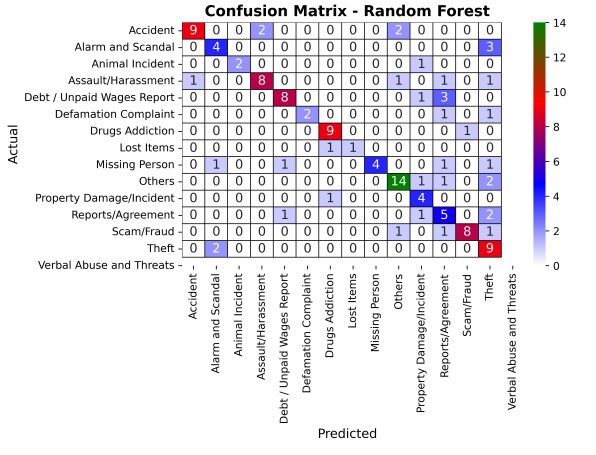
(a)



(b)



(c)



(d)

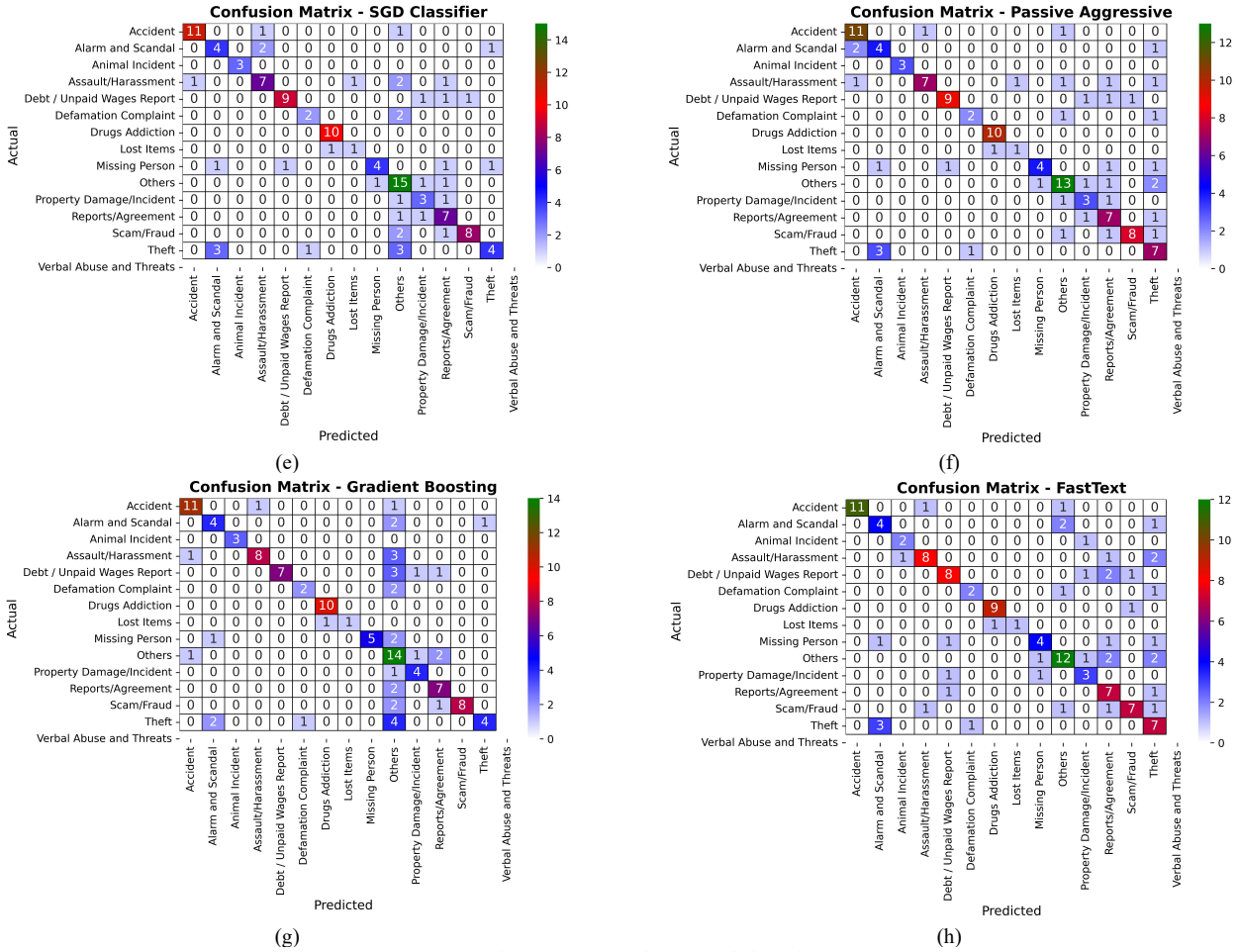


Fig. 2. Confusion matrices of compared algorithms.

Fig. 2(a) shows that Logistic Regression achieves reasonable classification accuracy, with strong recognition in categories such as “Accident” and “Others”. However, systematic misclassification is evident between semantically related categories such as “Assault/Harassment” and “Debt/Unpaid Wages Report”. These cross-category confusions suggest lexical and contextual overlap in complaint narratives. Linear classifiers such as LR rely on linear decision boundaries, which limit their ability to separate classes with subtle semantic distinctions [14, 47].

As illustrated in Fig. 2(b), Naïve Bayes exhibits broader dispersion in misclassifications, particularly between “Debt/Unpaid Wages Report” and “Scam/Fraud”. This pattern reflects the limitations of its conditional independence assumption, which fails to capture dependencies between words in real-world text data [48]. While NB maintains stable performance in dominant categories such as “Others”, it struggles in context-sensitive classes like “Defamation Complaint”, where semantic nuance is critical [49].

Fig. 2(c) demonstrates the superior discriminative capability of Linear SVM, with strong diagonal dominance across most categories, including “Accident” and “Drug Addiction”. Compared to LR and NB, SVM significantly reduces cross-category confusion, confirming its effectiveness in handling high-dimensional sparse representations [30, 31]. However, residual confusion persists among linguistically overlapping categories such as “Scam/Fraud”, “Debt/Unpaid Wages Report”, and “Verbal Abuse and Threats”, indicating that even robust linear separators are limited by feature overlap.

The Random Forest matrix in Fig. 2(d) shows relatively balanced classification across multiple classes, including “Accident”, “Others”, and “Scam/Fraud”. Its ensemble structure enables modeling of nonlinear feature interactions, thereby improving generalization [32, 33]. However, confusion remains in semantically similar categories, suggesting that tree-based partitioning alone is insufficient to fully resolve lexical ambiguity in textual data.

Fig. 2(e) shows that the SGD classifier achieves performance comparable to that of the SVM, with strong recognition in dominant categories such as “Others” and “Drug Addiction”. However, sporadic misclassifications are observed in categories such as “Theft” and “Assault/Harassment”, likely due to sensitivity to noisy features during stochastic optimization [34, 35]. This highlights a trade-off between computational efficiency and robustness.

The Passive Aggressive classifier in Fig. 2(f) demonstrates balanced and robust performance, with strong classification in “Drug Addiction”, “Debt/Unpaid Wages Report”, and “Scam/Fraud”. Its margin-based online learning mechanism allows effective adaptation to streaming text data [36]. However, confusion in categories such as “Verbal Abuse and Threats” and “Theft” suggests that overlapping linguistic expressions still pose challenges even for adaptive models.

Fig. 2(g) presents the confusion matrix for Gradient Boosting, which shows strong diagonal consistency across the categories “Others”, “Accident”, and “Drug Addiction”. Its iterative boosting process effectively reduces both bias and variance, thereby improving generalization [37, 38].

Nevertheless, minor confusion persists in low-frequency classes, indicating that even ensemble refinement cannot fully compensate for data scarcity.

Fig. 2(h) presents the FastText confusion matrix, which reflects the behavior of embedding-based semantic representations. FastText demonstrates reasonable classification performance in dominant categories such as "Others" and "Drug Addiction", indicating its ability to capture general semantic patterns. However, notable misclassification is observed between semantically similar categories such as "Scam/Fraud", "Debt/Unpaid Wages Report", and "Verbal Abuse and Threats". This suggests that while FastText captures word-level semantics, it may struggle with fine-grained distinctions in short, domain-specific complaint texts. Additionally, minority classes such as "Defamation Complaint" and "Lost Items" exhibit higher misclassification rates, likely due to insufficient training samples to effectively learn representative embeddings.

A consistent pattern across all models is the presence of systematic confusion between semantically overlapping categories. For example, financial-related complaints ("Debt/Unpaid Wages Report" vs. "Scam/Fraud") and interpersonal conflict categories ("Assault/Harassment" vs. "Verbal Abuse and Threats") frequently exhibit bidirectional misclassification. These errors are primarily driven by lexical similarity and shared contextual expressions within complaint narratives.

Another critical observation is the impact of class imbalance. Categories with very limited samples, such as "Defamation Complaint" and "Lost Items", consistently show higher error rates across all models. This indicates that insufficient training data significantly limits classifiers' ability to learn discriminative features for minority classes.

From a practical governance perspective, these misclassification patterns have important implications. Errors between closely related categories may delay appropriate responses or misallocate resources in incident management systems. For instance, misclassifying a fraud-related complaint as a general financial dispute may affect prioritization and intervention strategies.

Overall, margin-based classifiers such as SVM and SGD demonstrate the most consistent performance with minimal cross-category confusion, aligning with their strong quantitative results. Ensemble methods such as Random Forest and Gradient Boosting provide robustness but at a higher computational cost. FastText, while capable of capturing semantic relationships, does not significantly reduce misclassification in this dataset, suggesting that embedding-based methods may require larger and more diverse corpora to outperform sparse linear representations.

These findings reinforce the importance of both feature representation and dataset characteristics in multi-class text classification. While performance metrics provide a global view, confusion matrix analysis reveals critical class-level behaviors that are essential for designing reliable and deployable incident classification systems.

D. Model Selection for Deploying to Mobile Devices

Table 4 presents the theoretical training and inference complexities of the evaluated classifiers, where n denotes the number of training samples, d denotes the dimensionality of the TF-IDF feature space, and t denotes the number of trees in ensemble-based methods. This analysis provides insight into the computational feasibility of each model, particularly in mobile or resource-constrained deployment environments.

Table 4. Training and inference complexities of machine learning algorithms

Model	Training Complexity	Inference Complexity
LR	$O(nd)$	$O(d)$
NB	$O(nd)$	$O(d)$
SVM	$O(nd)$	$O(d)$
SGD	$O(nd)$	$O(d)$
PA	$O(nd)$	$O(d)$
RF	$O(tn \log n)$	$O(t \log n)$
GB	$O(tn \log n)$	$O(t \log n)$
FastText	$O(nkd)$	$O(kd)$

For linear models, including LR, SVM, SGD, and PA, both training and inference exhibit linear dependence on the number of samples and features. Specifically, training complexity is approximately $O(nd)$, reflecting iterative optimization over all samples and features. During inference, predictions require only a single dot-product operation between the learned weight vector and the input feature vector, resulting in $O(d)$ complexity. This low inference cost makes linear classifiers computationally attractive for real-time prediction scenarios, particularly when operating on high-dimensional but sparse TF-IDF representations.

NB similarly exhibits linear training complexity $O(nd)$, as it estimates class-conditional probabilities directly from feature counts. Inference also scales linearly with feature dimensionality. Despite its computational simplicity, the probabilistic independence assumption may limit predictive performance in complex textual datasets, as observed in the empirical results.

In contrast, ensemble-based methods such as Random Forest and Gradient Boosting incur higher computational cost. Training complexity scales as $O(tn \log n)$, as multiple decision trees must be constructed, each involving recursive partitioning of the dataset. Although inference complexity is reduced compared to training, predictions require traversal of all trees, resulting in $O(t \log n)$ operations. As the number of trees increases to improve predictive performance, both memory usage and inference latency grow proportionally.

FastText introduces a different computational profile by combining linear classification with dense word embeddings. Its training complexity can be approximated as $O(nkd)$, where k represents the embedding dimension. This reflects the need to update both word embeddings and classifier weights during training. During inference, FastText requires embedding lookup and vector averaging before classification, resulting in $O(kd)$ complexity. While this remains linear, it introduces additional computational overhead compared to TF-IDF-based linear models due to dense vector operations.

From a deployment perspective, inference complexity is particularly critical. In mobile or edge environments, prediction latency and computational overhead directly affect responsiveness and energy consumption. Linear classifiers demonstrate clear advantages in this regard, as their inference cost depends solely on feature dimensionality rather than embedding size or ensemble depth. Tree-based methods, while competitive in accuracy, may introduce additional

computational burden due to multiple tree evaluations per prediction. FastText, while still relatively efficient, incurs higher memory usage and computational cost compared to sparse linear models due to the storage and processing of dense embeddings.

When these theoretical findings are considered alongside the empirical results, a consistent pattern emerges. Linear SVM and SGD-based classifiers achieved the highest predictive performance while maintaining low inference complexity. Although FastText achieved competitive results and improved semantic representation, it did not surpass SVM performance and introduced additional computational overhead. This suggests that, for moderate-sized and domain-specific datasets, sparse linear representations remain highly effective.

The complexity analysis supports the central premise of this study: that carefully selected classical machine learning models, particularly linear classifiers, remain strong candidates for resource-aware text classification applications. While embedding-based approaches such as FastText offer semantic advantages, their additional computational requirements must be considered when targeting mobile-deployable systems.

Table 5 presents the comparative performance of the evaluated machine learning algorithms across three dimensions: accuracy (%), computational time (s), and model weight. These metrics provide a balanced assessment of predictive performance and computational efficiency, both of which are critical for deployment in real-time or resource-constrained environments such as mobile or edge devices [32].

Table 5. Accuracy, computational time, and model weight of machine learning algorithms

Algorithm	Accuracy (%)	Computational Time (s)	Model Weight
LR	68	0.034	Light
NB	64	0.013	Light
SVM	74	0.019	Moderate
RF	70	0.501	Heavy
SGD	70	0.036	Light
PA	71	0.030	Light
GB	70	3.686	Heavy
FastText	66	0.014	Light-Moderate

In terms of accuracy, the Linear SVM achieved the highest performance (74%), confirming its strong capability to learn optimal decision boundaries in high-dimensional feature spaces typical of text classification tasks. The Passive Aggressive (PA) and Stochastic Gradient Descent (SGD) classifiers followed closely, achieving 71% and 70%, respectively. These results highlight the effectiveness of margin-based and online learning approaches in handling sparse textual representations while maintaining strong generalization performance.

FastText achieved an accuracy of 66%, demonstrating competitive but lower performance compared to linear classifiers. Despite leveraging semantic representations through word embeddings and subword information, FastText did not outperform SVM or PA. This suggests that for moderately sized, domain-specific datasets, sparse TF-IDF representations remain sufficiently expressive, and the added semantic modeling provided by embeddings does not necessarily translate into higher classification accuracy.

By contrast, Naïve Bayes (64%) yielded the lowest accuracy, reflecting its simplifying assumption of feature independence, which is often violated in natural language data. Logistic Regression (68%) performed slightly better but remained limited by its linear decision boundary. Ensemble-based methods such as Random Forest (70%) and Gradient Boosting (70%) achieved competitive accuracy, demonstrating robustness in handling noisy and nonlinear data patterns. However, their performance gains were marginal relative to their increased computational cost.

In terms of computational time, a clear trade-off between model complexity and runtime efficiency is observed. Naïve Bayes (0.013 s) and FastText (0.014 s) were among the fastest models, followed closely by SVM (0.019 s) [29]. This indicates that FastText maintains efficient training and inference despite incorporating embedding-based representations. Logistic Regression (0.034 s), SGD (0.036 s), and PA (0.030 s) also exhibited lightweight computational behavior, reinforcing their suitability for real-time applications [37].

In contrast, ensemble-based methods demonstrated significantly higher computational cost. Random Forest required 0.501 s, while Gradient Boosting incurred the highest runtime at 3.686 s due to its sequential learning process. These results highlight the inherent computational burden of ensemble methods, particularly when multiple trees are used for both training and inference [36].

From a model weight perspective, lightweight models such as NB, LR, SGD, and PA are highly suitable for deployment in mobile environments due to their minimal memory requirements and low computational overhead. FastText, while relatively efficient, introduces additional memory requirements due to the storage of embedding vectors, positioning it between lightweight and moderate models. Ensemble methods, particularly Gradient Boosting, are considered heavyweight due to their large model sizes and computational demands.

An important observation from this comparison is the trade-off between accuracy and computational efficiency [33, 38]. While Gradient Boosting and Random Forests provide stable, robust performance, their computational cost may limit their practicality for real-time or mobile deployment. Conversely, lightweight models such as Naïve Bayes and Logistic Regression offer fast execution but at the expense of predictive performance.

Notably, SVM, PA, and SGD offer the best balance between accuracy and efficiency. SVM, in particular, demonstrates that high predictive performance can be achieved without incurring substantial computational overhead. Although FastText introduces semantic modeling capabilities and performs efficiently, it does not surpass linear classifiers in this context, suggesting that embedding-based methods may require larger datasets or more complex architectures to fully realize their advantages.

These findings reinforce the central premise of this study: that classical machine learning models, particularly linear margin-based classifiers, remain highly effective for domain-specific text classification tasks. When computational efficiency and deployment feasibility are critical considerations, models such as SVM, PA, and SGD offer optimal trade-offs, while FastText is a viable but not superior

alternative for moderate-sized datasets.

In terms of deployment, the recommendations are that the model choice will depend on the context in which the system is operational. For mass-scale systems that involve batchwise operation and require interpretability, RF and GB offer advantages through ensemble robustness and resistance to overfitting. But when dealing with real-time analytics, mobile, or streamwise data, SGD and PA Classifiers have the advantage through low latency and the possibility of online adaptability [36]. In the meantime, SVM remains optimal for environments that demand classification accuracy and stability with non-prohibitive computational overhead.

A comparative analysis shows that SVM, PA, and SGD classifiers offer the best trade-off between accuracy and computational speed, making them suitable for scalable incident classification applications. Ensemble methods such as RF and GB provide high robustness and stability, though at the cost of computational overhead, whereas LR and NB remain appropriate for fast, lightweight, and interpretable solutions. Balancing computational speed with classification accuracy remains a key consideration when choosing algorithms, and future work is suggested to investigate deep hybrid models and lightweight transformer-based approaches further to better balance this trade-off.

Compared with recent state-of-the-art studies that employ deep learning or transformer-based models, this study demonstrates that classical machine learning models can achieve competitive performance when applied to domain-specific, moderate-sized datasets. Importantly, this performance is achieved with significantly lower computational overhead, reinforcing the suitability of classical models for deployment-constrained environments.

Table 6. Statistical comparison of SVM against other models (repeated stratified 5 × 5 cross-validation, 25 runs)

Model	Mean Accuracy	Std.	*p*-value	Cohen's *d*	Sig.
LR	0.6579	0.0504	0.000724	0.896	S
NB	0.6213	0.0449	0.000018	1.955	S
RF	0.6794	0.0443	0.010074	0.557	S
SGD	0.6909	0.0433	0.642777	0.126	NS
PA	0.6803	0.0398	0.004537	0.672	S
GB	0.6573	0.0434	0.000152	1.153	S
FastText	0.6633	0.0533	0.000309	0.974	S

Note: Linear SVM (Mean Accuracy = 0.6932; Std. = 0.0408), Sig. = Significance ($\alpha = 0.05$), S = Significant, NS = Not Significant

The results presented in Table 6 summarize the statistical comparison between Linear SVM, which achieved the highest mean accuracy, and the remaining evaluated classifiers using repeated stratified 5 × 5 cross-validation (25 runs). The Wilcoxon signed-rank test was employed to determine whether observed differences in predictive performance were statistically significant at the $\alpha = 0.05$ level, while Cohen's d was computed to assess practical effect size.

The analysis reveals that Linear SVM significantly outperformed Logistic Regression ($p = 0.000724$), Naïve Bayes ($p = 0.000018$), Random Forest ($p = 0.010074$), Passive Aggressive ($p = 0.004537$), Gradient Boosting ($p = 0.000152$), and FastText ($p = 0.000309$). In contrast, no statistically significant difference was observed between Linear SVM and the SGD Classifier ($p = 0.642777$).

Effect size interpretation follows conventional thresholds, where $d \approx 0.2$ indicates a small effect, $d \approx 0.5$ a medium effect, and $d \geq 0.8$ a large effect. Based on this, large practical effects were observed when comparing Linear SVM against Logistic Regression ($d = 0.896$), FastText ($d = 0.974$), Naïve Bayes ($d = 1.955$), and Gradient Boosting ($d = 1.153$). Moderate effects were observed for Random Forest ($d = 0.557$) and Passive Aggressive ($d = 0.672$). In contrast, the negligible effect size between Linear SVM and SGD ($d = 0.126$) confirms nearly equivalent predictive behavior.

The absence of a significant difference between Linear SVM and SGD Classifier is theoretically consistent. The SGD classifier, when configured with hinge loss, approximates linear SVM optimization. Both methods construct linear decision boundaries in high-dimensional feature spaces, explaining their nearly identical performance across folds.

The significant differences observed between Linear SVM and tree-based ensemble methods suggest that sparse TF-IDF representations are better exploited by linear margin-based classifiers than by recursive partitioning methods. Text data typically produces high-dimensional and sparse vectors, conditions under which linear hyperplane separation is particularly effective.

Furthermore, the extremely large effect size observed relative to Naïve Bayes indicates that the model's conditional independence assumption is insufficient to capture the complex term dependencies in incident reports.

In comparison, FastText achieved competitive performance but was significantly outperformed by Linear SVM ($p = 0.000309$, $d = 0.974$). Although FastText incorporates subword-level semantic information through embeddings, its advantage did not translate into superior classification performance in this dataset. This suggests that for moderately sized, domain-specific corpora with structured language patterns, sparse TF-IDF representations may already capture sufficient discriminative information, reducing the marginal benefit of embedding-based approaches.

From a deployment perspective, these statistical findings align with the computational complexity analysis. Linear SVM achieves statistically superior performance while maintaining low inference complexity $O(d)$, making it highly suitable for real-time mobile deployment. Ensemble models, although competitive, require multiple tree traversals during prediction, increasing computational overhead. FastText, while efficient, introduces additional memory requirements and computational cost due to embedding storage and vector operations.

It should be noted that the deployment analysis is based on theoretical complexity and simulated runtime measurements. Actual performance may vary depending on hardware constraints such as memory capacity, processing power, and energy consumption.

In a broader context, these findings reinforce the idea that classical machine learning models remain highly competitive for domain-specific text classification tasks. While transformer-based systems may achieve higher performance on large-scale datasets, Linear SVM demonstrated statistically validated performance (0.6932) with significantly lower computational overhead. Given the deployment constraints inherent in local governance

systems [7, 9], such efficiency advantages are critical. These results therefore support the continued relevance of classical margin-based models as practical and deployable solutions in resource-constrained environments.

V. Limitations and Future Work

Despite the methodological rigor and statistically validated comparisons presented in this study, several limitations must be acknowledged.

First, the feature representation was restricted to TF-IDF, which, although computationally efficient and well-suited for sparse text classification, does not capture contextual semantics, word order, or polysemy. Incident narratives often contain nuanced descriptions, abbreviations, and context-dependent meanings that bag-of-words representations may fail to model effectively. As a result, classification performance may be constrained by representational limitations rather than the inherent capability of the learning algorithms.

Second, the dataset was derived from a specific local governance context, consisting of incident reports from five barangays within a single city. While this enhances ecological validity for barangay-level applications, it may limit generalizability to other regions with different linguistic characteristics, reporting styles, and category distributions. Additionally, the translation of reports from Filipino/Tagalog to English may have introduced semantic distortion and altered vocabulary patterns, potentially affecting model performance.

Third, the dataset exhibits class imbalance, with some incident categories underrepresented. Although weighted evaluation metrics were used to account for this imbalance, no explicit mitigation techniques (e.g., resampling or class weighting) were applied. This may have contributed to reduced performance in minority classes, as reflected in the confusion matrix analysis.

Fourth, while computational complexity was analyzed theoretically, empirical benchmarking on actual deployment platforms (e.g., mobile devices or low-resource servers) was not conducted. Real-world performance in terms of latency, memory usage, and energy consumption may differ from theoretical estimates.

Lastly, hyperparameter tuning was limited to standard configurations to ensure fair comparison across models. More extensive optimization may further improve performance.

Future work will address these limitations by exploring semantic feature representations, including Word2Vec and lightweight transformer-based models, to better capture contextual relationships in incident narratives. Hybrid approaches that combine embedding-based representations with efficient linear classifiers will also be investigated to balance semantic richness and computational efficiency.

In addition, future studies will incorporate class imbalance mitigation techniques, such as Synthetic Minority Oversampling Technique (SMOTE), class weighting, and data augmentation, to improve performance for underrepresented categories. Cross-regional and multilingual datasets will be utilized to evaluate model generalizability across diverse governance contexts.

Furthermore, empirical deployment studies will be conducted to evaluate real-world performance on mobile and edge devices, focusing on latency, memory footprint, and energy consumption. Longitudinal evaluation within live incident management systems will also be essential to assess scalability, adaptability, and sustained performance in operational environments.

VI. Conclusion

This study presented a statistically grounded comparative evaluation of classical machine learning algorithms for the automated classification of barangay incident reports using TF-IDF-based text representation. By examining models across probabilistic, linear, margin-based, online, and ensemble learning paradigms, the analysis extended beyond accuracy-centric comparison by incorporating statistical validation and computational considerations relevant to real-world deployment.

Using repeated stratified 5×5 cross-validation (25 runs), Linear Support Vector Machine (SVM) achieved the highest overall performance, with a mean accuracy of 0.6932 and a weighted F1-score of 0.6598. The FastText baseline achieved competitive performance (accuracy = 0.6633), but was significantly outperformed by SVM. Wilcoxon signed-rank testing confirmed that SVM significantly outperformed Logistic Regression, Naïve Bayes, Random Forest, Passive Aggressive, Gradient Boosting, and FastText ($p < 0.05$), with moderate to large effect sizes observed across multiple comparisons. No statistically significant difference was found between SVM and the SGD Classifier, reflecting their shared linear optimization characteristics.

Beyond predictive performance, computational complexity analysis demonstrated that linear and margin-based models incur substantially lower inference overhead than ensemble and embedding-based approaches. This efficiency advantage, combined with strong classification performance, highlights their suitability for deployment in mobile and resource-constrained governance environments.

The findings demonstrate that carefully selected classical machine learning models remain both effective and computationally practical for domain-specific text classification tasks. This study provides statistically validated, deployment-oriented insights to guide model selection for intelligent local governance systems. Future work will explore multilingual modeling and lightweight embedding-based approaches to further improve performance while maintaining deployment feasibility.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

A.P.M.D.R trained and tested the model, generated the metrics and confusion matrices for the compared models, and wrote the results and discussion section. D.Q.F. finalized the annotated dataset and developed the data-cleaning and pre-processing codes. R.P.R. wrote the literature review section and subsections in the methodology. E.C.S. wrote the remaining parts of the paper and identified the models to be compared. A.P.M.D.R. finalized the paper, and all authors approved of the final version for publication.

ACKNOWLEDGMENT

The authors would like to thank Stephan Kelian S. Aguilar and his colleagues for their help in annotating the dataset. The authors would also like to thank the five barangays for giving their data for the dataset.

REFERENCES

[1] C. J. M. Mejia, N. P. Macinas, J. K. A. Catacutan *et al.*, "Application of spatio-temporal analysis in criminal incident recording system," *European Journal of Engineering and Technology Research*, vol. 8, no. 2, pp. 79–82, 2023. doi: 10.24018/ejeng.2023.8.2.3016

[2] I. J. B. Young, S. Luz, and N. Lone, "A systematic review of natural language processing for classification tasks in the field of incident reporting and adverse event analysis," *International Journal of Medical Informatics*, vol. 132, 103971, 2019. doi: 10.1016/j.ijmedinf.2019.103971

[3] H. J. Aleqabie, M. S. Sfoq, R. A. Albeer *et al.*, "A review of textmining techniques: Trends, and applications in various domains," *Iraqi Journal for Computer Science and Mathematics*, vol. 5, no. 1, 9, 2024. doi: 10.52866/ijcsm.2024.05.01.009

[4] G. S. Rady, K. F. Hussain, and A. I. Taloba, "Advancements in text classification: A comprehensive survey of machine learning and deep learning models," in *Proc. 2024 International Conference on Computer and Applications (ICCA)*, 2024, pp. 1–6. doi: 10.1109/ICCA62237.2024.10927763

[5] E. Laparra, S. Bethard, and T. A. Miller, "Rethinking domain adaptation for machine learning over clinical language," *JAMIA Open*, vol. 3, no. 2, pp. 146–150, 2020. doi: 10.1093/jamiaopen/ooaa010

[6] H. Abdelmotaleb, C. McNeile, and M. Wojtys, "A comparative study of word embedding techniques for classification of star ratings," *Expert Systems with Applications*, vol. 297, 129037, 2025. doi: 10.1016/j.eswa.2025.129037

[7] X. Dou, W. Chen, L. Zhu *et al.*, "Machine learning for smart cities: A comprehensive review of applications and opportunities," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 9, 2023. doi: 10.14569/IJACSA.2023.01409104

[8] F. Muheidat, M. A. Mallouh, O. Al-Saleh *et al.*, "Applying AI and machine learning to enhance automated cybersecurity and network threat identification," *Procedia Computer Science*, vol. 251, pp. 287–294, 2024. doi: 10.1016/j.procs.2024.11.112

[9] A. B. Brillantes Jr and K. E. V. Ruiz, "Public Administration in the Philippines: Features, Trends, Issues, and Directions," in *Handbook on Asian Public Administration*, Cheltenham: Edward Elgar Publishing, 2023, pp. 175–195. doi: 10.4337/9781839104794.00020

[10] K. Denecke and H. Paula, "Analysis of critical incident reports using natural language processing," *Studies in Health Technology and Informatics*, vol. 313, pp. 1–6, 2024. doi: 10.3233/shti240002

[11] I. C. Obasi, P. Cheng, C. Varianou-Mikellidou *et al.*, "Machine learning for occupational accident analysis: Applications, challenges, and future directions," *Journal of Safety Science and Resilience*, vol. 7, no. 1, 100250, 2026. doi: 10.1016/j.jnlssr.2025.100250

[12] F. Dakalbab, M. A. Talib, O. A. Waraga *et al.*, "Artificial intelligence & crime prediction: A systematic literature review," *Social Science & Humanities Open*, vol. 6, no. 1, 100342, 2022. doi: 10.1016/j.ssaho.2022.100342

[13] R. Romero, P. Celard, J. M. Sorribes-Fdez *et al.*, "MobyDeep: A lightweight CNN architecture to configure models for text classification," *Knowledge-Based Systems*, vol. 257, 109914, 2022. doi: 10.1016/j.knosys.2022.109914

[14] K. Kowsari, K. J. Meimandi, M. Heidarysafa *et al.*, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, 150, 2019. doi: 10.3390/info10040150

[15] P. Guleria, J. Frnda, and P. N. Srinivasu, "NLP-based text classification using TF-IDF enabled fine-tuned long short-term memory: An empirical analysis," *Array*, 100467, 2025. doi: 10.1016/j.array.2025.100467

[16] D. Mutesi, J. Ngugi, and S. Djuma, "Enhancing customer service in financial institutions through automated complaint classification: A machine learning approach," *Journal of Information and Technology*, vol. 5, no. 8, pp. 61–71, 2025.doi: 10.70619/vol5iss8pp61-71

[17] J. Devlin, M. W. Chang, K. Lee *et al.*, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.

[18] J. A. Aquino, D. J. Liew, and Y. C. Chang, "Graph-aware pre-trained language model for political sentiment analysis in Filipino social media," *Engineering Applications of Artificial Intelligence*, vol. 146, 110317, 2025. doi: 10.1016/j.engappai.2025.110317

[19] O. A. A. Francia, M. Nuñez-del-Prado, and H. Alatrista-Salas, "Survey of text mining techniques applied to judicial decisions prediction," *Applied Sciences*, vol. 12, no. 20, 10200, 2022. doi: 10.3390/app122010200

[20] M. G. E. Edaño, R. J. S. Gonzales, R. C. B. Laguda *et al.*, "Stressor classification of Filipino political tweets using LDA, SVM, XGBoost, logistic regression," in *Proc. International Conference on Industrial Engineering and Operations Management*, 2022, pp. 1378–1388.

[21] J. B. Campit, "Analyzing customer satisfaction using support vector machine and naive bayes utilizing Filipino text," *WSEAS Transactions on Environment and Development*, vol. 19, no. 50, pp. 514–524, 2023. doi: 10.37394/232015.2023.19.50

[22] M. B. L. Virtucio, J. A. Aborot, J. K. C. Abonita *et al.*, "Predicting decisions of the Philippine supreme court using natural language processing and machine learning," in *Proc. 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 2018, pp. 130–135. doi: 10.1109/COMPSAC.2018.10348

[23] D. Effrosynidis, G. Sylaios, and A. Arampatzis, "The effect of training data size on disaster classification from Twitter," *Information*, vol. 15, no. 7, 393, 2024. doi: 10.3390/info15070393

[24] F. A. Alshuwaier and F. A. Alsulaiman, "Fake news detection using machine learning and deep learning algorithms: A comprehensive review and future perspectives," *Computers*, vol. 14, no. 9, 394, 2025. doi: 10.3390/computers14090394

[25] S. J. Bagastio, S. Suakanto, and H. Fakhrurroja, "Optimizing deep learning model CNN-BiLSTM for fake product review classification," in *Proc. 2024 International Conference on Digital Business and Technology Management (ICONDBTM)*, 2024, pp. 1–6. doi: 10.1109/ICONDBTM64135.2024.11122800

[26] J. M. George, E. Joy, and S. Evangeline, "End-to-end personalized medical recommendation framework for disease prediction," in *Proc. 2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL)*, 2025, pp. 1429–1433. doi: 10.1109/ICSADL65848.2025.10933217

[27] T. K. Mishra, G. Sucharitha, N. Siddhartha *et al.*, "Prediction of suicidal behavior among the users on social media using NLP and ML," in *Proc. 2024 International Conference on Emerging Systems and Intelligent Computing (ESIC)*, 2024, pp. 74–81. doi: 10.1109/ESIC60604.2024.10481588

[28] H. A. Alatabi and A. R. Abbas, "Sentiment analysis in social media using machine learning techniques," *Iraqi Journal of Science*, pp. 193–201, 2020. doi: 10.24996/ijs.2020.61.1.22

[29] D. Liu, "Improvement of Naïve Bayes text classifier based on ensemble technology and feature engineering," in *Proc. 2023 International Conference on Image, Algorithms and Artificial Intelligence*, 2023, pp. 557–563. doi: 10.2991/978-94-6463-300-9_57

[30] C. A. Suarez, M. Castro, M. Leon *et al.*, "Improving SVM performance through data reduction and misclassification analysis with linear programming," *Complex & Intelligent Systems*, vol. 11, no. 8, 356, 2025. doi: 10.1007/s40747-025-01989-4

[31] A. P. M. D. Rosa and R. P. P. Abad, "Sentiment analysis of student's perspectives on the integration of a mobile fitness application in a physical education course using machine learning techniques," *International Journal of Computing Sciences Research*, vol. 7, pp. 2333–2347, 2023.

[32] J. Wainer, "An empirical evaluation of imbalanced data strategies from a practitioner's point of view," *Expert Systems with Applications*, vol. 256, 124863, 2024. doi: 10.1016/j.eswa.2024.124863

[33] A. S. Palli, J. Jaafar, M. H. M. Saad *et al.*, "Smart adaptive ensemble model for multiclass imbalanced nonstationary data streams," *Scientific Reports*, vol. 15, no. 1, p. 21140, 2025. doi: 10.1038/s41598-025-05122-w

[34] A. Markova, A. Khoroshun, and S. Volkova, "Comparative analysis of stochastic optimization methods for image classification using convolutional neural networks," in *Proc. 4th International Conference on Optics, Computer Applications, and Materials Science*, 2025, 136510S. doi: 10.1117/12.3061492

[35] K. Taha, P. D. Yoo, C. Yeun *et al.*, "A comprehensive survey of text classification techniques and their research applications: Observational and experimental insights," *Computer Science Review*, vol. 54, 100664, 2024. doi: 10.1016/j.cosrev.2024.100664

[36] M. S. Kim, L. Liu, and H. K. Kwon, "OL4TeX: Adaptive online learning for text classification under distribution shifts," in *Proc. 2024 IEEE International Conference on Big Data (BigData)*, 2024, pp. 1340–1345. doi: 10.1109/BigData62323.2024.10826003

[37] N. M. Shahani, M. Kamran, X. Zheng *et al.*, "Application of gradient boosting machine learning algorithms to predict uniaxial compressive strength of soft sedimentary rocks at Thar Coalfield," *Advances in Civil Engineering*, vol. 2021, no. 1, 2565488, 2021. doi: 10.1155/2021/2565488

[38] A. Mohammed and R. Kora, "An effective ensemble deep learning framework for text classification," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 8825–8837, 2022. doi: 10.1016/j.jksuci.2021.11.001

[39] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, 27, 2019. doi: 10.1186/s40537-019-0192-5

[40] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018. doi: 10.1016/j.neunet.2018.07.011

[41] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd Ed. Sebastopol, CA: O'Reilly Media, 2019.

[42] J. T. Hancock, T. M. Khoshgoftaar, and J. M. Johnson, "Evaluating classifier performance with highly imbalanced big data," *Journal of Big Data*, vol. 10, no. 1, 42, 2023. doi: 10.1186/s40537-023-00724-5

[43] T. Islam, T. Mazumder, M. N. S. Roni *et al.*, "A comparative study of machine learning models for predicting Aman rice yields in Bangladesh," *Heliyon*, vol. 10, no. 23, 2024. doi: 10.1016/j.heliyon.2024.e40764

[44] R. Sebastian, L. Yuxi, and V. Mirjalili, *Machine Learning with PyTorch and Scikit-Learn*, Birmingham, UK: Packt Publishing, 2022.

[45] S. Yao, A. Kronenburg, A. Shamooni *et al.*, "Gradient boosted decision trees for combustion chemistry integration," *Applications in Energy and Combustion Science*, vol. 11, 100077, 2022. doi: 10.1016/j.jaecs.2022.100077

[46] E. Richardson, R. Trevizani, J. A. Greenbaum *et al.*, "The receiver operating characteristic curve accurately assesses imbalanced datasets," *Patterns*, vol. 5, no. 6, 2024. doi: 10.1016/j.patter.2024.100994

[47] S. U. Hassan, J. Ahamed, and K. Ahmad, "Analytics of machine learning-based algorithms for text classification," *Sustainable Operations and Computers*, vol. 3, pp. 238–248, 2022. doi: 10.1016/j.susoc.2022.03.001

[48] A. P. M. D. Rosa and C. V. Rosales, "Sentiment analysis on the implementation of limited face-to-face classes using Naïve Bayes algorithm: A student's post-pandemic perspective," *Journal of Engineering Science and Technology*, vol. 20, no. 1, pp. 195–208, 2025.

[49] J. Jeslin, A. Radhika, and M. Haripriya, "A comparative analysis of probabilistic and machine learning models for biological sequence classification," *JP Journal of Biostatistics*, vol. 25, no. 2, pp. 273–294, 2025. doi: 10.17654/0973514325014