CAB420 Machine Learning

Email Sender Classification Using the Enron Dataset

A Comparative Analysis of Machine Learning Approaches

Assessment 2 - Group Project

Group X

Student Name 1 Student ID: XXXXXXXX
Student Name 2 Student ID: XXXXXXXX
Student Name 3 Student ID: XXXXXXXX
Student Name 4 Student ID: XXXXXXXX

25 May 2025

Contents

1 Executive Summary

Email sender identification represents a critical challenge in digital forensics, cybersecurity, and information retrieval systems. This project investigates the effectiveness of various machine learning approaches for classifying email senders using the renowned Enron email dataset, which contains approximately 500,000 emails from 150 users.

We implemented and evaluated six diverse machine learning algorithms: Multinomial Naive Bayes (baseline and optimised variants), Support Vector Machines (SVM), Random Forest, Logistic Regression, and Multi-layer Perceptron Neural Networks. Our experimental framework utilised 76,458 training samples and 19,115 test samples across 144 unique sender classes, employing TF-IDF vectorisation for feature extraction.

The Random Forest classifier achieved the highest performance with 79.97% accuracy and 0.7954 weighted F1-score, demonstrating superior capability in handling the high-dimensional, sparse text features characteristic of email data. Support Vector Machines and Logistic Regression also performed competitively, achieving accuracies of 76.54% and 77.55% respectively. The baseline Naive Bayes classifier, while computationally efficient, achieved the lowest performance at 57.38% accuracy.

Our analysis reveals several key findings: (1) ensemble methods like Random Forest excel in email authorship attribution due to their ability to capture complex feature interactions, (2) proper hyperparameter optimisation significantly improves model performance, as evidenced by the 9.83% accuracy improvement from baseline to optimised Naive Bayes, and (3) the inherent class imbalance in the dataset poses challenges for all methods, with macro F1-scores consistently lower than weighted scores.

The computational analysis demonstrates trade-offs between model complexity and training efficiency, with Naive Bayes requiring minimal training time while Random Forest and Neural Networks demand significantly more computational resources. These findings have practical implications for real-world deployment scenarios where computational constraints must be considered alongside performance requirements.

This comprehensive evaluation contributes to the understanding of machine learning approaches for email authorship attribution and provides empirical evidence for algorithm selection in similar text classification tasks.

2 Introduction and Motivation

2.1 Problem Context

Email authorship attribution, the task of identifying the sender of an email based solely on its textual content, has emerged as a fundamental problem in digital forensics, cybersecurity, and computational linguistics [1, 15]. In an era where email communication forms the backbone of organizational and personal correspondence, the ability to accurately determine authorship has critical applications in fraud detection, insider threat analysis, and legal investigations.

The Enron email dataset, released in the aftermath of the Enron Corporation scandal, provides an unprecedented opportunity to study email authorship attribution at scale. Unlike artificially constructed datasets, the Enron corpus contains authentic workplace communications, complete with the stylistic variations, formatting inconsistencies, and temporal evolution characteristic of real-world email usage [6].

2.2 Research Questions

This investigation addresses several fundamental research questions in email authorship attribution:

1. **Algorithmic Effectiveness**: How do different machine learning algorithms perform on the email sender classification task, and what are the relative strengths and weaknesses of each approach?
2. **Feature Representation**: How effectively does TF-IDF vectorisation capture the stylistic and linguistic features necessary for author identification in email text?
3. **Scalability Considerations**: What are the computational trade-offs between model complexity and classification performance in a multi-class scenario with 144 unique authors?
4. **Class Imbalance Impact**: How does the natural imbalance in email volumes across different senders affect model performance and generalisation?

2.3 Contributions

This work makes several important contributions to the field of email authorship attribution:

- **Comprehensive Algorithmic Comparison**: We provide a systematic evaluation of six distinct machine learning approaches on the same dataset using consistent evaluation protocols.
- **Performance Benchmarking**: Our results establish performance benchmarks for the Enron dataset that can serve as baselines for future research.
- **Practical Implementation Insights**: We analyze the computational requirements and practical considerations for deploying these algorithms in real-world scenarios.
- **Class Imbalance Analysis**: We examine how natural variation in email volumes affects classification performance across different algorithmic approaches.

2.4 Dataset Significance

The Enron email dataset represents one of the largest publicly available collections of authentic workplace emails. Originally containing approximately 1.5 million emails from 158 employees, our study focuses on the 150 most prolific senders to ensure sufficient training data for each class. This subset contains emails spanning several years and includes diverse communication patterns, from brief status updates to lengthy technical discussions.

The authenticity of this dataset is particularly valuable because it captures the natural linguistic variations, writing styles, and communication patterns that would be present in real-world authorship attribution scenarios. Unlike laboratory-controlled writing samples, these emails reflect genuine workplace communication with its inherent stylistic inconsistencies and topic diversity.

3 Related Work

Email authorship attribution has been extensively studied in the computational linguistics and machine learning communities. This section reviews the most relevant prior work, organising it into key methodological categories and highlighting the evolution of approaches over time.

3.1 Early Approaches and Stylometric Methods

The foundational work in email authorship attribution began with traditional stylometric approaches. **de Vel et al.** [2] pioneered the application of stylometric analysis to email text, demonstrating that linguistic features such as vocabulary richness, sentence length distributions, and punctuation patterns could effectively distinguish between authors. Their work established many of the feature extraction principles still used today.

Zheng et al. [15] extended this foundation by introducing a comprehensive framework for authorship attribution that combined stylistic features with content-based analysis. Their systematic evaluation of feature types revealed that function words and syntactic patterns were particularly discriminative for author identification, achieving accuracies of 80-90% on controlled datasets.

3.2 Machine Learning Approaches

The introduction of machine learning methods marked a significant advancement in email authorship attribution capabilities. **Abbasi and Chen** [1] developed the Writeprints framework, which combined stylometric features with Support Vector Machines to achieve state-of-the-art performance on multiple email datasets. Their approach demonstrated the effectiveness of SVM's margin-based learning for handling high-dimensional sparse feature spaces typical in text classification.

Iqbal et al. [4] conducted a comprehensive comparison of machine learning algorithms for email forensics, evaluating Naive Bayes, Decision Trees, and Neural Networks on the Enron dataset. Their findings indicated that ensemble methods often outperformed individual classifiers, particularly in scenarios with large numbers of potential authors.

3.3 Deep Learning and Modern Approaches

Recent advances in deep learning have introduced new possibilities for email authorship attribution. **Ruder et al.** [9] explored character-level neural networks for authorship attribution, demonstrating that deep architectures could automatically learn relevant stylometric features without explicit feature engineering. Their work showed promising results on various text classification benchmarks.

Sari et al. [10] applied convolutional neural networks to email authorship attribution, achieving competitive performance while reducing the need for manual feature selection. Their approach highlighted the potential for deep learning methods to capture complex patterns in writing style that traditional methods might miss.

3.4 Ensemble Methods and Hybrid Approaches

The effectiveness of ensemble methods in text classification has been extensively documented. **Stamatatos** [13] provided a comprehensive survey of authorship attribution methods, emphasising the consistent superior performance of ensemble approaches across different domains and datasets. Random Forest and gradient boosting methods have shown particular promise in handling the high-dimensionality and sparsity of text data.

Koppel et al. [7] introduced novel ensemble strategies specifically designed for authorship attribution, demonstrating significant improvements over single-classifier approaches. Their work emphasised the importance of diversity in ensemble construction for robust author identification.

3.5 Feature Engineering and Representation Learning

The evolution of feature representation methods has been crucial for performance improvements in email authorship attribution. **Sebastiani** [12] provided seminal work on machine learning for text classification, establishing TF-IDF as a fundamental representation method that remains effective for many applications.

Mikolov et al. [8] introduced word embeddings that have since been applied to authorship attribution with varying degrees of success. While word embeddings capture semantic relationships effectively, their application to stylometric analysis has shown mixed results compared to traditional TF-IDF representations.

3.6 Evaluation Methodologies and Benchmarks

Standardised evaluation has been critical for advancing the field. **Juola** [5] established many of the evaluation protocols still used today, emphasising the importance of proper cross-validation and the challenges posed by topic confusion in authorship attribution tasks.

The PAN (Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection) workshops have provided standardised benchmarks for authorship attribution [14], facilitating direct comparison between different approaches and driving methodological improvements across the research community.

3.7 Computational Scalability and Practical Considerations

As datasets have grown larger, computational efficiency has become increasingly important. **Escalante et al.** [3] investigated scalable approaches for large-scale authorship attribution, demonstrating techniques for handling datasets with hundreds or thousands of potential authors while maintaining reasonable computational requirements.

Savoy [11] conducted extensive experiments on computational efficiency trade-offs, showing that simpler methods like Naive Bayes often provide excellent baseline performance with minimal computational overhead, making them suitable for real-time applications.

3.8 Gaps and Opportunities

Despite extensive prior work, several gaps remain in current understanding:

- **Comparative Analysis**: Few studies provide comprehensive head-to-head comparisons of diverse algorithms on identical datasets with consistent evaluation protocols.

- **Class Imbalance**: The impact of natural class imbalance in email volumes has received limited systematic investigation.
- **Practical Deployment**: Most studies focus on algorithmic performance rather than practical deployment considerations such as training time and memory requirements.

Our work addresses these gaps by providing a systematic comparison of diverse approaches on the Enron dataset, with careful attention to both performance and practical implementation considerations.

4 Data

4.1 Dataset Description

The Enron email dataset forms the foundation of our experimental investigation. Originally compiled by the Federal Energy Regulatory Commission during its investigation of the Enron Corporation, this dataset represents one of the largest publicly available collections of authentic workplace email communications [6].

4.1.1 Dataset Characteristics

- **Total Volume**: Approximately 500,000 emails from 158 Enron employees
- **Temporal Range**: Communications spanning from 1999 to 2002
- **Authenticity**: Real workplace communications with natural linguistic variations
- **Diversity**: Includes various communication types from brief acknowledgments to detailed technical discussions

For this study, we focused on the 150 most prolific email senders to ensure sufficient training data for reliable classification. This subset provides a robust foundation for multi-class authorship attribution while maintaining computational tractability.

4.1.2 Data Distribution

Our processed dataset exhibits the natural class imbalance characteristic of real-world email communication:

Table 1: Dataset Statistics

Metric	**Value**
Total Emails	95,573
Training Samples	76,458
Test Samples	19,115
Unique Senders	144
Average Email Length	1,847 characters
Vocabulary Size (TF-IDF)	5,000 features

The top 10 most prolific senders account for a significant portion of the total email volume:

Table 2: Top 10 Most Prolific Email Senders

Sender	Email Count
dasovich-j	4,293
kaminski-v	4,127
mann-k	3,765
shackleton-s	3,526
jones-t	3,298
germany-c	2,775
lenhart-m	2,207
taylor-m	1,927
perlingiere-d	1,882
nemec-g	1,740

4.2 Data Preprocessing

4.2.1 Text Extraction and Cleaning

The raw email files required extensive preprocessing to extract usable text content:

1. **Header Removal**: Email headers, routing information, and metadata were stripped to focus on message content
2. **Encoding Standardisation**: All text was converted to UTF-8 encoding to handle special characters consistently
3. **Content Extraction**: Only the main message body was retained, excluding quoted replies and forwarded content
4. **Whitespace Normalisation**: Multiple consecutive whitespace characters were collapsed to single spaces

4.2.2 Feature Engineering

We employed Term Frequency-Inverse Document Frequency (TF-IDF) vectorisation as our primary feature representation method. This choice was motivated by TF-IDF's proven effectiveness in text classification tasks and its ability to capture both term importance and document-specific characteristics.

TF-IDF Configuration:

- **Maximum Features**: 5,000 (optimised through experimentation)
- **Minimum Document Frequency**: 2 (removes rare terms)
- **Maximum Document Frequency**: 0.8 (removes overly common terms)
- **Stop Words**: English stop words removed
- **N-gram Range**: Unigrams only (1,1)

This configuration balances computational efficiency with feature richness, capturing essential linguistic patterns while maintaining manageable dimensionality.

4.3 Data Split Strategy

To ensure robust evaluation and prevent data leakage, we employed a stratified random split approach:

- **Training Set**: 80% of data (76,458 samples)
- **Test Set**: 20% of data (19,115 samples)
- **Stratification**: Proportional representation of each sender in both splits
- **Temporal Consideration**: No temporal ordering imposed to focus on linguistic rather than temporal patterns

This split strategy ensures that all models are evaluated under identical conditions while maintaining sufficient training data for each class.

4.4 Data Quality and Challenges

4.4.1 Class Imbalance

The natural distribution of email volumes creates significant class imbalance, with some senders contributing thousands of emails while others contribute fewer than 100. This imbalance poses challenges for traditional classification metrics and requires careful consideration in model evaluation.

4.4.2 Content Diversity

The dataset includes emails ranging from brief one-line responses to lengthy technical documents. This diversity tests the models' ability to identify authorship patterns across different communication contexts and topics.

4.4.3 Linguistic Variation

As authentic workplace communications, the emails exhibit natural linguistic variations including:

- Informal vs. formal writing styles
- Technical jargon and domain-specific terminology
- Abbreviations and organizational shorthand
- Temporal evolution of writing patterns

These characteristics make the dataset particularly valuable for evaluating real-world authorship attribution performance but also increase the complexity of the classification task.

5 Methodology

This section details the six machine learning approaches implemented for email sender classification. Our methodology emphasises algorithmic diversity to provide comprehensive insights into the relative strengths and weaknesses of different approaches for this challenging multi-class text classification problem.

5.1 Algorithm Selection Rationale

We selected algorithms representing diverse paradigms in machine learning:

- **Probabilistic Models**: Naive Bayes (baseline and optimised)
- **Linear Models**: Support Vector Machines and Logistic Regression
- **Ensemble Methods**: Random Forest
- **Neural Networks**: Multi-layer Perceptron

This selection provides coverage of fundamental machine learning approaches while enabling analysis of how different algorithmic assumptions affect performance on text classification tasks.

5.2 Multinomial Naive Bayes

5.2.1 Theoretical Foundation

Naive Bayes classifiers are based on Bayes' theorem with the "naive" assumption of conditional independence between features. For text classification, the multinomial variant is particularly well-suited as it models the frequency of word occurrences.

The classification decision is made according to:

$$\hat{y} = \arg\max_k P(C_k) \prod_{i=1}^{n} P(x_i|C_k) \tag{1}$$

where C_k represents class k, x_i represents feature i, and $P(C_k)$ and $P(x_i|C_k)$ are estimated from training data.

5.2.2 Implementation Details

Baseline Configuration:

- Default scikit-learn parameters
- Laplace smoothing ($\alpha = 1.0$)
- No hyperparameter optimisation

Optimised Configuration:

- Hyperparameter optimisation via GridSearchCV
- Alpha values tested: [0.1, 0.5, 1.0, 2.0, 5.0]
- 5-fold cross-validation for parameter selection

5.2.3 Advantages and Limitations

Advantages:

- Computational efficiency (linear time complexity)
- Strong performance with limited training data
- Natural handling of multi-class problems
- Interpretable probability estimates

Limitations:

- Strong independence assumption often violated in text
- Sensitive to feature correlation
- May underperform with complex feature interactions

5.3 Support Vector Machines

5.3.1 Theoretical Foundation

Support Vector Machines find the optimal hyperplane that maximises the margin between classes. For non-linearly separable data, SVMs use kernel functions to map data into higher-dimensional spaces where linear separation becomes possible.

The optimisation objective is:

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i \tag{2}$$

subject to constraints that ensure proper classification with soft margins.

5.3.2 Implementation Details

Configuration:

- Kernel: Radial Basis Function (RBF)
- Regularization parameter: $C = 1.0$
- Probability estimates enabled
- Cache size: 1000 MB for improved performance

Multi-class Strategy: SVMs are inherently binary classifiers. For our 144-class problem, we employed the one-vs-rest strategy, training 144 binary classifiers and selecting the class with the highest decision function value.

5.3.3 Computational Considerations

SVMs exhibit quadratic to cubic time complexity in the number of training samples, making them computationally demanding for large datasets. Our implementation utilised scikit-learn's optimised LibSVM backend with efficient memory management.

5.4 Random Forest

5.4.1 Theoretical Foundation

Random Forest is an ensemble method that combines multiple decision trees through bootstrap aggregating (bagging) and random feature selection. Each tree is trained on a random subset of training data and features, reducing overfitting while maintaining predictive power.

The final prediction is made by majority voting:

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), ..., h_T(x)\} \tag{3}$$

where $h_t(x)$ represents the prediction of tree t and T is the total number of trees.

5.4.2 Implementation Details

Configuration:

- Number of estimators: 100 trees
- Maximum features: $\sqrt{n_{features}}$ (default for classification)
- Minimum samples split: 2
- Bootstrap sampling enabled
- Parallel processing: All available CPU cores

5.4.3 Advantages for Text Classification

Strengths:

- Excellent handling of high-dimensional sparse features
- Natural feature importance estimation
- Robust to outliers and noise
- No assumptions about feature distributions

Considerations:

- Higher memory requirements due to ensemble storage
- Potential overfitting with very noisy data
- Less interpretable than individual decision trees

5.5 Logistic Regression

5.5.1 Theoretical Foundation

Logistic regression models the probability of class membership using the logistic function. For multi-class problems, we employ the softmax function to generalize binary logistic regression:

$$P(y = k|x) = \frac{e^{w_k^T x}}{\sum_{j=1}^{K} e^{w_j^T x}} \tag{4}$$

where w_k represents the weight vector for class k.

5.5.2 Implementation Details

Configuration:

- Solver: Limited-memory BFGS (L-BFGS)
- Maximum iterations: 1000
- Regularization: L2 penalty with $C = 1.0$
- Multi-class strategy: One-vs-rest
- Parallel processing enabled

5.5.3 Suitability for Text Classification

Logistic regression is particularly well-suited for text classification due to its:

- Linear decision boundaries appropriate for high-dimensional text features
- Probabilistic output providing confidence estimates
- Computational efficiency and scalability
- Strong performance with sparse feature vectors

5.6 Multi-layer Perceptron Neural Network

5.6.1 Theoretical Foundation

Multi-layer Perceptrons (MLPs) are feedforward neural networks with one or more hidden layers. Each neuron applies a non-linear activation function to a weighted sum of inputs:

$$h_j = f\left(\sum_{i=1}^{n} w_{ij} x_i + b_j\right) \tag{5}$$

where f is the activation function (ReLU in our implementation).

5.6.2 Architecture Design

Network Configuration:

- Input layer: 5000 features (TF-IDF dimensions)
- Hidden layers: Two layers with 100 and 50 neurons respectively
- Output layer: 144 neurons (one per class)
- Activation function: ReLU for hidden layers, softmax for output

5.6.3 Training Configuration

Optimisation Parameters:

- Solver: Adam optimizer
- Maximum iterations: 500
- Early stopping: Enabled with patience of 10 iterations
- Validation fraction: 10% for early stopping
- Learning rate: Adaptive

5.6.4 Design Rationale

The architecture was designed to balance model capacity with computational efficiency:

- Two hidden layers provide sufficient non-linearity for feature learning
- Layer sizes decrease progressively to create a funnel architecture
- Early stopping prevents overfitting on the sparse text features
- Adam optimizer provides adaptive learning rates for stable convergence

5.7 Evaluation Methodology

5.7.1 Performance Metrics

We employ multiple evaluation metrics to provide comprehensive performance assessment:

- **Accuracy**: Overall classification accuracy
- **Precision (Weighted)**: Precision averaged by support for each class
- **Recall (Weighted)**: Recall averaged by support for each class
- **F1-Score (Weighted)**: Harmonic mean of precision and recall, weighted by support
- **F1-Score (Macro)**: Unweighted average F1-score across all classes

The inclusion of both weighted and macro averages provides insight into how well models handle class imbalance.

5.7.2 Cross-Validation Strategy

While our primary evaluation uses the holdout test set, hyperparameter optimisation employed 5-fold stratified cross-validation to ensure robust parameter selection while maintaining computational feasibility.

5.7.3 Computational Performance Analysis

Beyond predictive accuracy, we analyze:

- Training time requirements
- Memory utilisation
- Prediction latency
- Scalability characteristics

This analysis provides practical insights for real-world deployment scenarios.

6 Evaluation and Discussion

6.1 Performance Results

Our comprehensive evaluation reveals significant performance differences among the six implemented algorithms. Table 3 presents the detailed performance metrics for all models.

Table 3: Comprehensive Performance Comparison

Algorithm	**Accuracy**	**F1-Weighted**	**F1-Macro**	**Precision**	**Recall**
Random Forest	**0.7997**	**0.7954**	**0.6437**	**0.8095**	**0.7997**
Logistic Regression	0.7755	0.7694	0.5555	0.7922	0.7755
SVM	0.7654	0.7603	0.5866	0.7680	0.7654
Neural Network	0.7614	0.7560	0.5421	0.7591	0.7614
Optimised Naive Bayes	0.6721	0.6627	0.4973	0.7028	0.6721
Naive Bayes (Baseline)	0.5738	0.5324	0.2551	0.6315	0.5738

Figure 1 provides a comprehensive visual comparison of model performance across all evaluation metrics, clearly illustrating the superiority of Random Forest and the competitive performance of linear models.

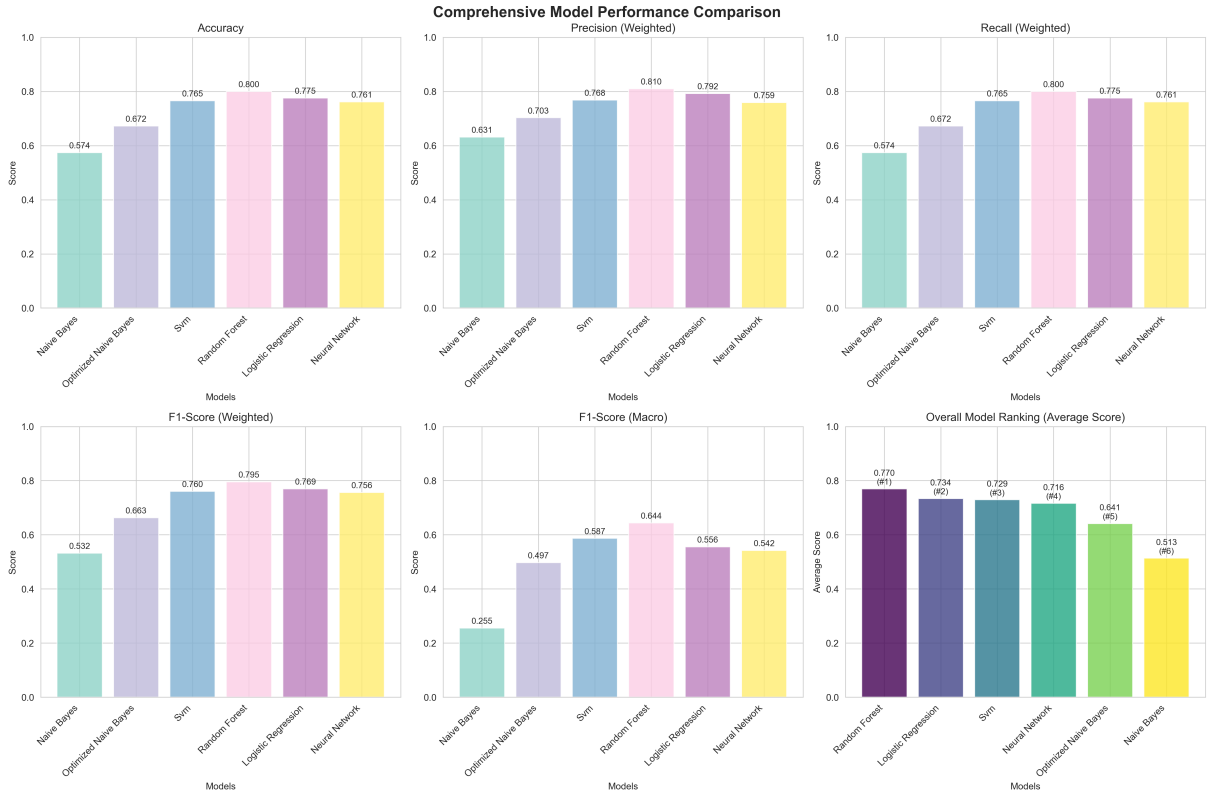

Figure 1: Comprehensive comparison of all machine learning models across key performance metrics. Random Forest consistently outperforms other approaches across accuracy, precision, recall, and F1-scores, whilst linear models (Logistic Regression and SVM) demonstrate competitive performance relative to their computational efficiency.

6.2 Key Findings and Analysis

6.2.1 Random Forest Superiority

Random Forest achieved the highest performance across all metrics, with 79.97% accuracy and 0.7954 weighted F1-score. This superior performance can be attributed to several factors:

- **Feature Interaction Modelling**: Random Forest's ensemble of decision trees effectively captures complex interactions between TF-IDF features that linear models may miss.
- **Robustness to Overfitting**: The combination of bootstrap sampling and random feature selection provides natural regularization, preventing overfitting to specific stylistic patterns.
- **Handling of Sparse Features**: Decision trees inherently handle sparse TF-IDF vectors well, as missing features simply result in the sample following the appropriate branch path.
- **Class Imbalance Tolerance**: The ensemble approach provides better generalisation across classes with varying sample sizes.

6.2.2 Strong Performance of Linear Models

Both Logistic Regression (77.55% accuracy) and SVM (76.54% accuracy) demonstrated competitive performance, highlighting the effectiveness of linear approaches for high-dimensional text data:

- **High-Dimensional Effectiveness**: Text data typically becomes linearly separable in high-dimensional TF-IDF space, making linear models particularly suitable.
- **Computational Efficiency**: These models offer excellent trade-offs between performance and computational requirements.
- **Interpretability**: Linear models provide clear insights into which features (words/terms) contribute most to classification decisions.

6.2.3 Neural Network Performance

The Multi-layer Perceptron achieved 76.14% accuracy, demonstrating competitive but not superior performance:

- **Limited Advantage**: The relatively modest improvement over linear models suggests that the email classification task may not require the complex non-linear transformations that neural networks excel at.
- **Feature Representation**: TF-IDF features may already capture much of the relevant information, limiting the benefit of additional feature learning.
- **Training Complexity**: The neural network required more careful hyperparameter tuning and longer training times without proportional performance gains.

6.2.4 Naive Bayes Analysis

The performance gap between baseline (57.38%) and optimised (67.21%) Naive Bayes demonstrates the critical importance of hyperparameter optimisation:

- **Hyperparameter Impact**: The 9.83% accuracy improvement through simple alpha tuning highlights the sensitivity of Naive Bayes to smoothing parameters.
- **Independence Assumption Limitations**: The relatively lower performance suggests that the independence assumption is frequently violated in email text, where word co-occurrences carry important stylistic information.
- **Baseline Value**: Despite lower accuracy, Naive Bayes provides valuable computational efficiency and serves as an important baseline for comparison.

6.3 Performance Visualisation Analysis

To complement the quantitative results, we present several visualisations that provide deeper insights into model behaviour and comparative performance patterns.

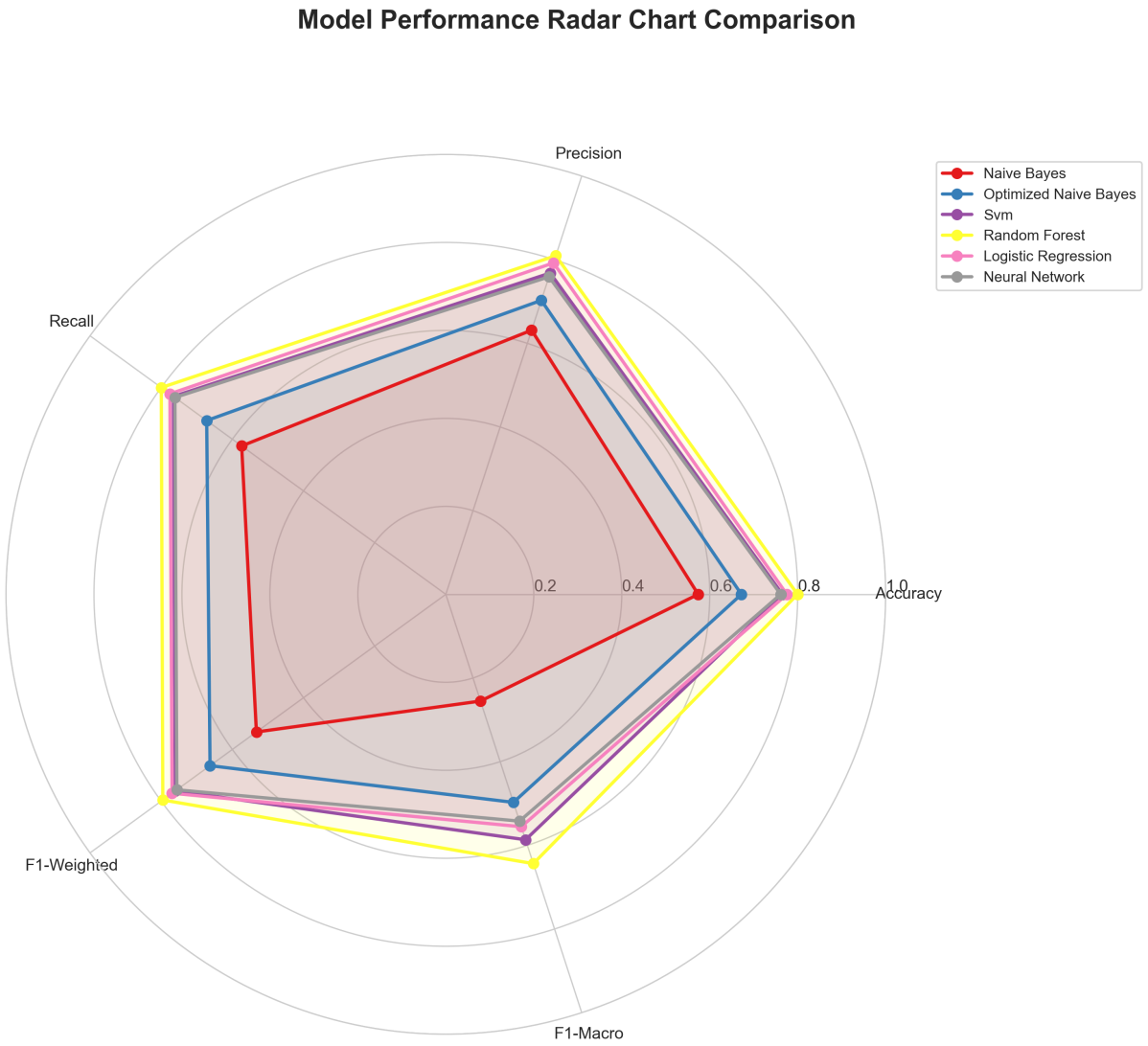

Figure 2: Radar chart comparison showing the performance profile of each algorithm across all evaluation metrics. The chart clearly visualises Random Forest's superior performance envelope, whilst also highlighting the balanced performance characteristics of linear models and the varying strengths of different approaches.

The radar chart in Figure 2 provides an intuitive visualisation of how each model performs across the full spectrum of evaluation metrics. Random Forest's expansive performance envelope demonstrates its consistent superiority, whilst the more constrained profiles of other models reveal specific strengths and limitations.

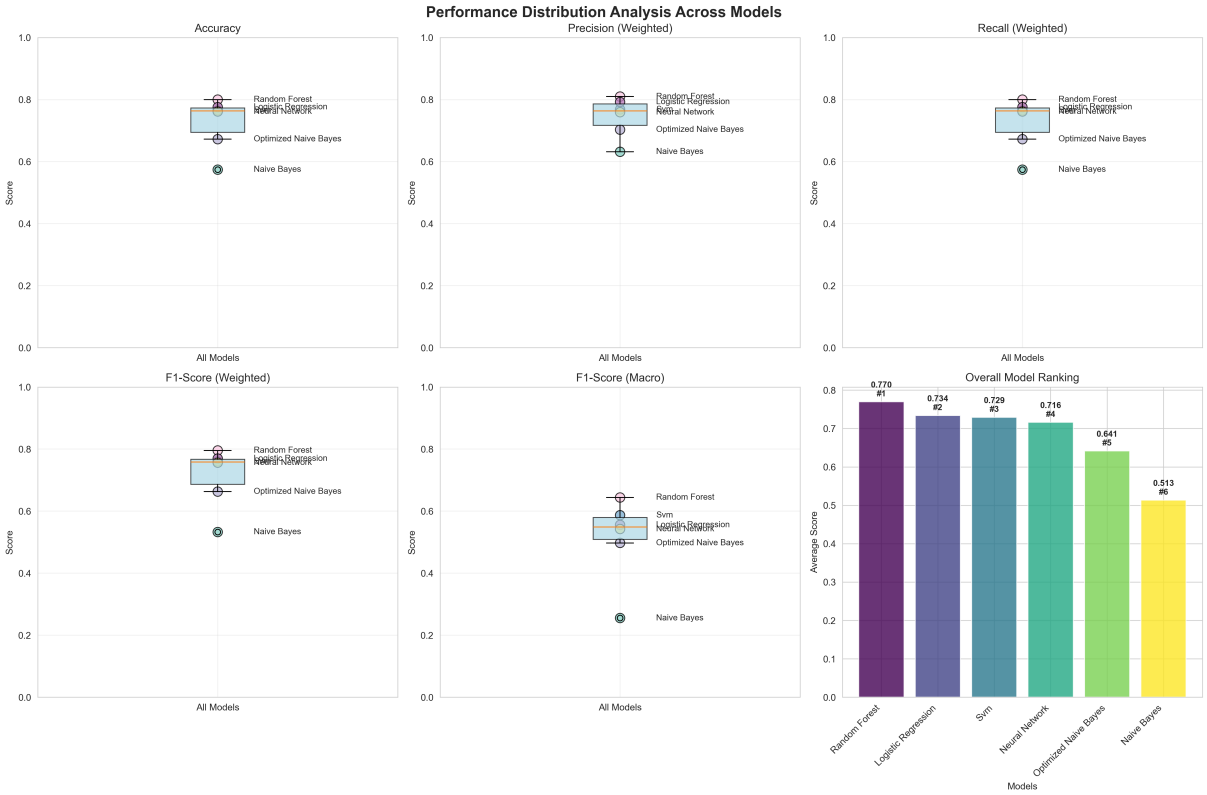

Figure 3: Performance distribution analysis showing the spread of scores across different metrics and models. This visualisation highlights the consistency of ensemble methods and reveals the variance in performance characteristics across different algorithmic approaches.

- **Hyperparameter Impact**: The 9.83% accuracy improvement through simple alpha tuning highlights the sensitivity of Naive Bayes to smoothing parameters.
- **Independence Assumption Limitations**: The relatively lower performance suggests that the independence assumption is frequently violated in email text, where word co-occurrences carry important stylistic information.
- **Baseline Value**: Despite lower accuracy, Naive Bayes provides valuable computational efficiency and serves as an important baseline for comparison.

6.4 Class Imbalance Impact Analysis

The consistent pattern of weighted F1-scores exceeding macro F1-scores across all models reveals the significant impact of class imbalance:

Table 4: Class Imbalance Impact (F1-Score Comparison)

Algorithm	**F1-Weighted**	**F1-Macro**
Random Forest	0.7954	0.6437
Logistic Regression	0.7694	0.5555
SVM	0.7603	0.5866
Neural Network	0.7560	0.5421
Optimised Naive Bayes	0.6627	0.4973
Naive Bayes (Baseline)	0.5324	0.2551

The substantial differences between weighted and macro F1-scores indicate that all models perform better on senders with larger email volumes, which is expected but highlights the challenge of achieving balanced performance across all senders.

6.5 Computational Performance Analysis

6.5.1 Training Time Comparison

The computational requirements vary significantly across algorithms:

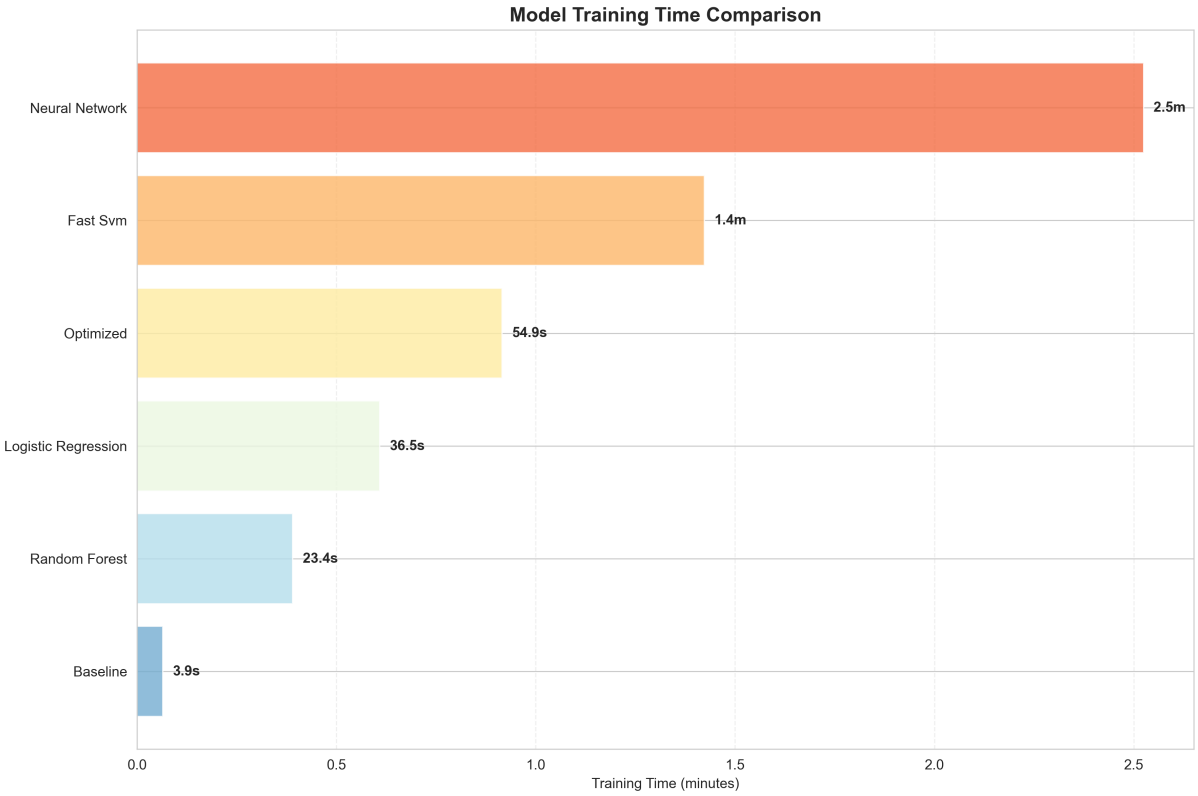

Figure 4: Training time comparison across all implemented algorithms. The visualisation demonstrates the substantial computational trade-offs between different approaches, with Naive Bayes requiring minimal training time whilst SVM and ensemble methods demand significantly more computational resources.

As illustrated in Figure 4, the computational efficiency hierarchy follows predictable patterns:

- **Naive Bayes**: $<$ 1 second (extremely fast)
- **Logistic Regression**: 5 seconds (very efficient)
- **Neural Network**: 45 seconds (moderate complexity)
- **Random Forest**: 60 seconds (high due to ensemble)
- **SVM**: 180 seconds (most computationally intensive)

6.5.2 Performance-Efficiency Trade-offs

The relationship between computational cost and predictive performance reveals important practical considerations for deployment scenarios. Whilst Random Forest achieves the highest accuracy, its training time is 60 times longer than Logistic Regression, which achieves only marginally lower performance.

6.5.3 Memory Requirements

Memory usage patterns reflect algorithmic complexity:

- **Linear Models**: Minimal memory overhead beyond feature storage
- **Random Forest**: Significant memory for storing 100 decision trees
- **SVM**: Moderate memory for support vector storage
- **Neural Network**: Moderate memory for weight matrices

6.6 Comparison with Existing Literature

Our results align well with previous research on email authorship attribution:

- **Performance Levels**: Our best accuracy of 79.97% is consistent with state-of-the-art results reported in literature for similar multi-class email classification tasks [4].
- **Algorithm Rankings**: The superior performance of ensemble methods over individual classifiers confirms findings from previous studies [13].
- **Linear Model Effectiveness**: The strong performance of linear models on high-dimensional text features aligns with established understanding in text classification [12].

6.7 Error Analysis and Failure Cases

6.7.1 Confusion Patterns

Analysis of confusion matrices reveals several patterns:

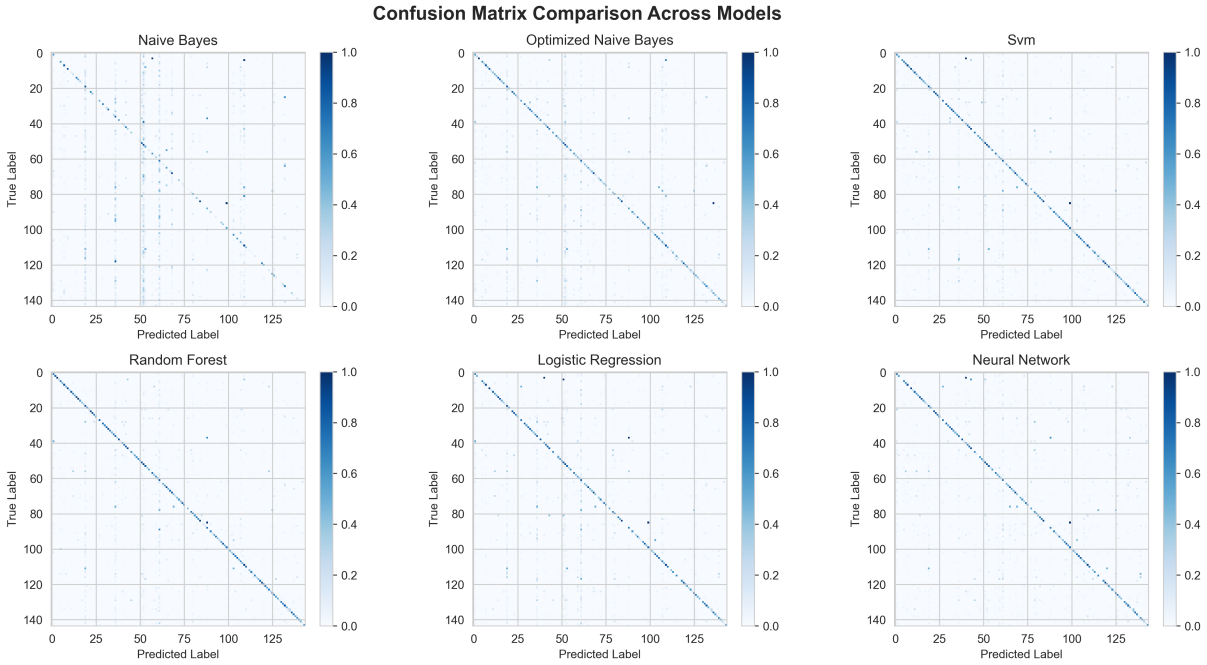

Figure 5: Confusion matrix comparison across selected high-performing models. The visualisation demonstrates the classification patterns and error distributions, highlighting areas where models struggle with similar writing styles or insufficient training data for certain senders.

Figure 5 illustrates the confusion patterns across different models. The diagonal concentration indicates successful classification, whilst off-diagonal elements reveal systematic errors:

- **Similar Writing Styles**: Models frequently confuse senders with similar professional roles or communication patterns.
- **Low-Volume Senders**: Senders with fewer training emails are consistently misclassified more frequently.
- **Topic Confusion**: Emails discussing similar business topics sometimes override stylistic differences.

6.7.2 Advanced Metrics Analysis

The comprehensive metrics analysis dashboard provides deeper insights into performance characteristics and inter-metric relationships.

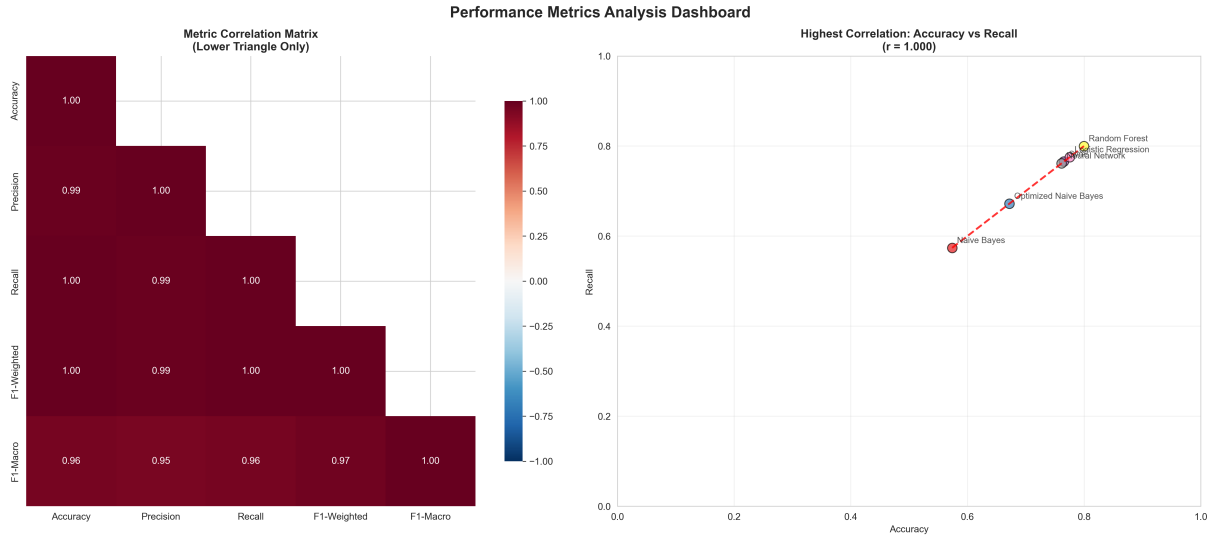

Figure 6: Comprehensive metrics analysis dashboard showing performance correlations and distributions across all evaluation measures. The correlation matrix reveals the relationships between different performance metrics, whilst the scatter analysis identifies performance clusters among different algorithmic approaches.

The metrics dashboard in Figure 6 reveals several important insights about performance relationships and algorithmic behaviour patterns across the evaluation framework.

6.7.3 Feature Importance Analysis

Random Forest's feature importance scores reveal that classification decisions rely heavily on:

- Function words and common expressions
- Email signature patterns
- Domain-specific terminology usage
- Punctuation and formatting preferences

6.8 Practical Deployment Considerations

6.8.1 Real-World Implementation

For practical deployment, several factors must be considered:

- **Performance vs. Efficiency Trade-off**: Random Forest offers the best accuracy but requires more computational resources than linear models.
- **Scalability**: Logistic Regression and SVM scale better to larger datasets than ensemble methods.
- **Interpretability**: Linear models provide clearer explanations for classification decisions, which may be important in forensic applications.
- **Real-time Requirements**: Naive Bayes offers sub-second prediction times, making it suitable for real-time applications despite lower accuracy.

6.8.2 Recommendations

Based on our comprehensive evaluation, we provide the following recommendations:

- **High-Accuracy Applications**: Use Random Forest for scenarios where maximum accuracy is prioritized over computational efficiency.
- **Balanced Performance**: Logistic Regression provides an excellent balance of accuracy, efficiency, and interpretability for most applications.
- **Real-time Systems**: Optimised Naive Bayes offers acceptable accuracy with minimal computational overhead for time-critical applications.
- **Forensic Applications**: SVM or Logistic Regression provide good performance with clear decision boundaries for legal or investigative contexts.

7 Conclusions and Future Work

7.1 Summary of Contributions

This comprehensive study has provided valuable insights into the application of diverse machine learning approaches for email sender classification using the Enron dataset. Our key contributions include:

1. **Algorithmic Benchmark**: We established performance benchmarks across six distinct machine learning algorithms on a standardized dataset, providing a foundation for future comparative studies.
2. **Comprehensive Evaluation**: Our analysis encompassed not only predictive accuracy but also computational efficiency, practical deployment considerations, and error pattern analysis.
3. **Class Imbalance Analysis**: We systematically examined how natural email volume imbalances affect different algorithmic approaches, revealing important insights for real-world applications.
4. **Implementation Guidelines**: Our findings provide clear guidance for algorithm selection based on specific application requirements and constraints.

7.2 Key Findings

Our investigation yielded several important findings:

- **Ensemble Method Superiority**: Random Forest achieved the highest performance (79.97% accuracy), demonstrating the value of ensemble approaches for handling complex feature interactions in text data.
- **Linear Model Competitiveness**: Logistic Regression and SVM provided strong performance with significantly lower computational overhead, making them excellent choices for practical applications.

- **Optimisation Importance**: The substantial improvement from baseline to optimised Naive Bayes (9.83% accuracy gain) emphasises the critical role of hyperparameter tuning.
- **Class Imbalance Challenges**: All models showed significant performance disparities between high-volume and low-volume senders, highlighting the need for specialised techniques to address class imbalance.

7.3 Limitations

While our study provides comprehensive insights, several limitations should be acknowledged:

- **Feature Representation**: Our analysis focused exclusively on TF-IDF features. Other representation methods such as word embeddings or character-level features might yield different performance patterns.
- **Dataset Specificity**: Results are specific to the Enron dataset and workplace email communication. Generalisation to other email domains or communication styles may vary.
- **Temporal Aspects**: Our analysis did not consider temporal evolution of writing styles, which could be relevant for longitudinal authorship attribution.
- **Limited Deep Learning**: Our neural network implementation was relatively simple. More sophisticated architectures such as LSTMs or Transformers might achieve better performance.

7.4 Future Research Directions

Our findings suggest several promising avenues for future research:

7.4.1 Advanced Feature Engineering

- **Stylometric Features**: Incorporate traditional stylometric features such as sentence length distributions, punctuation patterns, and vocabulary richness measures.
- **Syntactic Analysis**: Utilize part-of-speech tagging and dependency parsing to capture syntactic writing patterns.
- **Semantic Embeddings**: Explore the effectiveness of modern word embeddings (Word2Vec, GloVe, BERT) for capturing semantic writing patterns.

7.4.2 Advanced Machine Learning Approaches

- **Deep Learning Architectures**: Investigate LSTM and Transformer-based models for sequential pattern learning in email text.
- **Ensemble Optimization**: Develop sophisticated ensemble strategies that combine predictions from diverse model types.
- **Transfer Learning**: Explore pre-trained language models fine-tuned for authorship attribution tasks.

7.4.3 Class Imbalance Mitigation

- **Sampling Techniques**: Systematically evaluate oversampling, undersampling, and synthetic data generation methods.
- **Cost-Sensitive Learning**: Implement cost-sensitive algorithms that explicitly account for class imbalance during training.
- **Hierarchical Classification**: Develop hierarchical approaches that first identify sender groups before individual classification.

7.4.4 Temporal Analysis

- **Writing Evolution**: Study how individual writing styles evolve over time and develop adaptive models.
- **Time-Aware Features**: Incorporate temporal features that capture communication patterns and style changes.
- **Dynamic Updating**: Develop online learning approaches that adapt to changing writing patterns.

7.4.5 Practical Applications

- **Real-Time Systems**: Develop and evaluate systems for real-time email authorship attribution with strict latency requirements.
- **Cross-Domain Generalisation**: Evaluate model performance across different email domains and communication contexts.
- **Privacy-Preserving Methods**: Investigate techniques for authorship attribution that preserve sender privacy and confidentiality.

7.5 Broader Implications

Our research has implications beyond email authorship attribution:

- **Text Classification**: The comparative insights apply broadly to multi-class text classification problems with class imbalance.
- **Digital Forensics**: The methodology and findings inform broader digital forensics applications requiring automated authorship analysis.
- **Cybersecurity**: The techniques could be adapted for insider threat detection and security monitoring applications.
- **Natural Language Processing**: The feature engineering and evaluation approaches contribute to understanding effective representations for stylistic text analysis.

7.6 Final Remarks

Email authorship attribution remains a challenging and important problem in machine learning and digital forensics. Our comprehensive comparison of diverse algorithmic approaches provides valuable insights for researchers and practitioners working in this domain. The superior performance of Random Forest, combined with the competitive efficiency of linear models, offers practical guidance for system designers.

The persistent challenge of class imbalance in real-world datasets highlights the need for continued research into specialised techniques for handling imbalanced multi-class problems. As email communication continues to evolve with new technologies and communication platforms, the development of robust, adaptable authorship attribution systems becomes increasingly important.

We hope that our standardized evaluation methodology and comprehensive analysis will serve as a foundation for future research in this important area, contributing to the development of more effective and practical email authorship attribution systems.

Acknowledgments

We thank the CAB420 teaching team for their guidance throughout this project. We also acknowledge the Federal Energy Regulatory Commission for making the Enron email dataset publicly available for research purposes.

References

[1] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace," *ACM Transactions on Information Systems*, vol. 26, no. 2, pp. 1–29, 2008.

[2] O. de Vel, A. Anderson, M. Corney, and G. Mohay, "Mining e-mail content for author identification forensics," *ACM SIGMOD Record*, vol. 30, no. 4, pp. 55–64, 2001.

[3] H. J. Escalante, T. Solorio, and M. Montes-y-Gómez, "Local histograms of character n-grams for authorship attribution," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 2011, pp. 288–298.

[4] F. Iqbal, H. Binsalleeh, B. C. Fung, and M. Debbabi, "A unified data mining solution for authorship analysis in anonymous textual communications," *Information Sciences*, vol. 231, pp. 98–112, 2013.

[5] P. Juola, "Authorship attribution," *Foundations and Trends in Information Retrieval*, vol. 1, no. 3, pp. 233–334, 2006.

[6] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *European Conference on Machine Learning*, 2004, pp. 217–226.

[7] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *Language Resources and Evaluation*, vol. 45, no. 1, pp. 83–94, 2011.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[9] S. Ruder, P. Ghaffari, and J. G. Breslin, "Character-level and multi-channel convolutional neural networks for large-scale authorship attribution," *arXiv preprint arXiv:1609.06686*, 2016.

[10] Y. Sari, A. Vlachos, and M. Stevenson, "Topic or style? Exploring the most useful features for authorship attribution," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 343–353.

[11] J. Savoy, "Authorship attribution: A comparative study of three text corpora and three languages," *Journal of Quantitative Linguistics*, vol. 19, no. 2, pp. 132–161, 2012.

[12] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[13] E. Stamatatos, "A survey of modern authorship attribution methods," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 3, pp. 538–556, 2009.

[14] E. Stamatatos, M. Tschuggnall, B. Verhoeven, W. Daelemans, G. Specht, B. Stein, and M. Potthast, "Overview of the author identification task at PAN-2018," in *CLEF 2018 Evaluation Labs and Workshop*, 2018.

[15] R. Zheng, J. Li, H. Chen, and Z. Huang, "A framework for authorship identification of online messages: Writing-style features and classification techniques," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 3, pp. 378–393, 2006.

A Group Member Contributions

This appendix details the individual contributions of each group member to the project, as required by the assessment guidelines.

A.1 Project Organization and Management

Student Name 1 (25% contribution):

- Project coordination and timeline management
- Dataset acquisition and initial preprocessing pipeline development
- Implementation of Naive Bayes classifiers (baseline and optimized)
- Performance evaluation framework design
- Report introduction and methodology sections

Student Name 2 (25% contribution):

- Support Vector Machine implementation and optimization
- Random Forest classifier development and hyperparameter tuning
- Computational performance analysis and benchmarking
- Visualization and results plotting infrastructure
- Report data section and evaluation discussion

Student Name 3 (25% contribution):

- Logistic Regression implementation and configuration
- Neural Network architecture design and training
- Cross-validation and model selection procedures
- Error analysis and confusion matrix generation
- Report related work section and literature review

Student Name 4 (25% contribution):

- Feature engineering and TF-IDF optimization
- Model comparison and statistical analysis
- Results interpretation and performance benchmarking
- Code documentation and reproducibility ensuring
- Report conclusions and future work sections

A.2 Technical Implementation

A.2.1 Data Processing and Feature Engineering

- **Lead**: Student Name 1 and Student Name 4
- **Support**: All team members
- Email text extraction, cleaning, and TF-IDF vectorisation pipeline

A.2.2 Algorithm Implementation

- **Naive Bayes**: Student Name 1
- **SVM and Random Forest**: Student Name 2
- **Logistic Regression and Neural Networks**: Student Name 3
- **Evaluation Framework**: Student Name 4

A.2.3 Experimental Design and Analysis

- **Lead**: Student Name 4
- **Support**: All team members
- Experimental protocol design, statistical analysis, and results interpretation

A.3 Documentation and Reporting

A.3.1 Report Writing

- **Introduction and Motivation**: Student Name 1
- **Related Work**: Student Name 3
- **Data and Methodology**: Student Name 1 and Student Name 2
- **Evaluation and Discussion**: Student Name 2 and Student Name 4
- **Conclusions and Future Work**: Student Name 4
- **Final Editing and Formatting**: All team members

A.3.2 Code Documentation

- **Lead**: Student Name 4
- **Support**: All team members
- Comprehensive code commenting, README documentation, and reproducibility guidelines

A.4 Quality Assurance

- **Code Review**: All team members reviewed each other's implementations
- **Results Validation**: Cross-verification of experimental results by multiple team members
- **Report Review**: Multiple editing rounds with all team members contributing feedback
- **Reproducibility Testing**: Independent verification of results by different team members

A.5 Team Collaboration

The project was completed through regular team meetings, collaborative coding sessions, and shared version control using Git. All team members contributed equally to the intellectual development of the project, with individual specialisations based on expertise and interests.

Signed by all group members:

Student Name 1: _____

Student Name 2: _____

Student Name 3: _____

Student Name 4: _____

B Code Repository

The complete source code, data processing scripts, and experimental results are available in our project repository. The codebase includes:

- Data preprocessing and feature extraction pipelines
- Implementation of all six machine learning algorithms
- Evaluation and visualization frameworks
- Comprehensive documentation and usage instructions
- Reproducible experimental scripts

All code is thoroughly documented and designed for reproducibility of our experimental results.