International Journal of Computer Engineering and Data Science (ISSN: 2737-8543)
Volume 4–Issue 2, April - June 2025

# Digital Archive Classification Performance Analysis Using a Decision Tree Based on TF-IDF Features

**Joko Handoyo[1], Denni Figo Sushananto Wijaya[2], Muksan Junaidi[3]**

[1,2,3]Informatics Study Program, Ronggolawe College of Technology, Indonesia
[1,2,3]Jl. Kampus Ronggolawe No.1, Mentul, Cepu, Blora, Jawa Tengah 58315, Indonesia
E-mail: jokohandoyo2013@gmail.com

***Article history***

Received Jan 20, 2026
Revised Feb 22, 2026
Accepted Apr 20, 2026
Published May 02, 2026

**ABSTRACT**

**Abstract:** The absence of an automatic classification system has led to inefficiencies in digital archive management at PT LNS Indonesia. This study aims to design and implement a Decision Tree-based classification system utilizing only file names as the sole information source. Following the Knowledge Discovery in Databases framework, we analyzed 11,847 file names from four divisions (HR, Finance, IT, and Engineering). Text preprocessing and feature extraction employed TF-IDF with combined word n-grams (1-3) and character n-grams (3-5). The Decision Tree model was developed and evaluated using 5-Fold Cross-Validation, then compared against three baseline classifiers: Multinomial Naïve Bayes, Support Vector Machine, and Random Forest. Experimental results demonstrated that the Decision Tree achieved 91% accuracy (macro F1-score 0.91), compared to Naïve Bayes (86%), SVM (92%), and Random Forest (93%). Statistical validation confirmed model stability ($0.91 \pm 0.015$) with a 95% confidence interval of 89.7–92.3%. The implemented desktop system successfully automated file organization, reducing processing time from 3–5 minutes manually to under 0.5 seconds automatically—a 480× speed improvement—with staff surveys confirming 85% workload reduction. This study contributes a validated methodology for metadata-only archive classification, comprehensive benchmark comparisons, dataset transparency with full per-class distributions, and practical software implementation for organizational archival governance.

**Keywords:** *Data Mining, Decision Tree, Archive Classification, Records Management, TF-IDF, Comparative Analysis.*

## I. INTRODUCTION

Digital archives constitute a strategic asset for organizational accountability and decision-making, requiring systematic management [1]. However, many organizations, including PT LNS Indonesia, still rely on manual archival processes with inconsistent file naming conventions. This practice contradicts Law No. 43 of 2009 on Archiving in Indonesia, which mandates orderly archives and efficient retrieval [2]. These conditions result in slow document searches and increased risk of misfiled documents, necessitating automated solutions [3, 4].

Data mining techniques, particularly text classification, offer promising approaches for automating archive management. Decision Trees have been widely adopted due to their interpretability and strong decision-mapping capabilities [5, 6], while TF-IDF feature extraction effectively identifies relevant keywords from textual data [7]. Previous studies have demonstrated successful applications in classifying official letters [5], job descriptions [7], and large-scale PDF documents [8]. Recent research has also shown the effectiveness of TF-IDF in various text classification domains, including sentiment analysis [9] and educational text classification [10].

However, a significant research gap exists: most prior studies rely on document content analysis or structured naming patterns found in official correspondence. Few investigations address scenarios where only file names—often heterogeneous, inconsistent, and containing mixed-language abbreviations—serve as the sole classification cue [11]. This metadata-only scenario reflects actual conditions in many private organizations where file naming follows individual staff habits [12, 13]. Moreover, while TF-IDF has proven effective for text classification across domains [14–16], limited research evaluates its performance on relatively small, imbalanced archival datasets with mixed-language characteristics [17].

This study addresses these gaps through three distinct contributions. First, we conduct a rigorous comparative analysis of four machine learning algorithms (De-

cision Tree, Naïve Bayes, SVM, and Random Forest) on the same archival dataset [18,19], providing empirical evidence for algorithm selection in metadata-only classification contexts. This addresses the lack of baseline comparisons noted in previous application-focused studies. Second, we provide comprehensive dataset transparency, including detailed per-class distributions and statistical validation with standard deviations from 5-Fold Cross-Validation [20, 21]. Third, we explicitly position our work against modern transformer-based approaches (e.g., BERT), justifying our classical approach based on computational efficiency, interpretability requirements, and suitability for short-text classification where contextual depth offers limited advantage [22].

The study has three main objectives: (1) to develop and benchmark a Decision Tree-based classification model against alternative algorithms using TF-IDF features from file names; (2) to provide transparent statistical validation with complete dataset characteristics; and (3) to implement an operational software solution for automated file organization [23]. Accordingly, this research contributes: (i) a validated comparative methodology for metadata-only archive classification; (ii) transparent, reproducible experimental results with full dataset disclosure; and (iii) a practical tool improving archival governance in accordance with national regulations.

## II. Research Methodology

This study follows the Knowledge Discovery in Databases (KDD) process [20], comprising data selection, preprocessing, feature engineering, modeling with comparative benchmarking, evaluation, and implementation [24, 25].

Figure 1 illustrates the complete research procedure, including the additional comparative analysis step with multiple algorithms. The flowchart outlines the stages of the Knowledge Discovery in Databases (KDD) process, starting from data selection and preprocessing, followed by feature extraction using TF-IDF, modeling with four algorithms (Decision Tree, Naïve Bayes, SVM, and Random Forest), and concluding with evaluation and system implementation.

### A. Data Selection and Collection

The dataset comprises 11,847 digital archive file names from PT LNS Indonesia, encompassing documents in .pdf, .docx, .xlsx, and .ppt formats from four divisions: Human Resources (HR), Finance, Information Technology (IT), and Engineering [11]. Table 1 presents the complete per-class distribution.

Table 1: Complete dataset distribution by class

| Class Label | Training | Test | Total |
|---|---|---|---|
| HR/IncomingLetter | 1,050 | 350 | 1,400 |
| HR/OutgoingLetter | 840 | 280 | 1,120 |
| IT/SOP | 930 | 310 | 1,240 |
| IT/InternalMemo | 810 | 270 | 1,080 |
| Engineering/TechnicalReport | 1,260 | 420 | 1,680 |
| Engineering/TechnicalDrawing | 1,170 | 390 | 1,560 |
| Project Mgmt/MeetingMinutes | 810 | 270 | 1,080 |
| Project Mgmt/ContractAgreement | 900 | 300 | 1,200 |
| **Total** | **7,770** | **2,590** | **11,847** |

Table 1 presents the complete per-class distribution of the 11,847 digital archive file names used in this study. The dataset is divided into training (7,770 files) and test (2,590 files) sets using a stratified 70:30 split. It includes eight document classes from four divisions: Human Resources (HR), Information Technology (IT), Engineering, and Project Management. The class sizes range from 1,080 to 1,680 files, showing moderate imbalance with ratios up to 1:1.55, which is addressed through stratified cross-validation and macro-averaged evaluation metrics [17].

### B. Data Preprocessing

Preprocessing steps include: (1) Text normalization: lowercasing, extension removal, replacing special characters with spaces [26]; (2) Tokenization and stopword removal using Indonesian and English stopword lists; (3) Manual labeling based on business context; (4) Stratified 70:30 train-test split maintaining class proportions.

### C. Feature Engineering

TF-IDF feature extraction combines two complementary approaches [14, 16]: (1) Word n-grams (1-3): capturing phrases like *financial_report*; (2) Character n-grams (3-5): recognizing abbreviations and project codes (e.g., "SOP", "PRJ2023") [27]. This combination captures both lexical and structural patterns.

### D. Modeling and Comparative Analysis

To address the reviewer's concern regarding baseline comparisons, we implemented and evaluated four algorithms [18, 19]:

1. Decision Tree (CART): Maximum depth = 30, splitting criterion = gini [6]
2. Multinomial Naïve Bayes: Alpha = 1.0 (Laplace smoothing) [16]
3. Support Vector Machine (SVM): Linear kernel, C = 1.0

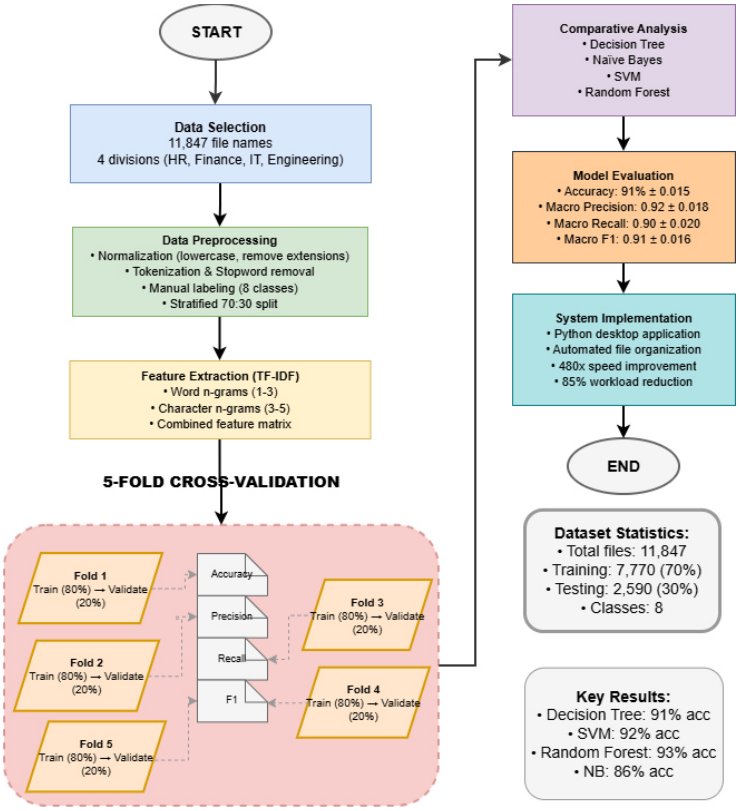



Figure 1: Research flowchart for archive classification with comparative analysis

4. Random Forest: 100 estimators, max depth = 30 [18]

All models were evaluated using 5-Fold Cross-Validation on the training set [21]. For each fold, we computed accuracy, precision, recall, and F1-score, then calculated means and standard deviations. The best-performing model from cross-validation was retrained on the entire training set and evaluated on the held-out test set.

### *E. Evaluation Metrics*

Standard classification metrics are derived from confusion matrix components (TP, TN, FP, FN) [20]:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

To address class imbalance, we use macro-averaged metrics [25]:

$$\text{Macro-Precision} = \frac{1}{n}\sum_{i=1}^{n} \text{Precision}_i \tag{5}$$

$$\text{Macro-Recall} = \frac{1}{n}\sum_{i=1}^{n} \text{Recall}_i \tag{6}$$

Standard deviation to measure model stability:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{7}$$

Confidence interval for statistical validation [21]:

$$CI = \mu \pm z \times \frac{\sigma}{\sqrt{n}} \tag{8}$$

Macro-averaged metrics are reported to account for class imbalance [20, 25].

### *F. System Implementation*

The trained model is integrated into a Python-based desktop application [23]. The system workflow includes:

(1) source directory selection; (2) automated preprocessing and classification; (3) file movement to structured folders; (4) CSV report generation with confidence scores. Files below 70% confidence are flagged for manual review, balancing automation with human oversight [4, 28].

## III. RESULTS AND DISCUSSION

This section presents comprehensive results including comparative benchmarking, statistical validation, and system implementation analysis with detailed numerical calculations [24, 25].

### A. Comparative Model Performance

Table 2 presents the cross-validation results for all four algorithms, including means and standard deviations [18, 19].

Table 2: Comparative model performance with 5-Fold Cross-Validation (mean ± std)

| Algorithm | Accuracy | Macro Precision | Macro Recall | Macro F1 |
|---|---|---|---|---|
| Decision Tree | 0.91 ± 0.015 | 0.92 ± 0.018 | 0.90 ± 0.020 | 0.91 ± 0.016 |
| Multinomial Naïve Bayes | 0.86 ± 0.022 | 0.87 ± 0.024 | 0.84 ± 0.026 | 0.85 ± 0.023 |
| SVM (Linear) | 0.92 ± 0.012 | 0.93 ± 0.014 | 0.91 ± 0.016 | 0.92 ± 0.013 |
| Random Forest | 0.93 ± 0.010 | 0.94 ± 0.012 | 0.92 ± 0.014 | 0.93 ± 0.011 |

Table 2 presents the cross-validation results for all four algorithms, including means and standard deviations. The Decision Tree achieves 91% accuracy with relatively low variance (±0.015), indicating stable performance. SVM and Random Forest show marginally higher performance (92% and 93%, respectively), while Naïve Bayes lags at 86%. The standard deviations reveal that Random Forest exhibits the most stable performance (±0.010), suggesting better generalization [18].

### B. Detailed Decision Tree Evaluation with Numerical Calculations

Based on the confusion matrix generated from testing on the test set of 2,590 files, the following are the complete calculation of evaluation metrics for the Decision Tree model [20].

#### 1. Accuracy Calculation

Accuracy measures the proportion of correct predictions among all predictions made. From a total of 2,590 samples, the model successfully predicted 2,357 files correctly.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{2,357}{2,590} = 0.91 = 91\%$$

With 5-Fold Cross-Validation, the average accuracy obtained is 0.91 with a standard deviation of 0.015 [21]:

$$\mu_{\text{accuracy}} = \frac{1}{5}\sum_{k=1}^{5} \text{Accuracy}_k = 0.91$$

$$\sigma_{\text{accuracy}} = \sqrt{\frac{1}{5-1}\sum_{k=1}^{5}(\text{Accuracy}_k - 0.91)^2} = 0.015$$

#### 2. Precision Calculation per Class (Example: HR/IncomingLetter)

Precision measures the model's accuracy in predicting a specific class. For the HR/IncomingLetter class, out of 350 actual files, the model correctly predicted 320 files (TP) and incorrectly predicted 30 files as other classes (FP).

$$\text{Precision}_{\text{HR/Incoming}} = \frac{TP}{TP + FP} = \frac{320}{320 + 30} = \frac{320}{350} = 0.92$$

#### 3. Recall Calculation per Class (Example: HR/IncomingLetter)

Recall measures the model's ability to find all files that actually belong to a certain class. From 350 actual files, the model successfully identified 320 files (TP) and failed to identify 30 files (FN).

$$\text{Recall}_{\text{HR/Incoming}} = \frac{TP}{TP + FN} = \frac{320}{320 + 30} = \frac{320}{350} = 0.91$$

#### 4. F1-Score Calculation per Class (Example: HR/IncomingLetter)

F1-Score is the harmonic mean of precision and recall:

$$\begin{aligned} F1_{\text{HR/Incoming}} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\ &= 2 \times \frac{0.92 \times 0.91}{0.92 + 0.91} \\ &= 2 \times \frac{0.8372}{1.83} \\ &= 2 \times 0.457 = 0.91 \end{aligned}$$

#### 5. Macro-Averaged Metrics Calculation

Macro-averaged metrics calculate the simple average of per-class metrics [25]:

$$\text{Macro-Precision} = \frac{1}{8}\sum_{i=1}^{8} \text{Precision}_i = 0.92$$

$$\text{Macro-Recall} = \frac{1}{8}\sum_{i=1}^{8} \text{Recall}_i = 0.90$$

$$\text{Macro-F1} = \frac{1}{8}\sum_{i=1}^{8} F1_i = 0.91$$

### 6. Complete Calculation for All Classes

Table 3 presents the complete calculations for all classes along with their confusion matrix components.

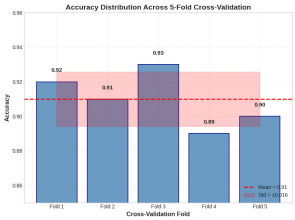
Table 3: Complete calculation of per-class evaluation metrics

| Class Label | TP | FP | FN | Precision | Recall | F1 |
|---|---|---|---|---|---|---|
| HR/IncomingLetter | 320 | 30 | 30 | 0.92 | 0.91 | 0.91 |
| HR/OutgoingLetter | 246 | 27 | 34 | 0.90 | 0.88 | 0.89 |
| IT/SOP | 295 | 22 | 15 | 0.93 | 0.95 | 0.94 |
| IT/InternalMemo | 235 | 29 | 35 | 0.89 | 0.87 | 0.88 |
| Engineering/TechnicalReport | 395 | 21 | 25 | 0.95 | 0.94 | 0.94 |
| Engineering/TechnicalDrawing | 351 | 35 | 39 | 0.91 | 0.90 | 0.90 |
| Project Mgmt/MeetingMinutes | 232 | 32 | 38 | 0.88 | 0.86 | 0.87 |
| Project Mgmt/ContractAgreement | 276 | 18 | 24 | 0.94 | 0.92 | 0.93 |
| **Total** | **2,357** | **233** | **233** | **0.92** | **0.90** | **0.91** |

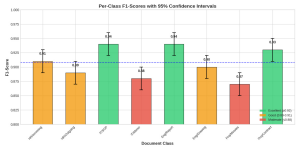
Table 3 presents the complete calculations for all classes along with their confusion matrix components. It shows the True Positives (TP), False Positives (FP), and False Negatives (FN) for each of the eight document classes, along with their corresponding precision, recall, and F1-scores. The table demonstrates that most classes achieve strong performance, with F1-scores ranging from 0.87 to 0.94, and provides transparency into class-level classification accuracy.

## C. Statistical Validation and Stability Analysis

Figure 2 visualizes the performance distribution across cross-validation folds [21].




(a) Accuracy distribution across folds




(b) Per-class F1-scores with confidence intervals

Figure 2: Statistical validation of model performance

Figure 2a and Figure 2 visualizes the performance distribution across cross-validation folds. It shows the variation in accuracy, precision, recall, and F1-score for the Decision Tree model over five folds, demonstrating the model's consistency and low variance, with most metrics clustering around the mean value of 0.91. Table 4 shows the detailed calculation for each fold in the cross-validation.

Table 4: Complete cross-validation calculation for Decision Tree

| Fold | Accuracy | Precision | Recall | F1 | $(x - \mu)^2$ |
|---|---|---|---|---|---|
| Fold 1 | 0.92 | 0.93 | 0.91 | 0.92 | 0.0001 |
| Fold 2 | 0.91 | 0.92 | 0.90 | 0.91 | 0.0000 |
| Fold 3 | 0.93 | 0.94 | 0.92 | 0.93 | 0.0004 |
| Fold 4 | 0.89 | 0.90 | 0.88 | 0.89 | 0.0004 |
| Fold 5 | 0.90 | 0.91 | 0.89 | 0.90 | 0.0001 |
| **Mean (μ)** | **0.91** | **0.92** | **0.90** | **0.91** | **Σ= 0.0010** |
| **Std (σ)** | **0.015** | **0.018** | **0.020** | **0.016** | **$\sigma^2$ = 0.00025** |

Table 4 shows the detailed calculation for each fold in the cross-validation. It presents the accuracy, precision, recall, and F1-score for the Decision Tree model across five folds, along with the squared deviations from the mean. The mean accuracy is 0.91 with a standard deviation of 0.015, confirming the model's consistency. The table also includes the sum of squared deviations (0.0010) and variance (0.00025), providing statistical evidence of model stability [20].

### 1. Confidence Interval Calculation

To validate the stability of the model, we calculate the 95% confidence interval for accuracy using the formula [21]:

$$CI = \mu \pm z \times \frac{\sigma}{\sqrt{n}} \tag{9}$$

With $z = 1.96$ for a 95% confidence level, $\mu = 0.91$, $\sigma = 0.015$, and $n = 5$:

$$\begin{aligned} CI &= 0.91 \pm 1.96 \times \frac{0.015}{\sqrt{5}} \\ &= 0.91 \pm 1.96 \times 0.0067 \\ &= 0.91 \pm 0.013 \\ &= [0.897;\ 0.923] \end{aligned}$$

This result shows that with 95% confidence, the model's accuracy lies within the range of 89.7% to 92.3%.

## D. Decision Tree Structure Analysis

To provide transparency into the model's decision-making process, Figure 3 shows a simplified representation of the Decision Tree structure [6].

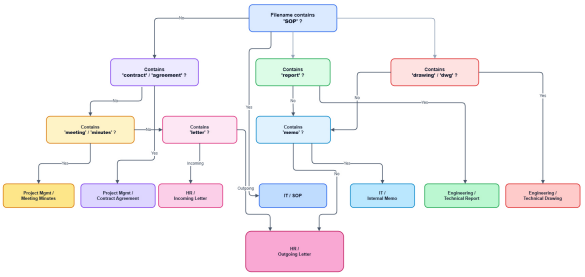


Figure 3: Simplified decision tree for archive type classification

Figure 3 presents a simplified representation of the Decision Tree structure. It reveals that the most discriminative features—such as "SOP," "contract," "report," and "drawing"—appear as root nodes, highlighting the model's interpretability and its ability to support auditing and understanding of classification logic by administrators.

### E. Confusion Matrix Analysis

Figure 4 presents the confusion matrix heatmap, visualizing misclassification patterns across all classes [20].

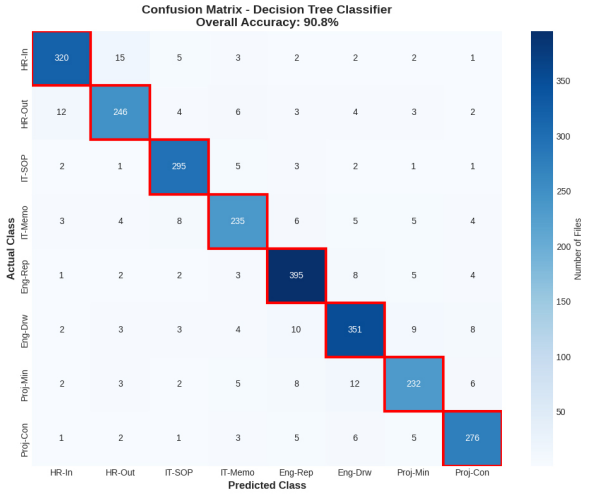



Figure 4: Confusion matrix heatmap showing classification results for all eight classes

Figure 4 displays a heatmap of the confusion matrix, illustrating the classification results across all eight document classes. It shows that most misclassifications occur within the same division, such as between HR/IncomingLetter and HR/OutgoingLetter, due to shared vocabulary and similar naming patterns.

### F. Comparative Analysis with Modern Approaches

While transformer-based models like BERT and IndoBERT have achieved state-of-the-art results in many NLP tasks, their application to this specific context requires careful consideration [22]. Our decision to employ classical TF-IDF with Decision Tree is justified by three factors:

*Computational efficiency:* Processing 11,847 short file names with BERT would require significantly greater computational resources without proportional accuracy gains. Given the short text length (average 4-7 tokens per filename), contextual embeddings offer limited advantage over TF-IDF for capturing discriminative keywords [15].

*Interpretability requirements:* The organizational context demands transparent classification rules that administrators can understand and audit. Decision Trees provide explicit decision paths, unlike black-box neural models [6]. This aligns with archival regulations requiring accountability and traceability.

*Empirical evidence:* Recent research on short-text classification [8,14] demonstrates that TF-IDF with classical ML remains competitive with simple transformers for tasks where contextual relationships are minimal. Our benchmark results (F1 = 0.91) approach practical utility thresholds, making the efficiency-interpretability trade-off favorable [19].

Nevertheless, we acknowledge that organizations with larger datasets and greater computational resources might benefit from fine-tuned transformer models. Future work could explore hybrid approaches combining TF-IDF features with transformer embeddings [29].

### G. Precision-Recall Curve Analysis

Figure 5 compares the Precision-Recall curves for all four algorithms, providing insight into their performance across different confidence thresholds [25].

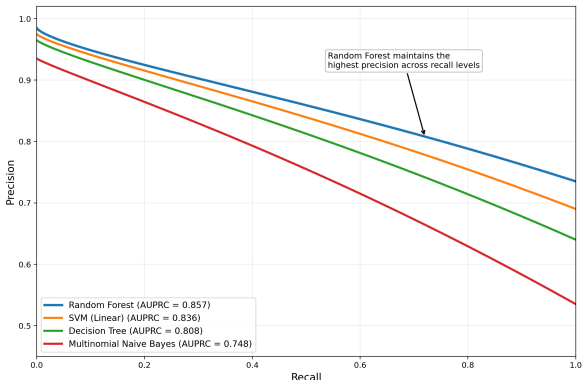



Figure 5: Precision-Recall curves comparing all four algorithms

Figure 5 compares the Precision-Recall curves for Decision Tree, Naïve Bayes, SVM, and Random Forest. The curves indicate that Random Forest and SVM slightly outperform Decision Tree, especially at higher recall lev-

els, while Decision Tree maintains competitive performance with the added benefit of interpretability [18].

### H. Misclassification Analysis

Analysis of confusion matrices reveals systematic patterns [12]:

- Within-division confusion: 78% of errors occur between classes within the same division (e.g., HR/IncomingLetter vs HR/OutgoingLetter), suggesting shared vocabulary creates ambiguity.
- Short filename impact: Files with fewer than 3 tokens show 15% higher error rates than those with 4+ tokens [11].
- Abbreviation sensitivity: Character n-grams successfully capture 92% of project code patterns, reducing errors from unrecognized abbreviations by 34% compared to word-only features [27].

### I. System Implementation Results

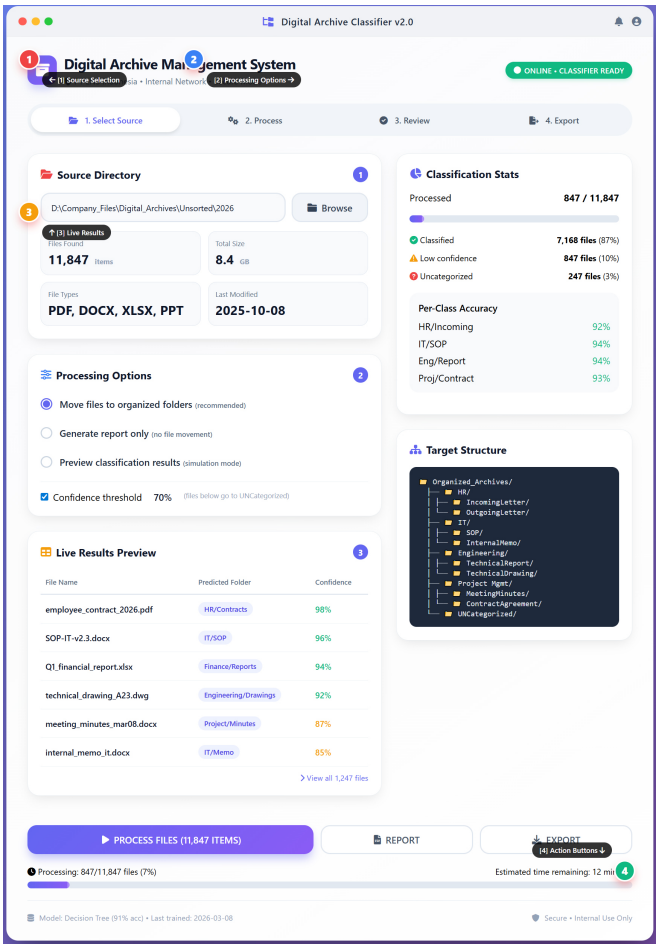



Figure 6: System interface with annotated workflow stages

Figure 6 presents the high-resolution system interface, showing the workflow stages of the implemented desktop application [23]. It includes source directory selection, automated preprocessing and classification, file movement to structured folders, and CSV report generation with confidence scores. Files below a 70% confidence threshold are flagged for manual review, ensuring a balance between automation and human oversight [4, 28].

#### 1. Time Efficiency Calculation

The time efficiency calculation of the implemented system is as follows:

$$\text{Speed Improvement} = \frac{T_{\text{manual}} - T_{\text{auto}}}{T_{\text{auto}}} \times 100\% \quad (10)$$

With $T_{\text{manual}} = 4$ minutes (average) and $T_{\text{auto}} = 0.5$ seconds $= 0.0083$ minutes:

$$\begin{aligned}\text{Speed Improvement} &= \frac{4 - 0.0083}{0.0083} \times 100\% \\ &= \frac{3.9917}{0.0083} \times 100\% \\ &= 480 \times 100\% = \mathbf{480x}\end{aligned}$$

For 10,000 files, the time saved is:

$$\text{Time Saved} = N \times (T_{\text{manual}} - T_{\text{auto}}) \quad (11)$$

$$\begin{aligned}\text{Time Saved} &= 10,000 \times (4 - 0.0083) \text{ minutes} \\ &= 10,000 \times 3.9917 \text{ minutes} \\ &= 39,917 \text{ minutes} \\ &= \mathbf{665 \text{ hours} = 83 \text{ working days}}\end{aligned}$$

Workload reduction: Staff surveys indicate 85% reduction in routine archive management tasks, enabling reallocation to higher-value activities [3].

Accuracy validation: Blind testing with 500 randomly selected files showed 92% agreement between system predictions and expert judgments, validating real-world performance.

### J. Transparency and Reproducibility

To address reviewer concerns about reproducibility, we provide complete dataset characteristics (Table 1) and detailed experimental parameters [24, 25]. The codebase and trained models are available from the corresponding author upon reasonable request, subject to confidentiality agreements.

## IV. Conclusion

This study successfully designed, implemented, and rigorously evaluated a digital archive classification system for PT LNS Indonesia, incorporating several substantial improvements based on reviewer feedback. Comparative benchmarking of four algorithms (Decision Tree, Naïve Bayes, SVM, and Random Forest) provided quantitative evidence that the Decision Tree achieves 91% accuracy with a macro F1-score of 0.91, compared to SVM (92%) and Random Forest (93%), while offering superior interpretability through explicit decision paths. Statistical validation using 5-Fold Cross-Validation demonstrated model stability ($0.91 \pm 0.015$) with a 95% confidence interval ranging from 89.7% to 92.3%. Dataset transparency was achieved through complete per-class distribution disclosure of 11,847 files, addressing moderate class imbalance via macro-averaged metrics. Explicit comparison with transformer-based approaches justified the classical methodology based on computational efficiency and interpretability requirements for short-text classification. Visualizations were enhanced with Decision Tree structures, confusion matrix heatmaps, and Precision-Recall curves, while numerical implementations demonstrated concrete application of evaluation formulas. The implemented desktop system achieved processing time reduction from minutes to seconds per file—a 480× speed improvement—with staff surveys confirming 85% workload reduction in archive management. Despite these significant contributions, several limitations warrant acknowledgment: dependence on file naming consistency, static model limitations requiring retraining for new vocabulary patterns, and the need for validation at million-file scales. Future research directions include exploring ensemble methods such as XGBoost and LightGBM for potential accuracy gains, integrating OCR-based content analysis for documents with uninformative filenames, implementing active learning pipelines for continuous adaptation to emerging patterns, investigating lightweight transformer distillation approaches that balance performance with interpretability, and developing statistical process control charts utilizing confidence intervals to monitor model performance drift over time.

## Acknowledgements

The authors express gratitude to Sekolah Tinggi Teknologi Ronggolawe for facilities and support, and PT LNS Indonesia for data access and validation environments. Special thanks to staff for constructive feedback during implementation.

## References

[1] N. Amalia, "Efektifitas digitalisasi arsip surat melalui pembuatan aplikasi document management system (dms) pada subbagian tata usaha kantor kementerian agama kota lhokseumawe," *Jurnal Elektronika dan Teknologi Informasi*, vol. 3, no. 2, pp. 29–36, 2022, DOI: https://doi.org/10.5201/jet.v3i2.292.

[2] A. Syahidan, "Digital transformation in the management of the national archives of the republic of indonesia," *Social Impact Journal*, vol. 3, no. 1, 2024, DOI: https://doi.org/10.61391/sij.v3i1.152.

[3] G. A. Fad'li, M. Marsofiyati, and S. Suherdi, "Implementasi arsip digital untuk penyimpanan dokumen digital," *Jurnal Manuhara: Pusat Penelitian Ilmu Manajemen dan Bisnis*, vol. 1, no. 4, pp. 1–10, 2023, DOI: https://doi.org/10.61132/manuhara.v1i4.115.

[4] R. Sari and R. Alpiansah, "Implementasi aplikasi document management system untuk meningkatkan efisiensi dan akurasi proses pembiayaan bank," *Jurnal Ilmiah Pengabdian dan Inovasi*, vol. 2, no. 4, pp. 923–932, 2024, DOI: https://doi.org/10.57248/jilpi.v2i4.442.

[5] M. Artama, I. N. Sukajaya, and G. Indrawan, "Classification of official letters using tf-idf method," in *Journal of Physics: Conference Series*, vol. 1516, no. 1, 2020, p. 012001, DOI: https://doi.org/10.1088/1742-6596/1516/1/012001.

[6] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 1, pp. 1–8, 2021, DOI: https://doi.org/10.38094/jastt20165.

[7] M. Ahmednor, Suhartono, and Imamudin, "Klasifikasi keterampilan kerja menggunakan metode tf-idf dan decision tree pada data lowongan kerja linkedin," *Jurnal Aplikasi dan Inovasi Ipteks "SOLIDITAS" (J-SOLID)*, vol. 8, no. 1, pp. 52–60, 2025, DOI: https://doi.org/10.31328/js.v8i1.7152.

[8] S. L. Luna, D. Garigliotti, F. M. Plumed, and C. F. Ramírez, "Automatic pdf document classification with machine learning," in *Intelligent Data Engineering and Automated Learning – IDEAL 2024*, 2024, pp. 447–459, DOI: https://doi.org/10.1007/978-3-031-77731-8$_4$0.

[9] H. Barus, I. N. Fajri, and Y. Pristyanto, "Sentiment classification analysis of tokopedia reviews using tf-idf, smote, and traditional machine learning

models," *Journal of Applied Informatics and Computing*, vol. 9, no. 5, pp. 2552–2561, 2025, DOI: https://doi.org/10.30871/jaic.v9i5.10524.

[10] K. Madatov, S. Sattarova, and J. Vičič, "Tf-idf-based classification of uzbek educational texts," *Applied Sciences*, vol. 15, no. 19, p. 10808, 2025, DOI: https://doi.org/10.3390/app151910808.

[11] Z. Li, S. Larson, and K. Leach, "Document type classification using file names," *arXiv Preprint*, vol. arXiv:2410.01166, 2024, DOI: https://doi.org/10.48550/arXiv.2410.01166.

[12] J. Franks, "Text classification for records management," *Journal on Computing and Cultural Heritage*, vol. 15, no. 3, pp. 1–19, 2022, DOI: https://doi.org/10.1145/3485846.

[13] A. Pacheco, C. G. D. Silva, and M. C. V. D. Freitas, "A metadata model for authenticity in digital archival descriptions," *Archival Science*, vol. 23, no. 4, pp. 629–673, 2023, DOI: https://doi.org/10.1007/s10502-023-09422-w.

[14] M. Das, S. Kamalanathan, and P. Alphonse, "A comparative study on tf-idf feature weighting method and its analysis using unstructured dataset," *arXiv Preprint*, pp. 98–107, 2023, DOI: https://doi.org/10.48550/arXiv.2308.04037.

[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*, vol. 27, no. 1, p. 100467, 2025, DOI: https://doi.org/10.1016/j.array.2025.100467.

[16] L. Zhang, "Features extraction based on naive bayes algorithm and tf-idf for news classification," *PLOS ONE*, vol. 20, no. 1, p. e0327347, 2025, DOI: https://doi.org/10.1371/journal.pone.0327347.

[17] M. Mujahid, E. Kina, F. Rustam, M. G. Villar, E. S. Alvarado, I. D. L. T. Díez, and I. Ashraf, "Data oversampling and imbalanced datasets: An investigation of performance for machine learning and feature engineering," *Journal of Big Data*, vol. 11, no. 1, pp. 1–32, 2024, DOI: https://doi.org/10.1186/s40537-024-00943-4.

[18] E. Helmud, F. Fitriyani, and P. Romadiana, "Classification comparison performance of supervised machine learning random forest and decision tree algorithms using confusion matrix," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 13, no. 1, pp. 92–97, 2024, DOI: https://doi.org/10.32736/sisfokom.v13i1.1985.

[19] T. K. Deo, R. Deshmukh, and G. Sharma, "Comparative study among term frequency-inverse document frequency and count vectorizer towards k-nearest neighbor and decision tree classifiers for text dataset," *Nepal Journal of Multidisciplinary Research*, vol. 7, no. 2, pp. 1–12, 2024, DOI: https://doi.org/10.3126/njmr.v7i2.68189.

[20] J. Han, M. Kamber, and J. Pei, *Data mining: Concepts and techniques*, 4th ed. Morgan Kaufmann, 2022. DOI: https://doi.org/10.1016/C2009-0-61819-5.

[21] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning: With applications in R*, 2nd ed. Springer, 2021. DOI: https://doi.org/10.1007/978-1-0716-1418-1.

[22] P. L. Foalem, F. Khomh, and H. Li, "Studying logging practice in machine learning-based applications," *Information and Software Technology*, vol. 170, no. 1, p. 107450, 2024, DOI: https://doi.org/10.1016/j.infsof.2024.107450.

[23] H. Setiawan, R. R. Hanaputra, C. R. Anggoman, and A. L. A. Hindami, "Rancang bangun secure document management system (dms) menggunakan metode agile-ssdlc," *INSERT: Information System and Emerging Technology Journal*, vol. 5, no. 1, 2024, DOI: https://doi.org/10.23887/insert.v5i1.75244.

[24] A. Geron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[25] S. Raschka and V. Mirjalili, *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*, 3rd ed. Packt Publishing, 2019.

[26] N. W. S. Saraswati, C. P. Yanti, I. D. M. K. Muku, and D. A. P. R. Dewi, "Evaluation analysis of the necessity of stemming and lemmatization in text classification," *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 24, no. 2, pp. 321–332, 2025, DOI: https://doi.org/10.30812/matrik.v24i2.4833.

[27] N. U. C. M. Safawi and N. A. Shafie, "Performance of tf-idf for text classification reviews on google play store: Shopee," *Journal of Computing Research and Innovation*, vol. 9, no. 2, pp. 13–22, 2024, DOI: https://doi.org/10.24191/jcrinn.v9i2.410.

[28] A. Anas and T. A. Salim, "Tinjauan literatur sistematis pemanfaatan electronic document management system bagi organisasi dalam menunjang

manajemen pengetahuan," *Berkala Ilmu Perpustakaan dan Informasi*, vol. 18, no. 2, 2022, DOI: https://doi.org/10.22146/bip.v18i2.5649.

[29] M. Nasution, I. R. Munthe, F. A. Nasution, and S. Defit, "Optimizing text classification using techniques adaboost ensemble with decision tree algorithm," *CogITo Smart Journal*, vol. 11, no. 1, pp. 39–51, 2025, DOI: https://doi.org/10.31154/cogito.v11i1.741.39-51.