LUND UNIVERSITY

SCHOOL OF ECONOMICS AND MANAGEMENT

A Comparative Analysis of Classification Methods Across Document Granularities and Unseen Large Language Models

Declan George O'Leary

Department of Statistics
Lund University School of Economics and Management

Supervisor:	Hamed Sabahno
Course:	STAN40 – Statistics: Master's Thesis, 15 credits
Seminar Date:	June 3, 2026

Abstract

The rapid public adoption of large language models (LLMs) has transformed written communication and creative production, while simultaneously creating growing concerns regarding authenticity, misinformation, and the preservation of human authorship. As AI-generated literary content becomes increasingly sophisticated, the ability to reliably distinguish machine-generated prose from authentic human writing has become a significant challenge. This thesis evaluates the effectiveness of multiple machine learning and deep learning classification methods in detecting AI-generated literary imitations and examines how document granularity and cross-model generalization impact classification performance. A curated dataset was constructed using eight English-language novels obtained from Project Gutenberg alongside stylistically constrained imitation chapters generated by three modern LLMs: OpenAI's ChatGPT, Anthropic's Claude, and Google's Gemini. To minimize contextual leakage, prompts enforced consistency in setting, characters, and authorial style, requiring classifiers to rely primarily on lexical and structural characteristics rather than obvious narrative cues. Textual data were transformed using TF-IDF vectorization and evaluated using Logistic Regression, Random Forest, Support Vector Machine (SVM), Naive Bayesian Classifier (NBC), a custom Neural Network, and the transformer-based RoBERTa model.

Experiments were conducted across three document granularities: chapters, pages, and paragraphs. Results demonstrate that document segmentation substantially influences classifier behavior. RoBERTa consistently achieved the strongest performance, maintaining near-perfect accuracy across all document sizes, while Random Forest and NBC achieved perfect chapter-level accuracy but degraded significantly as document size decreased. Logistic Regression and SVM exhibited improved performance at intermediate page-level segmentation, suggesting that some linear models benefit more from increased corpus size than from larger contextual windows. Across multiple models, a persistent false negative bias emerged, with AI-generated text frequently misclassified as authentic human writing. Further experiments explored model robustness through repeated random-seed evaluation, attribution of AI-generated text to its originating LLM, and Leave-One-Out (LOO) testing against unseen language models. While several classifiers successfully identified the originating LLM with near-perfect accuracy, most models demonstrated substantial degradation when evaluated against previously unseen AI architectures, revealing limited cross-model generalization. Among the evaluated methods, NBC demonstrated the strongest robustness under zero-shot conditions.

Overall, this thesis established that AI-generated literary data contains detectable statistical signatures, but that classifier effectiveness is highly dependent on document structure and exposure to diverse generator models. These findings contribute a scalable benchmark framework for AI text detection and highlight the importance of document granularity and out-of-distribution evaluation in the development of future detection systems.

Contents

1 Introduction

The release of ChatGPT in 2022 has brought the technology of artificial intelligence (AI) from the pages of science fiction irrevocably into all our lives. The most famous and widespread applications of this technology focus on text, and function as large language models (LLM). These models largely operate using an expanded form of the decoder section of transformer architecture and function by receiving a text prompt and returning a text output. There are a myriad of uses for this technology applied every day in both personal and professional settings. People are using these innovative technologies to quickly draft memos, summarizing articles, constructing marketing material, social media posts, and even writing entire novels. Despite the opportunity for increased productivity, there are many negative aspects of the implementation of these AI systems. Their use on social media platforms has at best flooded a public forum with posts devoid of real world backing and at worst a method to spread propaganda and misinformation at unprecedented frequency and credibility. These technologies are also being leveraged in the arts, and specifically for this project novels. Online marketplaces like Amazon are littered with AI generated novels being produced with minimal effort to the detriment of the marketplace through harming their reputation, impacting true novelists by changing the perception of what a novel is meant to be, and obfuscating new novels, thus making it more difficult for new writers to pursue the craft. In Baidou-Anu and Ansah's 2023 paper on the proliferation of AI in education they write that AI "is a nearly undeniable fact that ChatGPT and other generative AI have come to stay and will continue revolutionizing the current educational system."[3]

As the use of LLM have spread, the ability to differentiate between writing generated by these models and created by humans has become increasingly important. The process of identifying AI generated text is still developing with the method utilized in this study being the use of TF-IDF vectorization on processed textual data to reveal the underlying lexical fingerprint of the AI text. In their 2026 paper, Makhmutova et. al found a wide disparity in the effectiveness of different commercially available AI detection and identification systems. This led them to express the risks associated with reliance on these models with "the variability observed across detectors underscores the lack of standardized evaluation metrics and raises concerns about equity and fairness when institutions rely on a single system for high-stakes decisions."[15] Because modern LLMs are trained to mimic human syntax with extreme fidelity, standard algorithmic evaluations often fail to grasp the structural or distributional anomalies that separate machine outputs from human creativity. Thus, developing robust, transparent, and computationally accessible classification pipelines is essential to preserving intellectual property and defending the economic viability of creative industries.

This project addresses these challenges by constructing a rigorous benchmarking framework designed to evaluate how effectively different statistical and deep learning models can distinguish between authentic, human literature and sophisticated AI-generated imitations. Utilizing a curated dataset of eight significant novels from Project Gutenberg, this study pairs authentic text chapters with high-fidelity imitations generated by three state-of-the-art LLMs: OpenAI's ChatGPT[18], Anthropic's Claude[1], and Google's Gemini[12]. By enforcing strict constraints on setting, character pools, and stylistic prompts the data generation process deliberately isolates the structural and vocabulary nuances of the prose itself, forcing the downstream classifier to rely entirely on stylistic characteristics rather than obvious contextual clues.

To evaluate these differences, this thesis deploys and compares a diverse spectrum of machine learning architectures. These range from computationally lightweight baseline algorithms to advanced, deep contextual networks. On the classical statistical front, Logistic Regression is implemented to establish a transparent, linear baseline, alongside Naive Bayesian Classifier to exploit word frequency

distributions. To capture non-linear decision boundaries and complex feature interactions within the text matrices, Support Vector Machines (SVM) with various kernel functions and Random Forest ensembles are introduced. Furthermore, a custom designed Multi-Layer Perceptron (Neural Network) is deployed to automatically extract high dimensional hierarchical feature representations from sparse vector spaces. Finally, as a state-of-the-art comparison, RoBERTa. RoBERTa is a deep learning, bidirectional transformer-based model used to shift the analysis from static word frequencies to deep contextual semantic awareness.

The ultimate significance of this thesis lies in the systemic exploration of document granularity and cross-model generalizability. Rather than evaluating text detection at a fixed scale, this research breaks the data corpus down across three distinct structural tiers: chapters, pages, and paragraphs. In doing so, this study maps the exact inverse relationship between document size and corpus volume, revealing how this relationship impacts classifier accuracy. Furthermore, by training models on specific LLMs and testing them against others, this work uncovers whether these text classifiers are genuinely identifying universal traits of AI produced text, or simply overfitting to the proprietary idiosyncrasies of individual models. Ultimately, these insights provide a vital road map for building scalable, future-proof frameworks capable of safeguarding human artistry in an era riddled with computer generated content.

2 Data

The purpose of this project is to assess the quality of different classification methods capabilities at distinguishing between true human written novels and AI generated imitations. The selection of novels written by humans was sourced from Project Gutenberg. Using their library, a series of eight famous novels were selected. These novels are: 1. A Study in Scarlet by Arthur Conan Doyle[10], 2. Frankenstein; or, the modern prometheus by Mary Wollstonecraft Shelley[21], 3. Alice's Adventures in Wonderland by Lewis Carroll[6], 4. The Strange Case of Dr. Jekyll and Mr. Hyde by Robert Louis Stevenson[22], 5. Pride and Prejudice by Jane Austen[2], 6. Treasure Island by Robert Louis Stevenson[23], 7. The Adventures of Tom Sawyer by Mark Twain[24], and 8. Hard Times by Charles Dickens[9]. All eight of these novels were originally written in English. The novels were downloaded as txt files, cleaned, and processed to be utilized for the described range of applications in Python[20].

The generated AI data was imitations of the selected novel, paired so there would be an equal number of chapters of AI generated content and true novels. The rationale behind doing imitation chapters of real novels is that through prompting the LLM to specifically create an imitation of the specified novel it would counteract the influence of factors like setting, character names, and key novel elements that could otherwise influence the classification models unfairly. The specific prompts are denoted in Appendix section 6.1, but a general description of their purpose is to prompt the LLM to return a chapter that is approximately the length of the specified models mean length chapter inside the setting of the novel and with the style of the author. This was applied to three of the most prominent publicly available LLMs, ChatGPT, Claude, and Gemini. The specific models utilized were OpenAI's GPT-5.3, Anthropic's Sonnet 4.6, and Google's Gemini 3 Flash.

The decision to use these three models is multifaceted. One reason is to address concerns about data generation times. Each of these models has a maximum usage limit that when reached either reduces the quality of the response or completely stop accepting queries until a time threshold. This capacity limit proved to be a persistent issue even with these measures followed particularly for Claude. Anthropic's Claude demonstrated a noticeably lower threshold for the number of chapters that could be generated

in a day, which proved to be a source of delay. Another reason behind the use of three different LLMs for data generation is this opens up the avenue of exploration to understand how distinct the writings of different LLMs are and how effectively different LLMs can be used as training data for one another. The distribution of LLM of origin for the AI imitation chapters can be found in the table below. A key factor in the creation of this data is the continued effort to have the same number of chapters originating from each LLM. The final dataset has two extra chapters in ChatGPT, but throughout the process this ended up being as close it could come without redoing a chapter. The leave-one-out analysis has the data split by LLM of origin. The approximately even partition seen below allows for the training dataset to always be approximately two-thirds of the entire dataset, ensuring that it has sufficient data.

Table 1: Identifying the Number of AI Imitation Chapters per LLM

Model	**Number of Chapters**
ChatGPT	77
Claude	75
Gemini	75
Total AI Chapters	227

There were several issues that required a greater amount of action than simply following the algorithmic process of entering the prompt, entering the chapter in the standardized format, and repeating. The first of these challenges appear when using ChatGPT. When creating imitations of novels with longer average chapter length (generally over 3,000 words) would be unable to be generated in a single prompt. This issue was mitigated by taking chapter length into consideration when selecting novels to add to the dataset as well as addressing the shorter than minimum chapter length and requesting for it to be extended if it is clear to the eye. Google Gemini encountered a similar challenge as ChatGPT, but instead of generating chapters that were too short it generated chapters that were too long, sometimes being thousands of words longer than intended. These two issues were both treated with a low level of severity as if these caused the classification methods to be more readily able to identify their text then that would be a flaw in the model's output and not necessarily a consideration for exclusion. This policy was only ignored in blatantly obvious instances of length errors. A more serious issue with Gemini is its pattern of repeating the same outputs. This produced a more serious concern due to the potential impact of incorrectly adding identical datapoints into the dataset could both incorrectly skew the results and reduce the amount of available data. A quirk with the chapters prompted from Gemini is that it identified them with a chapter title. This title was used to search for repeated chapters and when they were identified an augmented prompt found in Appendix section 6.1 was used to make the LLM produce new material.

3 Methodology

This section outlines the complete methodological framework utilized throughout this research to evaluate the efficacy of different machine learning and deep learning classification models in distinguishing between novels and AI imitations. The methodology was designed to ensure both consistency and scalability across multiple forms of analysis including binary human and AI classification, multi-class LLM labeling, and the leave-one-out generalization experiment.

The overall process consisted of several major stages. First, the textual dataset was collected, cleaned, standardized, and partitioned into multiple document sizes. Second, the resulting feature matrices were divided into training, validation, and testing datasets to facilitate both model evaluation and

hyperparameter tuning. Third, the processed text was transformed into numerical feature representation through TF-IDF vectorization. Finally, a diverse range of statistical, ensemble, neural, and transformer-based classification methods were utilized and compared across the selected experimental states. A major focus of this thesis is the exploration of the inverse relationship between document size and corpus size within a fixed textual dataset, the methodological pipeline was intentionally designed to remain structurally consistent across chapter, page, and paragraph document sizes. This consistency ensures that changes in performance can be attributed primarily to document granularity and classifier behavior rather than inconsistencies in preprocessing or data construction.

3.1 Data Cleaning

Throughout the data gathering phase, the AI imitation chapters were gathered in a consistent style and format regardless of novel, but the same is not true for the novels sourced from Project Gutenberg. Project Gutenberg includes prefaces and footers at the end of each novel, which needed to be unilaterally removed. Additionally, there are inconsistent formatting and stylistic characteristics of the different novels, which provide a unique challenge in cleaning and splitting the data. Due to compute constraints of the selected LLMs the AI imitation data was generated on a chapter-by-chapter basis, which leads to the true human made novels needing to be split on a chapter-by-chapter basis. This is addressed through using REGEX on the most common methods of data splitting with more precise action taken for specific cases that warranted it. This newly formed chapter data was stored in a pandas data frame along with columns that describe the novel and chapter number the text corresponds to along with a Boolean denoting if it was generated through AI. This Boolean column is included due to AI imitation chapters also being given chapter numbers and novel identifiers, so that it can be utilized to directly compare the identifiability and other characteristics of the LLMs through established pairings of data. In order to prepare the data to be used in the selected classification models, the text data is stored as the input variable X, while the binary variable describing if the paired text originates from AI is denoted as the dependent, y, variable.

To expand the scope of this research beyond simply comparing the capabilities of the selected statistical classification methods on the data of novel chapters, one avenue for expansion is to increase the number of documents in the data corpus. This project utilizes a fixed amount of data, so in order to increase the number of documents in the corpus, the size of the documents needs to be reduced. Specifically, the new partition utilized is one page. Due to the text data being stored in txt files and then through chapters the symbols that typically denote a page break are not present leading to the necessity of utilizing a word count to determine when a page ends. The process for separating a page from the chapter of text is to split the text into 275 token (after preprocessing) bins until the end of the chapter. Due to it being unlikely that the end of a chapter would have exactly 275 words and to avoid the possibility of pages that are simply too short being included and negatively impacting the data quality and consistency the remaining text was only included in the data as a page if it contained at least 200 tokens.

To continue the expansion, a document size of a paragraph was also applied. This facilitates the continued research into how the inverse relationship between document size and corpus size on a fixed text data set can affect different classification methods. The method of developing paragraph sized data is similar to the documents of page size, but with the distinction of the line break symbols still being largely included in the text. The paragraphs are selected on a chapter-by-chapter basis with the primary splitting method being at the line break symbol. Due to the risk of reading issues or unique formats a secondary splitting system was also utilized. This safety system drops any paragraphs with less than 40 tokens and forces a split at 200 tokens. The idea being that the target of paragraphs should exceed the

next potentially smaller document size of sentence, while continuing to be smaller than pages.

3.2 Data Partitions

The final aspect before the application of the models is the inclusion of a training, test, and validation split of the data. The purpose of splitting this data is to provide a distinction within the data between what the various classification models learn from and what they are assessed on. The validation data is utilized for the tuning of hyperparameters for the various model but is not included at all when exploring the relationships, generalizability, or other characteristics on an LLM basis.

The specific data split methodology deployed for chapter based analysis, including the page and paragraph document size analysis, has the same process for splitting the data. There are two major principles in this system, the first being that each AI imitation chapter is paired with an actual human written chapter with from the same novel and with the same chapter number. This enshrines the pair together and ensures that whatever the other criteria used for the split will be there will be a 50/50 true/false split in the data for chapter document size data. The second key characteristic is the decision to create the splits on a novel-by-novel basis. This means that each novel will be entirely partitioned into one of the three datasets. Due to there being 8 novels included in the data, five are partitioned to training data, one for validation data, and two for test data. These assignments are determined through a pseudo-random python function.

Table 2: Description of Partitioning of Data across Document Sizes

Category	**Split**	**Human**	**AI**	**Total**
Chapter	Train	181	181	362
	Validation	24	24	48
	Test	22	22	44
Page	Train	1353	1380	2733
	Validation	253	268	521
	Test	173	161	334
Paragraph	Train	3552	4090	7642
	Validation	643	745	1388
	Test	405	441	846

The above table demonstrates a potential risk with the data partitioning methodology utilized in this project. As stated earlier, the partitioning between the train, validation, and test set is done by assigning the chapters from the randomly selected novel to the given split. This results in the AI imitation and human chapters having the same number of documents due to the pairing of chapters. This true 1:1 ratio between does not persist for page and paragraph document sizes, as the pages and paragraphs are split from inside the chapters, thus the lengths of the chapters of the randomly selected novels impact the number of documents they produce at the smaller document sizes. This could potentially be a limitation due to a growing imbalance, which is seemingly caused by AI imitation chapters being on average longer, resulting in more documents being produced when document sizes are smaller. The imbalance likely does not have a significant effect in this analysis as the imbalance is still relatively small. The drastic difference in data between the train, validation, and test sets comes from the length of the novels that are randomly assigned to each split. The selected random seed resulted in two of the shortest novels being assigned as test data, and another shorter novel being assigned as validation data.

The largest difference between the data splitting between the general document size classification problems and the LLM leave-one-out is that in leave-one-out analysis, no validation set is used. The split is done along the line of LLM of origin for the associated AI chapter, so for each LLM that is being focused on there will be an even split of true and false values and it will contain approximately a third of the total data. This leaves the other two thirds to serve as the training data, allowing for research to be conducted comparing these models in regard to their inter-AI identifiability, but also assessing how well the different LLM data can be generalized across other LLMs. This is particularly interesting, because each novel has an approximately equal number of datapoints from each LLM which ensures that each LLM is in the training, validation, and test set regardless of the random seed.

3.3 Text Pre-processing

To handle the preprocessing[4] for this text data, a thorough but standard method was utilized in this vectorizer. The most basic facets of this tool are the include ensuring all characters in the text are lower cased and removing punctuation. An additional characteristic of this vectorizer is the removal of English stop words from the data. These are the most common words in the English language that are filtered out due to their lack of value provided. The final component of this preprocessing method is using a standard lemmatizer with part of speech tagging for context. A lemmatizer strips away the prefixes and suffixes of words and returns them to their most basic form without even modification for tense. This is supported by part of speech tagging which ensures that the lemmatizer can identify the correct root word in areas of confusion. The vectorizer together converts raw text data into a processed version ready to be vectorized.

$$\text{TF-IDF} = \frac{\text{Frequency of word, w, in document d}}{\text{Number of terms in document d}} \times \ln \frac{\text{total number of Documents}}{\text{Number of documents with word w}} \tag{1}$$

There are several major embedding techniques that can vectorize text, but the method utilized in this case is Term Frequency-Inverse Document Frequency (TF-IDF). This method measures a words importance to a document within the corpus, or as said in Zhang and Zhiyang (2021), "The larger the document frequency is, the more likely it is that a word is uninformative high frequency word."[26] This clearly highlights the TF-IDF weights in favor of rarer words and phrases. This produces a vector where each word has an associated value per document that highlights its rarity. This vector is then used as the independent, X, value in the following classification models, which results in the models striving to differentiate between human written novels and AI imitation novels through the frequency and rarity of words used.

3.4 Logistic Regression

Logistic Regression[19] is a supervised machine learning algorithm widely used for classification tasks and serves as one of the baseline models assessed in this study. The method estimates the probability that an observation belongs to a particular class by modeling the relationship between the input variables and the log-odds of the outcome. Logistic Regression operates on similar premise to linear regression as Lindholm notes in Machine Learning (2022), "Logistic regression can be viewed as a modification of the linear regression model so that it fits the classification (instead of regression) problem."[13] In other words, unlike linear regression Logistic Regression applies the logistic sigmoid function to constrain predicted outputs to values between 0 and 1, enabling probabilistic binary classification.

$$h(z) = \frac{e^z}{1+e^z} \tag{2}$$

Due to its relatively simple mathematical structure, Logistic Regression remains one of the most widely adopted and interpretable classification models in machine learning. The model provides computational efficiency, directly interpretable feature coefficients, and strong performance on linearly separable or moderately complex datasets. These characteristics make it particularly suitable as a baseline classifier against which more complex ensemble methods can be compared. In the context of this research, Logistic Regression establishes a reference point for evaluating the effectiveness of more sophisticated models in detecting AI-generated text and comparing the generalizability of large language models (LLMs).

Throughout this project, some instances of analysis need to be conducted in a multi-class environment. In those cases a baseline model that excels for the same reasons as Logistic Regression is Softmax Regression. Softmax Regression is a variant of Logistic Regression that allows for multi-class classification to be conducted. The output for this function is a vector containing the probabilities associated with each class, or the probability that the datapoint belongs to that specific class. Though this method is used as a standalone multi-class classification method it is most prominently utilized in neural networks as the output function that the results are fed through to produce the end results of a large classification problem.

$$g_m(x) = \frac{e^{\theta_m^T x}}{\sum_{j=1}^{M} e^{\theta_j^T x}} \tag{3}$$

3.5 Random Forest

Random Forest is an ensemble classifier assessed in this study that builds upon many decision trees through bootstrap aggregation (bagging) to improve classification. Decision trees are non-parametric supervised learning models that recursively partition the feature space into increasingly similar regions by selecting splits that maximize a purity criterion. The resulting hierarchical structure approximates complex nonlinear decision boundaries while remaining highly interpretable, as each internal node represents a decision rule and each terminal node corresponds to a prediction.

A weakness of individual decision trees is that each individual tree is prone to high variance and overfitting due to small disturbances in the training data. Random forests address this limitation through ensemble learning by constructing a large collection of decorrelated decision trees using bagging and random feature selection at each split. Breiman (2001) defines a Random Forest as “a classifier consisting of a collection of tree-structured classifiers... where each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest.”[5] Predictions from these trees are aggregated through majority voting, reducing variance while improving classification accuracy and generalization performance. Furthermore, Breiman (2001) argues that Random Forests are resistant to overfitting, noting that “use of the Strong Law of Large Numbers shows that they always converge so that overfitting is not a problem”[5] as the number of trees increases.

The application of Random Forest on these problems of identifying AI in a balanced dataset and comparing the generalizability of LLMs is heavily reliant on the power that having many iterations of decision trees determining the final classification provides. Random Forest, unlike Logistic Regression can readily handle multi-class classification problems, which facilitates this methods application across the full range of this research. The final major reason behind including random forest in this research is that it retains the flexibility of the decision trees, while having greater stability, accuracy, and robustness on challenging data.

3.6 Support Vector Machines

Support Vector Machines (SVM) are robust supervised learning models that classify data by constructing an optimal hyperplane in a high-dimensional feature space. The focus of SVM is denoted in Cortes and Vapnik (1995) where they write, "an optimal hyperplane is here defined as the linear decision function with maximal margin between the vectors of the two classes."[7] This quote signifies the priority of SVM to maximize the distance between the hyperplane and the datapoints to provide the cleanest division. This focus on the margin allows SVMs to achieve high generalization power even with relatively small datasets, as the model's complexity is defined by the number of support vectors rather than the total dimensionality of the input space. Relying entirely on a hyperplane would limit SVMs to only being capable of handling linear classification, but with kernels they are able to impact non-linear data types as well. Applying a kernel function maps the data into a higher dimensional space, which allows for a linear hyperplane to be applied. Four kernel functions are included as potential tuning options throughout this research: Linear Kernel, Polynomial Kernel, Radial Basis Function (RBF) Kernel, and Sigmoid Kernel. Linear kernels do not map to higher dimensions and as such are the default for SVM as described above. Polynomial kernels work best when instead of a linear hyperplane dividing the data a polynomial function can accomplish the task. RBF kernel is among the most frequently applied kernels as it moves the data into a space with a functionally infinite amount of dimensions and then measures the distance to establish the boundary from this transformation. Finally, Sigmoid kernels can allow SVM to serve as an alternative to neural networks with a Sigmoid activation function and can produce an output similar to logistic regression. Before tuning to determine the kernel for each scenario, the expected kernels are Sigmoid due to its intimate ties to binary classification and RBF due to its great capacity to adapt the dimensions to identify a linear partition.

The inclusion of SVMs in this research is driven by their capacity for high-dimensional classification through expanded dimensions and their built-in nature as a binary classifier, while still maintaining capabilities in multi-class problems. This adaptability ensures the model can be applied across the full scope of this research, comparing the distinct fingerprints of various LLMs. In scenarios with balanced datasets, which all of this projects research questions are, SVMs goal of maximizing the margin is often less susceptible to noise from generative patterns. Ultimately, SVMs offer a high-performance alternative to ensemble methods, providing a stable and mathematically grounded approach to identifying AI generated content across diverse datasets.

3.7 Naive Bayesian Classifier

The Naive Bayesian Classifier, also known as Naive Bayes, is a probabilistic supervised learning model grounded in Bayesian statistics and based on Bayes' theorem. The classifier operates under the naive assumption that features are conditionally independent given the class label. Although this assumption is rarely fully satisfied in real-world datasets, it significantly simplifies probability estimation by decomposing complex joint probability distributions into individual conditional probabilities. By estimating the posterior probability of each class from the prior probabilities and the likelihood of the observed features, the model assigns the class with the highest posterior probability to a given document. This study specifically utilizes the Multinomial Naive Bayes model, which is particularly well suited for textual classification tasks involving word frequencies and token counts. As described by McCallum and Nigam (2001), "the multinomial model captures word frequency information in documents,"[17] making it highly effective for sparse, high-dimensional textual datasets. Rather than modeling continuous feature distributions, Multinomial Naive Bayes represents documents through the frequency distribution

of terms within the vocabulary, allowing the model to efficiently identify statistical patterns associated with different classes.

An aspect of the Naive Bayesian Classifier that is particularly relevant to this research is its historically strong performance in natural language processing (NLP) and text classification tasks. Within textual analysis, document vocabularies naturally produce extremely high-dimensional feature spaces in which individual words function as features. Despite the simplifying independence assumption, word frequencies alone often provide sufficiently strong statistical signals for highly effective document classification. Consequently, Naive Bayes has become a traditional benchmark model in textual analysis due to its computational efficiency, scalability, and strong empirical performance without requiring the substantial computational resources associated with deep learning architectures.

The inclusion of Naive Bayes in this research is motivated by its effectiveness in high-dimensional textual classification and its ability to naturally support multiclass classification problems. These characteristics allow the model to be applied consistently across the full scope of this research, including comparisons between multiple large language models (LLMs) and human-authored literary texts. Furthermore, the probabilistic framework of Naive Bayes enables the identification of distinct vocabulary distributions and stylistic tendencies associated with AI-generated content. As a result, Naive Bayes provides a computationally efficient and mathematically grounded baseline for evaluating more complex machine learning approaches in detecting AI-generated writing.

3.8 Neural Network Based Approach

Neural Networks[16] are supervised learning models designed to identify complex nonlinear relationships within high-dimensional datasets through layers of interconnected computational nodes. This study specifically utilizes a feedforward neural network architecture. As described by Goodfellow et. al in the 2016 book Deep Learning, these models are referred to as feedforward because "information flows through the function being evaluated from x, through the intermediate computations used to define f, and finally to the output y."[11] Unlike linear classifiers, feedforward neural networks sequentially transform input features through hidden layers, enabling the model to learn nonlinear representations and complex decision boundaries that may not be linearly separable. The complexity and representational capacity of the network are determined by architectural choices including the number of hidden layers, the number of nodes per layer, and the activation functions used throughout the model. Nonlinearity is introduced through activation functions, allowing the network to model sophisticated classification boundaries within sparse, high-dimensional textual data. In this research, the neural network architecture consists of two hidden layers utilizing Rectified Linear Unit (ReLU) activation functions, followed by an output layer employing either a Sigmoid activation function for binary classification or a Softmax activation function for multiclass classification tasks.

The significant importance of neural networks to this specific study stems from the strong precedent of neural networks successful adaptation in natural language processing and textual analysis problems. When applied to text classification using TF-IDF matrices, the network is forced to operate in an expansive environment with a large amount of sparsity. Many traditional models treat the words based on their individual value, neural networks are able to understand the impact of complex combinations of word importance, which allows the neural networks to compress sparse TF-IDF vectors and matrices into complex representations that describe the semantic variations.

The inclusion of neural networks in this research is motivated by their effectiveness in high dimensional classification tasks and their flexibility as probabilistic learning models. Model optimization

is performed using the Adaptive Moment Estimation (Adam) optimizer in conjunction with binary cross-entropy or categorical cross-entropy loss functions depending on the classification objective. The adaptability of neural network architecture allows the model to be applied consistently across the full scope of this research, including both binary and multiclass classification problems involving large language models (LLMs). Consequently, neural networks provide a powerful and mathematically grounded approach for identifying complex linguistic patterns associated with AI-generated text while serving as a high-performance alternative to traditional machine learning classifiers.

3.9 RoBERTa - Transformer Architecture

RoBERTa (Robustly Optimized BERT Approach) is a transformer-based contextual language model derived from BERT and optimized for improved downstream sequence classification performance. Unlike traditional machine learning approaches that rely on sparse statistical representations such as TF-IDF, transformer architectures generate contextual embeddings through bidirectional self-attention mechanisms that evaluate tokens relative to surrounding linguistic context. The underlying BERT architecture utilizes masked language modeling, where Devlin et al. (2018) explain that the model "randomly masks some of the tokens from the input, and the objective is to predict the original vocabulary id of the masked word based only on its context."[8] This bidirectional training process enables the model to capture semantic and syntactic dependencies across entire sequences rather than treating words as isolated features. Building upon this architecture, Liu et al. (2019) "propose modifications to the BERT pretraining procedure that improve end-task performance,"[14] specifically using dynamic masking, training on full sentences without the next sentence prediction objective, larger mini-batches, and an expanded byte-level Byte Pair Encoding tokenizer. These optimizations allow RoBERTa to develop richer contextual representations and improved generalization performance when fine-tuned on downstream classification tasks. In contrast to the TF-IDF vectorization methods used by the other models within this study, RoBERTa utilizes its own pretrained contextual embedding pipeline through the Hugging Face Transformers package[25]. Raw text was tokenized using byte-level BPE encoding and truncated or padded to a maximum sequence length of 256 tokens prior to transformer processing. A binary classification head consisting of two output labels was then attached to the transformer sequence representation to facilitate AI-versus-human text classification.

The relevance of RoBERTa in this research is the position it holds as a state-of-the-art natural language processing and sequence classification tool. Within classification, document vocabularies inherently present dense syntactic interdependencies that shallow architectures can fail to extract. RoBERTa's use of bidirectional transformer architecture allows it to process the full context of a word based on its context, thus allowing for a greater amount of information than standard practices. This makes transformer models an exceptional benchmark for distinguishing generative text from human writing, as it moves beyond simple vocabulary mapping to evaluate the deeper stylistic cohesion, contextual syntax, and underlying logic of a passage. By applying this framework to the nuances of AI identification, the model can efficiently weigh structural and semantic disparities, making it highly adept at separating the subtle, hyper polished patterns of human prose from AI imitations.

A limitation with RoBERTa is the fact that it only utilizes 256 tokens with how the Hugging Faces API and RoBERTa itself are established in this project. This means that when working for both Chapter and Page document sizes there is a real possibility that not all the data is being accounted for, particularly in chapters as the maximum size of a page is set at 275 tokens, so the loss would be less significant. This constraint is unlikely to affect the paragraph document size due to the generally shorter size of paragraphs.

The inclusion of RoBERTa in this research is driven by its capabilities for high-dimensional classification through deep contextual embedding and its built-in nature as an advanced sequence classifier, optimized here through Hugging Face's fine-tuning. Due to the computational costs of RoBERTa and the general goals of this project it was only applied as a benchmark for state of the art models on the research conducted on the different document size datasets to best facilitate a comparison. All in all, RoBERTa offers a cutting-edge deep learning alternative to the classical and ensemble methods described above to provide an exceedingly robust approach to identifying AI imitations from the real novels.

3.10 Tuning of Hyperparameters

Several of the classification models utilized throughout this research contain hyperparameters that significantly influence model complexity, decision boundaries, and overall predictive performance. Unlike learned parameters, which are determined automatically during model training, hyperparameters must be selected prior to training and therefore require systematic optimization. To ensure that each model operated under conditions that maximized generalization performance while limiting unnecessary computational complexity, a structured hyperparameter tuning process was conducted using the validation dataset. The hyperparameter tuning methodology remained consistent across all document sizes. After the training, validation, and testing datasets were established, each tunable model was trained repeatedly using different combinations of predefined hyperparameter values. Following each training iteration, validation accuracy was calculated using the validation dataset. The hyperparameter configuration that achieved the greatest validation accuracy was then selected as the optimal configuration for the associated experiment. In situations where multiple hyperparameter configurations produced identical validation accuracies, preference was generally given to the computationally simpler configuration in order to reduce unnecessary model complexity and training cost.

For the Random Forest classifier, hyperparameter tuning focused on the number of decision trees included in the ensemble. Because Random Forest operates through bootstrap aggregation across multiple decorrelated decision trees, the number of estimators directly influences both variance reduction and computational cost. The tuning process evaluated forests containing 25, 50, 100, 200, 300, 400, and 500 trees. Each configuration was trained using the training dataset and evaluated on the validation dataset, with the configuration producing the highest validation accuracy selected for final testing. For Support Vector Machines, tuning focused on the selection of the kernel function. Kernel choice is particularly important in SVM classification because it determines how the feature space is transformed and whether the resulting decision boundary remains linear or becomes non-linear. Four kernel functions were evaluated throughout this research: Linear, Polynomial, Radial Basis Function (RBF), and Sigmoid kernels. Each kernel was independently trained on the TF-IDF feature matrix and evaluated using validation accuracy. The kernel that achieved the greatest validation performance was selected for the final model implementation associated with each experiment. The Neural Network required the most extensive hyperparameter tuning due to the flexibility of its architecture. The tuning process explored variations in hidden layer size, batch size, and training duration. Specifically, each hidden layer was tested using 8, 16, 32, 64, and 128 nodes with the potential for both layers to have the same number. Batch sizes of 4, 8, 16, and 32 were evaluated alongside training durations of 10 and 15 epochs. For each configuration, the network was trained using the ADAM optimizer and binary cross-entropy loss function, while validation accuracy was monitored throughout training. The configuration that produced the highest observed validation accuracy was selected as the optimal architecture for the associated document size.

Not all models required extensive hyperparameter tuning. Logistic Regression and Naive Bayesian

Classification were implemented largely using standard parameter configurations due to their relative simplicity and lower sensitivity to architectural variation. Similarly, RoBERTa relied primarily on the pretrained transformer architecture provided through the Hugging Face framework, with fine-tuning performed using standardized downstream training procedures rather than exhaustive architectural modification. This tuning framework ensured that each classification method was evaluated under optimized yet controlled conditions, thereby allowing performance comparisons between models to reflect meaningful differences in classifier behavior rather than poorly selected parameter configurations.

3.11 Limitations

Though the methodology of this research was carefully crafted, there are some key limitations that need to be addressed. Many limitations arise from the data collected, such as the relatively small amount of data. The eight selected novels provide a significant amount of data, but it is uncertain what types of data they can represent and generalize too. All the selected novels are famous English fictional novels from the 19th century. The primary constraint on the amount of data utilized in this project is the usage limits of the LLMs. Claude had a noticeably lower usage ceiling, which resulted in Claude imitation chapters being produced days after Gemini and ChatGPT had finished their portions of the eight novels. Additionally in the realm of data collection, a limitation and potential weakness could be around the prompting utilized. Several issues occurred throughout the prompting and AI collection process, but the most common revolved around the size of the AI imitation chapter. Despite having the expected word count stated in the prompt, ChatGPT on several observed occasions produced a shorter chapter, while Gemini on several occasions produced a longer chapter. This was mitigated when through a visual scan a chapter was clearly well outside the range, its word count was checked and if it was more than twice or less than a half the number of words of the average chapter mentioned in the prompt, it would be removed and remade. A similar limitation could be the prompt itself, and the unknown effects that prompting can have on the writing style of the LLM, which could be explored in a piece of research comparing non-imitation AI generated literary data against human data to determine if the results are the same.

Another section of limitations in this project is related to the models themselves. The scope of this project is limited to six models due to computational and time cost of expansion. This limitation is furthered by the lack of RoBERTa in the analysis of classifiers at distinguishing the LLM of origin and the leave-one-out LLM study were both similarly caused by the time and computational cost constraints. The final constraint regarding the models is the fact that RoBERTa is capped at 256 tokens per sequence during tokenization. This maximum length restriction, enforced via truncation, means that any textual data exceeding this limit was cut short, potentially omitting valuable contextual features from the longer documents in the dataset. This in particular affects chapter and page document sizes as both would often be longer than 256 tokens.

4 Results

4.1 Chapter Based Results

The original focus of this project and principal piece of analysis that the later sections have grown from is comparing the efficacy of various classification methods when distinguishing between the chapters of a novel and an AI imitation of that novel. As discussed in the methodology section, the classification methods being utilized are Logistic Regression, Random Forest, Support Vector Machines, Naive Bayesian Classifier, a Neural Network, and RoBERTa. Each model had its hyperparameters tuned

to maximize validation accuracy and was trained on the training dataset. The test accuracy from these models can be found on the table below.

Table 3: Detailed Class-wise Performance Metrics for Chapter Document Size Classification

Model	Class	Precision	Recall	Test Accuracy
Logistic Regression	False	71.00%	100.00%	79.55%
	True	100.00%	59.00%	
Random Forest	False	100.00%	100.00%	100.00%
	True	100.00%	100.00%	
Support Vector Machine	False	73.00%	100.00%	81.82%
	True	100.00%	64.00%	
Naive Bayesian Classifier	False	100.00%	100.00%	100.00%
	True	100.00%	100.00%	
Neural Network	False	92.00%	100.00%	95.45%
	True	100.00%	91.00%	
RoBERTa	False	100.00%	100.00%	100.00%
	True	100.00%	100.00%	

The most obvious result of this table is the excellent performance of Random Forest, Naive Bayesian Classifier, and RoBERTa with 100% test accuracy. The performance of RoBERTa is expected due to the large amount of pretraining and complex transformer architecture. The perfect accuracies from the Naive Bayesian Classifier and Random Forest are less expected due to being standard machine learning techniques without addition factors boosting them. The most interesting facet of these results come from the models that did not produce perfect scores. Across Logistic Regression, SVM, and the Neural Network all misclassified chapters were AI imitation chapters being identified as human. This false negative bias highlights the quality of these imitation as both SVM and Logistic Regression of the recalls of the true classes, which settle at 64% and 59% respectively. This trend carries to a lesser extent with the Neural Network as it has a true recall of 91% and a false recall of 100%. In other words, throughout all six models with a document size of chapter the tuned models are quite accurate and when they do misclassify a chapter they have a strong false negative bias.

As discussed in the hyperparameter tuning methodology section, each document size needs to undergo its own hyperparameter tuning. Not all of the classification models used require the application of hyperparameter tuning, with the most important being Random Forest, Support Vector Machine, and the Neural Network as discussed in the methodology section. The below table states the optimal hyperparameter values for each of the tuned models. This Random Forest model has 200 trees, which provides greater stability and is not a greater number of trees due to the policy of selecting the simplest method when multiple produce the same validation accuracy. The linear kernel was selected for SVM for much the same reason, as it had the same validation accuracy as the Sigmoid kernel. Finally, the Neural Network produced the greatest validation accuracy when the first layer had 64 nodes, the second had 32 nodes, and all of this was done with a batch size of 16.

4.2 Page Based Results

After completing the analysis on the chapter document size, to continue exploring the efficacy of these classification methods on novel based data the natural progression is to reduce the size of the documents,

Table 4: Table of Optimized Hyperparameter Values for Chapter Document Size Models

Model	Tuned Hyperparameter Values
Random Forest	200
Support Vector Machine	Linear
Neural Network	(64, 32, 16)

thus increasing the number of documents in the corpus. This was accomplished by splitting the text on a page-by-page basis. Page datasize presents a set of unique challenges and questions when compared to chapter data. The documents being short results in the TF-IDF vectors having less content to inform the models, but the limited number of chapters resulted also may have negatively impacted the training and testing of some models. A strength of page document size is that more data points will reduce the impact of specifically tricky misclassifications. All in all, the expectation is that some models will be hurt by the smaller TF-IDF vectors in regards to both training and testing, while others that misidentified specific characteristics will be less heavily impacted due to potential deciding factors for misclassification affecting a smaller overall percentage of the data.

Table 5: Detailed Class-wise Performance Metrics for Page Document Size Classification

Model	Class	Precision	Recall	Test Accuracy
Logistic Regression	False	80.00%	99.00%	86.83%
	True	99.00%	73.00%	
Random Forest	False	74.00%	94.00%	79.34%
	True	90.00%	64.00%	
Support Vector Machine	False	82.00%	99.00%	88.62%
	True	99.00%	77.00%	
Naive Bayesian Classifier	False	95.00%	92.00%	93.71%
	True	92.00%	95.00%	
Neural Network	False	90.00%	99.00%	94.01%
	True	99.00%	89.00%	
RoBERTa	False	100.00%	100.00%	100.00%
	True	100.00%	100.00%	

The above table highlights the change brought by the greater corpus size and lower document size. One major change is that only RoBERTa was able to maintain 100% test accuracy. This could be a combination of factors including the pretraining and the fact that it is simply the most sophisticated model. One of the most interesting aspects demonstrated in this table is that Logistic Regression and SVM performed better than on the chapter data. This suggests that the positive impact brought by having more data points to both train on and test against provides more benefit than the harm of reducing the document size for these models. Random Forest provides the largest change, falling approximately 21% down to 79.34%, indicates the extent of benefit the larger corpus provides. This could indicate that there are specific characteristics and patterns that Random Forest is utilizing that are exceedingly functional on a large document size, but less effective or ineffective on the smaller document size of page. The Naive Bayesian Classifier also decreased in test accuracy from 100% to 93.71%. This performance is still strong enough to indicate that the model is effective in this use case, but also shows a proverbial crack in the armor. The Neural Network performed approximately 1% worse on this document size compared to

chapter data. This continued high performance indicates an overall strong model that is continuing to effectively handle the classification problem. A major trend demonstrated in the table is that there is a continued False Negative bias that is present in Logistic Regression, Random Forest, SVM, and Neural Networks. This is shown by the low values in true recall that suggests that a weak point is incorrectly labeling True data as False, which is further expanded upon by being consistently the worse performing recall suggesting that it accounts for much of the misclassification rate.

The final point of consideration for the comparison of models with regard to a document size of page is the hyperparameter tuning for the 3 tunable models. To tune Random Forest, the hyperparameter of the number of trees was tuned. In this instance 500 trees produced the greatest validation accuracy and was also the greatest number of trees included as a tuning parameter. SVM selected the Sigmoid kernel, which in the chapter document size tuning exercise also received a perfect validation accuracy, but was not selected due to the added cost over the linear kernel. Finally, the Neural Network tuned to the greatest validation accuracy with a first layer of 16 nodes, a second layer of 64 nodes, and a batch size of 8.

Table 6: Table of Optimized Hyperparameter Values for Page Document Size Models

Model	**Tuned Hyperparameter Values**
Random Forest	500
Support Vector Machine	Sigmoid
Neural Network	(16, 64, 8)

4.3 Paragraph Based Results

To best understand the impacts of the inverse relationship of document size and corpus size in a fixed environment on the selected classification models, a third document size of a paragraph was selected. The splitting of data is done on a line break by line break basis with additional safeguards to remove paragraphs that are too short and to ward against errors. This document size presents similar challenges to those with a document size of a page, but with a few points of added complexity. The document size of a paragraph provides a more extreme comparison in the assessment of how models perform and if the larger corpus size can make up for the smaller size of documents. The smaller TF-IDF vectors will likely be a major source of misclassification due to the simpler input being fed into the models for both training and testing.

The above table denotes the impact that the smaller document size has on the selected models with an overall decrease in test accuracy across almost all models. Unlike in the prior two document size comparisons, RoBERTa fell from 100% accuracy at both other document sizes to 97.31% test accuracy. This is also the only model that managed to produce a test accuracy greater than 90%. Naive Bayesian and Neural Network both are major changes with a sizable decrease in test accuracy. SVM and Logistic Regression experienced minor decreases in test accuracy. The one model that performed better than when applied to a page document size is random forest, but the minor increase in test accuracy could be indicative of the model performing at essentially a baseline with the chapter document size being the abnormally effective instance. Another prevalent trend demonstrated in the above table is the false negative bias. This bias applies to Logistic Regression, Random Forest, SVM, and the Neural Network, while RoBERTa and the Naive Bayesian Classifier are more balanced or even potentially lean towards a False Positive Bias. Given this context, a false negative bias indicated that the four prior mentioned models have a tendency to classify AI imitation chapters as being from novels at a much greater rate than vice

Table 7: Detailed Class-wise Performance Metrics for Paragraph Document Size Classification

Model	Class	Precision	Recall	Test Accuracy
Logistic Regression	False	80.00%	89.00%	84.86%
	True	90.00%	81.00%	
Random Forest	False	72.00%	92.00%	80.10%
	True	91.00%	69.00%	
Support Vector Machine	False	79.00%	94.00%	85.23%
	True	93.00%	78.00%	
Naive Bayesian Classifier	False	86.00%	85.00%	86.69%
	True	87.00%	88.00%	
Neural Network	False	79.00%	89.00%	83.57%
	True	88.00%	79.00%	
RoBERTa	False	98.00%	96.00%	97.31%
	True	97.00%	98.00%	

versa, while the light false positive bias shown in RoBERTa and the Naive Bayesian Classifier indicates that the misclassifications can be often attributed to labeling text from real novels as AI imitations.

The last aspect to consider when discussing the comparison of the models on the text data with a document size of paragraph is the tuning of the hyperparameters. Random Forest, similar to on the chapter document size was, was determined to have an ideal number of trees in the forest of 200. Unlike with the prior document size, for this tuning 200 was selected due to having the greatest validation accuracy rather than being the least computationally expensive with the same validation accuracy. SVM selects a Radial Basis Function kernel that indicates the need of the infinitely high dimensional space that it facilitates. Finally, the neural network was found to produce the greatest validation accuracy from 8 nodes in the first hidden layer, 128 nodes in the second hidden layer, and a batch size of 8.

Table 8: Table of Optimized Hyperparameter Values for Chapter Document Size Models

Model	Tuned Hyperparameter Values
Random Forest	200
Support Vector Machine	Radial Basis Function
Neural Network	(8, 128, 8)

4.4 Exploration of Impacts of the Inverse Relationship of Document Size and Corpus Size on a Fixed Dataset

Large quantities of data have become increasingly in demand with the rise of data science and the ever increasing amount of computational power available. This is particularly apparent in the AI and LLM space with the ever increasing demand for all data, but this project focuses on textual data. As the amount of new and available data begins to decrease for these power users grasping for as much as possible, a potential method to address the data shortage is to change what we consider a full data point or document. This dataset contains a fixed amount of tokens, but the efficacy of the six selected classification models can be compared across different combinations of document and corpus size. As described in the above section, the document sizes utilized in this study are chapter, page, and paragraph. Each of these provides

distinct ratios and trade-offs to consider between the amount of documents in the corpus and the size of the documents themselves.

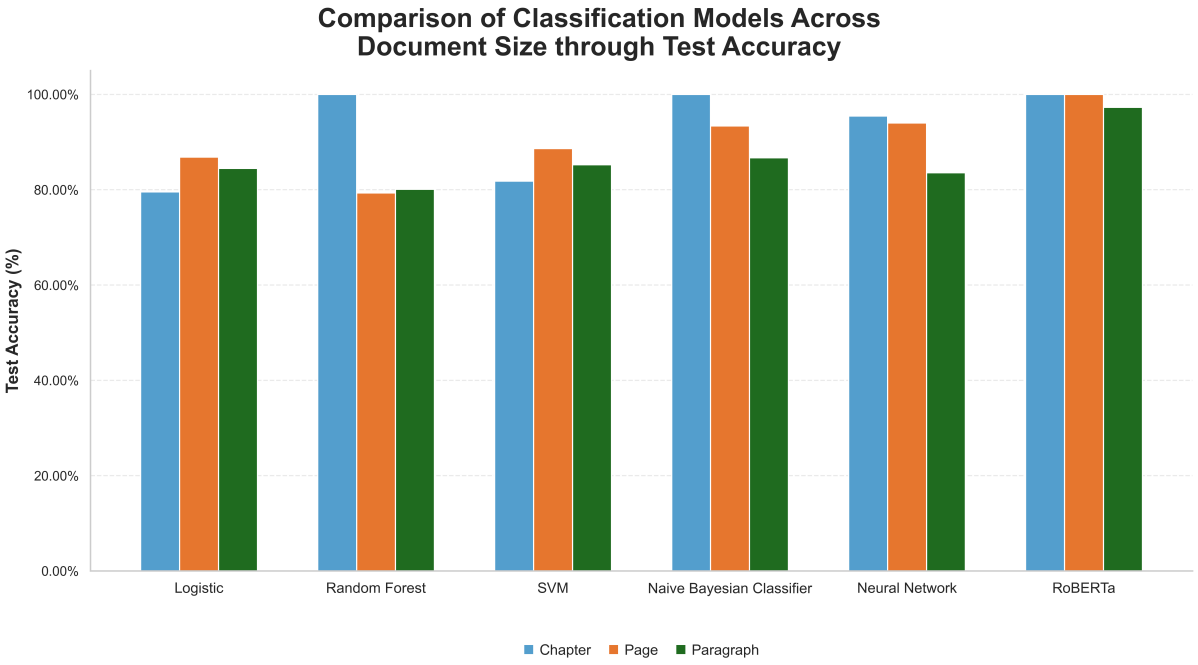

Figure 1: Bar Chart Visualizing the Test Accuracy across Classification Models across Document Size

The most basic observations to be found mirror the findings of the individual document sizes. RoBERTa is consistently tied for or the absolute best performing model. This could be due to the bidirectional transformer architecture taking in the context surrounding the words in addition to the vocabulary and word frequency of the documents, or it could be impacted by the use of pretrained parameter values making it less reliant on the data being used for training thus being impacted less by the lower document size associated with the page and paragraph models. The impact of the pretrained parameters values could also explain why RoBERTa with paragraph document size no longer has 100% test accuracy. A more extreme example of a model succeeding on sufficiently large document sizes can be found with both Random Forest and the neural network. Random Forest accurately labels each chapter of data, but for both other document sizes the test accuracy had a significant decline as it plateaued around 80% test accuracy. The impact of the relationship between document size and corpus size is even clearer on the neural network, as the neural network experiences a minor step down in test accuracy between the chapter and page document size. This is followed by a more significant decrease in test accuracy at the change from page document size to paragraphs. The last model with a similar trend of degradation as document size decreases in the Naive Bayesian Classifier. Without fail, the naive Bayesian classifier continues to produce a lower test accuracy with the increasingly small document sizes. Considering the results from the associated research, for Random Forest, Naive Bayesian Classifier, the Neural Network, and RoBERTa the benefit of a corpus size does not counteract the negatives associated with shrinking document size.

A deviation from the discussed above is that Logistic Regression and Support Vector Machines can have results with greater test accuracy when document size is smaller. Both Logistic Regression and SVM achieve their peak accuracy when the document size is a page, where they achieve greater test accuracy than either the chapter or paragraph document size. Additionally, both of these models produce

greater test accuracy from the paragraph document size data than from the chapter document size. This evidence leads to the premise that SVM and Logistic Regression benefit more from the greater number of data points from the larger corpus than they are hampered by the smaller document size, but the gains have a peak that produces the maximum test accuracy that is closer to the document size of pages than either of the other selected document sizes.

A significant trend across all three document sizes is the prevalent false negative bias, particularly in regard to Logistic Regression, Random Forest, SVM, and the Neural Network. When these models are misclassifying a document, regardless of its size, it will often be an AI imitation chapter that is being identified as from a real novel. The degree of this misclassification is tied to the overall test accuracy of the models at the given document size, so Logistic Regression and SVM have greater recall at the page document size.

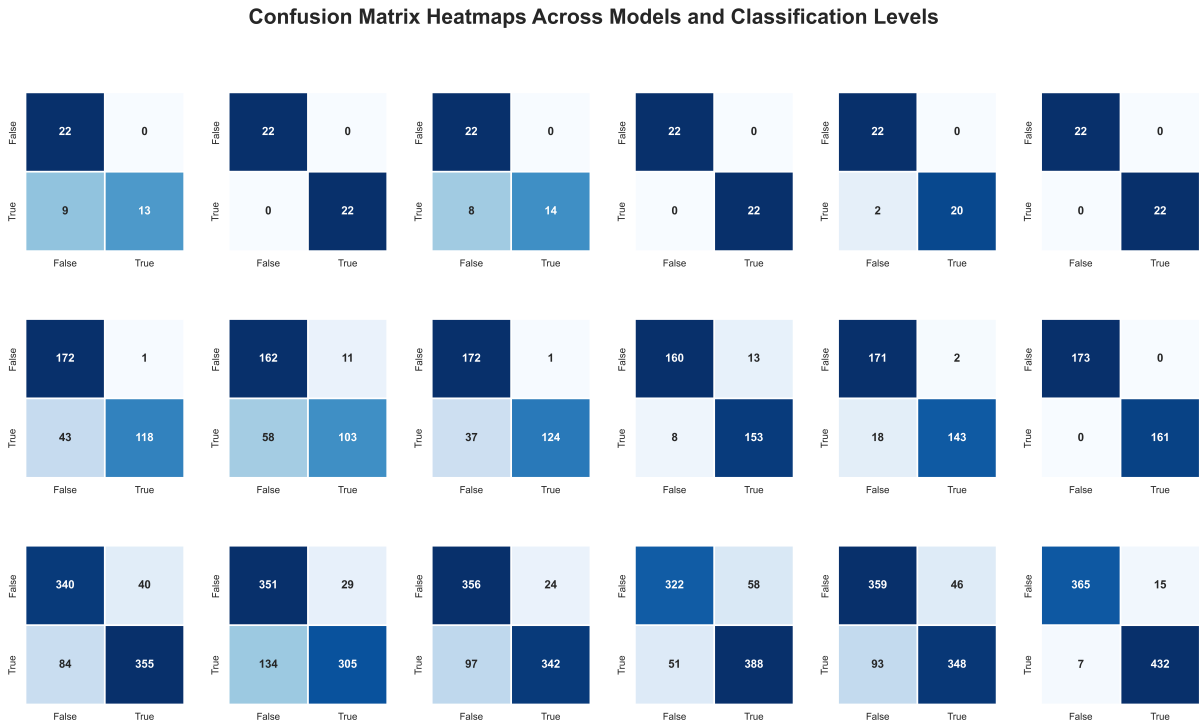

Figure 2: Confusion Matrices Visualizing Test Cases across Classification Model and Document Size

The above figure visualizes the confusion matrices for each model across each document size. Through this visualization, it is readily apparent when a model is functioning well that high performing models are characterized not only by greater overall test accuracy, but also by a more balanced distribution between false positives and negatives. RoBERTa demonstrates this most clearly, as its matrices remain strongly concentrated along the diagonal of correct identification across all three document sizes. Even when RoBERTa's performance decreases slightly at the paragraph level, the reduction is comparatively minor relative to the other models, suggesting that the contextual transformer architectures maintain robustness when document context is substantially reduced. Generally, weaker performing models have more asymmetric confusion matrices as document size decreases. Random Forest and the Neural Network provide particularly strong examples of this phenomenon. At the chapter level, both models maintain relatively balanced matrices with very few misclassifications, but at the page and paragraph levels the matrices become increasingly skewed towards false negatives. This indicates that these models retain the capability to identify authentic human writing while progressively struggling to recognize AI generated

content once the amount of contextual information available within each document decreases.

The confusion matrices also reveals that model degradation does not occur uniformly across classes, but instead most models specifically lose recall for the positive class as document size shrinks. This is especially true for Logistic Regression, Random Forest, SVM, and the Neural Network, where the lower-left quadrant of the matrices grows substantially larger at smaller document sizes. This trend suggests that shorter documents remove many of the stylistic or contextual signals that distinguish AI-generated writing, causing the classifier to default more frequently towards predicting documents as authentic human text.

Another significant aspect displayed in the figure is that the paragraph document size matrices contain substantially larger absolute errors, which appear despite a higher quantity of available training data. This further highlights the central trade-off explored in this study: increasing corpus size by reducing document size does not necessarily improve classification quality. Although paragraph segmentation provides many more training examples, the reduction in information appears to outweigh the statistical benefits of the expanded dataset at large. The visual progression from chapter to page to paragraph matrices demonstrates the additional data points alone are insufficient if the informational richness of each sample declines too far.

The last aspect of these models that needs to be discussed is the sensitivity to document length. Random Forest, Naive Bayes, and the Neural Network experience substantial degradation as document size decreases, indicating a heavy dependence on larger contextual windows. Logistic Regression and SVM demonstrate comparatively greater resilience and even modest improvements when moving from chapter to page sized documents, suggesting that these methods benefit more directly from the increase in available training samples. RoBERTa is unique as it maintains excellent classification performance regardless of document size and therefore exhibiting the greatest robustness to document size.

4.5 Comparison Of Results Across Different Random Seeds

A potential limitation of the prior analysis is its use on a single fixed random seed. While this aspect ensures the reproducibility across iterations, it also meant that the training, validation, and testing datasets were restricted to a single, specific permutation of novels. Consequently, there remains a valid concern that the observed performance may be the result of this specific data split rather than a reflection of a truly generalized trend. To determine whether the prior findings are genuinely representative of the underlying data a robust cross-validation stress test was conducted. The methodology followed was for each iteration, the dataset randomly reshuffled the novels assigned to the train and test split, while including the novel that would belong in the validation split in other sections of this analysis in the test data. Each classification model, tuned for chapter data, was retrained and evaluated on the newly generated split. This process occurred over 100 iterations in order to measure the consistency, stability, and variance of model performance under changing dataset configurations. The resulting distribution of accuracies were visualized in the following box plots, allowing for direct comparison of the stability of the selected models.

The boxplot visualizes the distribution of test accuracies obtained over the 100 iterations with chapter document size for Logistic Regression, Random Forest, Support Vector Machine, Naive Bayesian Classifier, and the Neural Network. RoBERTa was excluded from this piece of analysis due to its high computational cost. One of the clearest trends from the above model is the poor performance of Logistic Regression. Its median accuracy is substantially lower than the other models and the wide inter quartile range suggests that the performance was highly sensitive to the specific train-test

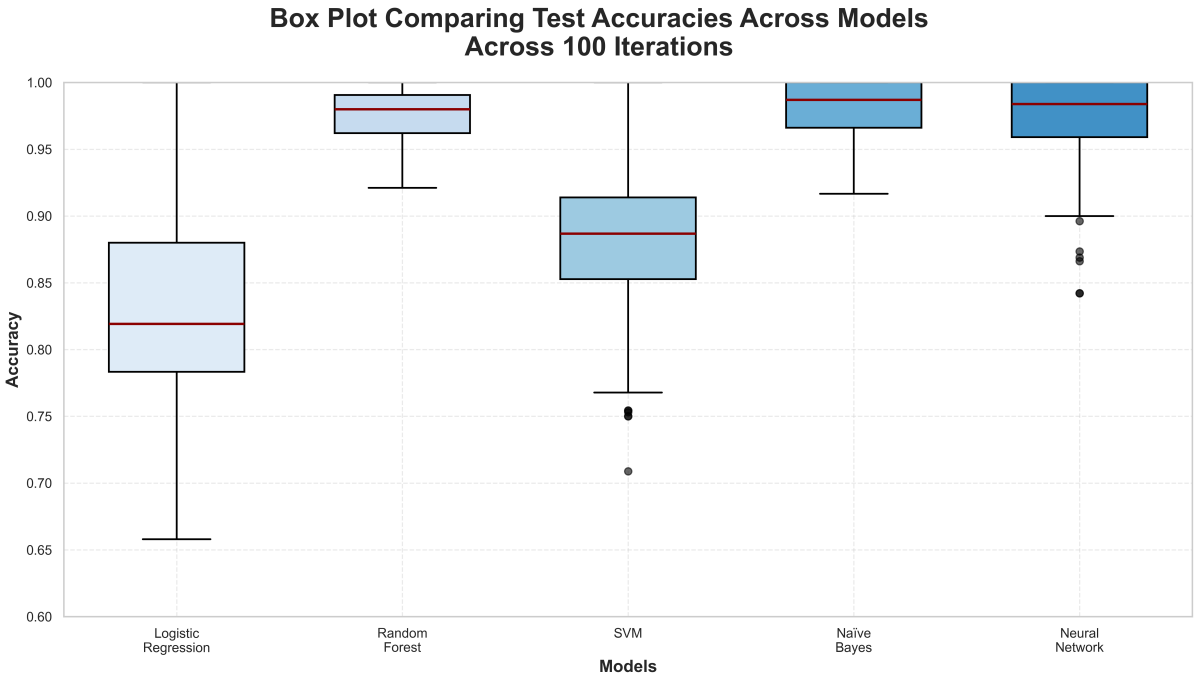

Figure 3: Box Plot Visualizing Comparison of Test Accuracies of selected Classification Models over 100 Iterations

split deployed. The presence of lower minimum values further suggests that the model struggled to generalize consistently across different dataset permutations. The SVM model achieved moderately strong performance exhibiting less variability than Logistic Regression, but more than any other model. While its median accuracy remained relatively high, the spread of the box plot and the presence of a notable low end outlier indicate occasional instability depending on the dataset split. This suggests that SVM performance was somewhat more sensitive to variations in the dataset.

In comparison, Random Forest produced significantly stronger and more stable results. The model achieved consistently high accuracies clustered near the upper range of the scale, with narrow inter quartile range indicating low variance between iterations. Although several lower outliers are visible, the overall distribution suggests that Random Forest generalized reliably across different random seeds. Similarly, the Naive Bayesian Classifier also has a high test accuracy median and low variability. Naive Bayesian Classifier in fact has one of the lower variance, and a high median on a range that is near 100% with at least the 25% of iterations achieving a perfect score. Although there are a small number of outliers, it maintains a strong predictive performance across iterations, indicating robust generalization capability. Finally, the Neural Network achieved very high median accuracy values, frequently approaching perfect classification performance. However, compared with the Naive Bayesian Classifier, the Neural Network produced a slightly wider spread and several lower outliers. This indicates that while the model was capable of exceptional performance, it was occasionally more affected by changes in the train-test split.

Overall, the repeated random-seed evaluation demonstrates that the strongest-performing models were not simply benefiting from favorable single data split. Instead, Naive Bayesian Classifier, Random Forest, and the Neural Network consistently maintain high accuracy across multiple dataset permutations, suggesting strong generalization performance. In contrast, Logistic Regression, and to an extent SVM, exhibit reduced robustness under variable training conditions.

4.6 Classification of LLM of Origin among AI Generated Content

Throughout this research, textual data sourced from three different LLMs have been utilized to assess different classification models capacity at distinguishing between the text of real human made novels and AI generated imitations. This section seeks to establish how well these same models can classify the AI imitation data based on the AI of origin. This provides a few benefits including assessing the parity between different LLMs' stylistic outputs, determining whether consistent linguistic characteristics remain detectable across generated literary text, and evaluating which machine learning approaches are most effective for model attribution tasks. While prior work has focused on distinguishing between human and AI-authored text, this section instead examines whether stylistic and lexical differences between individual LLMs remain detectable when generating the same type of content. Using TF-IDF vectorization of chapter-length novel imitations, the efficacy of multiple classification models can be compared in determining whether a generated chapter originated from ChatGPT, Claude, or Gemini.

Table 9: Detailed Class-wise Performance Metrics for LLM-Origin Classification

Model	**Class**	**Precision**	**Recall**	**Test Accuracy**
Logistic Regression	ChatGPT	100.00%	42.86%	63.64%
	Claude	46.67%	100.00%	
	Gemini	100.00%	50.00%	
Random Forest	ChatGPT	100.00%	100.00%	95.45%
	Claude	87.50%	100.00%	
	Gemini	100.00%	87.50%	
Support Vector Machine	ChatGPT	100.00%	42.86%	63.64%
	Claude	46.67%	100.00%	
	Gemini	100.00%	50.00%	
Naive Bayesian Classifier	ChatGPT	100.00%	100.00%	100.00%
	Claude	100.00%	100.00%	
	Gemini	100.00%	100.00%	
Neural Network	ChatGPT	100.00%	100.00%	100.00%
	Claude	100.00%	100.00%	
	Gemini	100.00%	100.00%	

The most immediate observation is that Naive Bayesian Classification and the Neural Network produced perfect test accuracy across all three classes, correctly identifying every generated chapter in the testing dataset. The Naive Bayesian Classifier additionally achieved perfect validation accuracy, suggesting that the TF-IDF distributions associated with each LLM contain highly separable probabilistic patterns. The effectiveness of Naive Bayes is particularly notable because TF-IDF representations inherently emphasize vocabulary frequency and distribution, features that align closely with the assumptions of probabilistic text classification. The Neural Network also achieved perfect test accuracy, which similarly suggests that the Neural Network is accurately identifying the non-linear relationship in the dataset.

Random Forest also demonstrates exceptionally strong performance, achieving 95.45% test accuracy. Unlike the probabilistic or neural approaches, Random Forest relies on ensemble decision boundaries that can capture non-linear interactions between TF-IDF features. The following confusion matrix indicates only a single misclassification, where one Gemini-generated chapter was identified as Claude. Nevertheless, the overall performance indicates that chapter-length TF-IDF representations preserve

sufficient stylistic information for highly accurate model attribution.

A substantial decline in performance can be observed for both Logistic Regression and Support Vector Machines. These models each achieved only 63.64% test accuracy. The confusion matrices reveal a strong tendency to over-predict the Claude class, resulting in perfect recall for Claude but substantially reduced recall for both ChatGPT and Gemini. Specifically, four ChatGPT chapters and four Gemini chapters were incorrectly classified as Claude by both models. This pattern suggests that the linear decision boundaries employed by Logistic Regression and SVM struggle to separate the stylistic characteristics of the generated chapters when the TF-IDF feature space becomes more complex. While these methods perform effectively in many binary text classification settings, they appear less capable of distinguishing nuanced stylistic variations between multiple AI language models.

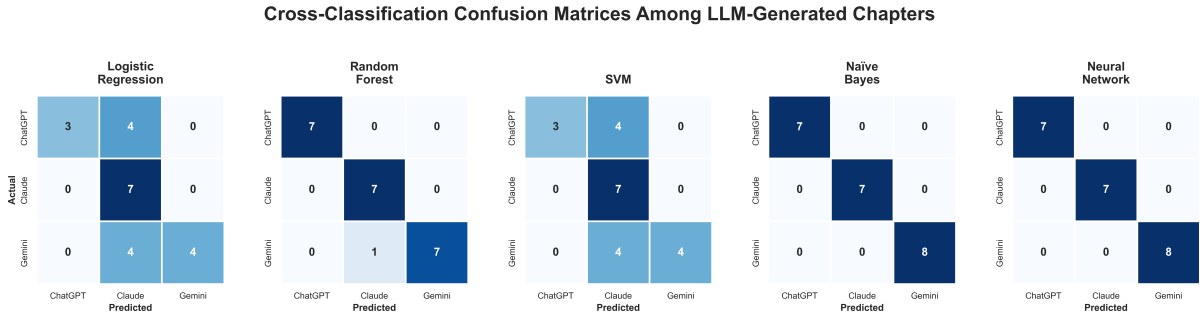

Figure 4: Confusion Matrices Visualizing the Classification Results of Selected Models on Identifying LLM of Origin of AI Data

The confusion matrices presented in the above visualization further reinforce the relationship between classifier performance and the distribution of prediction errors. Higher-performing models exhibit confusion matrices that are strongly concentrated along the diagonal, indicating that the majority of generated chapters were correctly attributed to their originating LLM. This pattern is particularly evident for Random Forest, Naive Bayesian Classifier, and the Neural Network. In the case of Naive Bayes and the Neural Network, the confusion matrices are entirely diagonal, demonstrating perfect separation between ChatGPT, Claude, and Gemini-generated chapters with no observable overlap between classes. Random Forest similarly displays strong diagonal dominance, with only a single off-diagonal error in which one Gemini chapter was incorrectly classified as Claude.

In contrast, Logistic Regression and SVM display substantially more asymmetric confusion matrices, revealing systematic rather than random classification failure. As shown in the figure above, both models overwhelmingly misclassify ChatGPT and Gemini samples as Claude, producing large concentrations of errors within the Claude prediction column. Specifically, four ChatGPT chapters and four Gemini chapters were incorrectly attributed to Claude by both classifiers. This explains the perfect recall achieved for the Claude class while simultaneously producing significantly lower recall for ChatGPT and Gemini. The visual similarity between the Logistic Regression and SVM confusion matrices also suggests that both linear classifiers encountered comparable limitations when operating within the TF-IDF feature space. Their decision boundaries appear insufficiently expressive to capture the more nuanced stylistic distinctions separating the three LLMs. Rather than producing evenly distributed misclassifications, these models collapse toward predicting Claude as a dominant or generalized class representation. This behavior implies that Claude-generated chapters may occupy a more central or overlapping region of the TF-IDF feature space, making them more difficult for simpler linear methods to distinguish from the outputs of other models.

The confusion matrix series therefore provides insight beyond overall accuracy metrics alone. While accuracy values quantify aggregate performance, the confusion matrices reveal the structural nature of each model's errors and demonstrate how effectively each classifier separates stylistic signatures across classes. The results collectively demonstrate that TF-IDF chapter representations preserve distinct stylistic signatures associated with different LLMs despite all samples being generated within the same literary domain. Although the generated chapters imitate novels and therefore share substantial thematic and structural similarities, measurable differences in lexical frequency, phrasing tendencies, and token distributions remain detectable through statistical learning methods. This is particularly evident in the strong diagonal dominance observed for Random Forest, NBC, and the Neural Network, where generated chapters are attributed to their originating models with near-perfect or perfect consistency. The consistently strong performance of these classifiers suggests that the stylistic fingerprints embedded within AI-generated literary text are sufficiently stable to permit highly reliable model attribution.

To conclude this branch of analysis, the tuning of the selected hyperparameters needs to be discussed. Like for other instances of hyperparameter tuning, the models focused on are Random Forest, Support Vector Machine, and the Neural Network. Through the tuning process, all number of trees utilized in random forest produced the same validation accuracy of 95.83%, so the selected hyperparameter value is 25 as it is the smallest tree tested. Similarly, SVM also had linear, sigmoid, and radial basis function kernels all produce 100% validation accuracy. This resulted in a linear kernel being selected due to its relatively cheaper computational cost. Finally, the neural network found the greatest validation accuracy when each of the two hidden layers had eight nodes, and it was run with a batch size of 4.

Table 10: Optimized Hyperparameter Values for LLM-Origin Classification Models

Model	**Tuned Hyperparameter Values**
Random Forest	25
Support Vector Machine	Linear
Neural Network	(8, 8, 4)

4.7 Analysis of Leave-One-Out LLM Training Efficacy across Models

While the preceding section established that classifiers can reliably distinguish between human and machine-authored text, as well as attribute a known AI-generated document to its specific originating LLM, a critical open question remains: how well do these models generalize to entirely unseen AI architectures? To evaluate this out-of-distribution robustness, a series of Leave-One-Out (LOO) experiments were conducted. In each iteration, data from one specific LLM was denoted as the entirety of the test data, while the other two LLMs were used entirely for the training phase. The classification models were then trained exclusively on a balanced dataset containing authentic human novels and imitation chapters from the remaining two LLMs. Finally, the models were evaluated against a test set of unseen novel chapters and associated AI imitations from the withheld LLM. This stress test isolates whether classifiers are learning a generalized representation of AI authored text or if they are merely overfitting the stylistic markers of specific models.

The empirical results, aggregated in the above Table, reveal a extreme false negative bias when evaluating classification models. For Logistic Regression, Random Forest, Support Vector Machines (SVM), and the Neural Network, the subsequent confusion matrices demonstrate a near-perfect or absolute zero false-positive rate. This means these classifiers almost never misclassify authentic human text as AI

Table 11: Detailed Class-wise Model Performance Metrics for Leave-One-Out (LOO) Unseen LLM Evaluation

Classification Model	Withheld LLM	Precision (True)	Recall (True)	Test Accuracy
Logistic Regression	ChatGPT	100.00%	51.00%	75.32%
	Claude	100.00%	59.00%	79.33%
	Gemini	100.00%	23.00%	61.33%
Random Forest	ChatGPT	100.00%	39.00%	69.48%
	Claude	100.00%	71.00%	85.33%
	Gemini	100.00%	24.00%	62.00%
Support Vector Machine	ChatGPT	100.00%	45.00%	72.72%
	Claude	100.00%	76.00%	88.00%
	Gemini	97.00%	39.00%	68.67%
Naïve Bayesian Classifier	ChatGPT	86.00%	71.00%	79.87%
	Claude	99.00%	96.00%	97.33%
	Gemini	97.00%	88.00%	92.67%
Neural Network	ChatGPT	100.00%	58.00%	79.22%
	Claude	100.00%	91.00%	95.33%
	Gemini	100.00%	49.00%	74.67%

generated. However, this clinical precision on the negative class comes at the cost of a severely degraded true positive recall rate. When an imitation chapter originates from an unseen LLM, these models heavily default to classifying it as a real human novel. This pattern implies that the linear decision boundaries and ensemble splits learned from the training LLMs fail to capture the unique lexical distribution of the withheld mode, causing the classifier to retreat to its baseline human textual model.

The degree of this degradation is intensely dependent on which specific LLM is withheld, exposing an asymmetry in stylistic overlap. Without exception, every single machine learning model achieved its peak LOO performance when Claude was the withheld class. This is punctuated by the Naive Bayesian Classifier's 97.33% test accuracy and the Neural Network's 95.33% accuracy. This strongly signals that the stylistic footprints, vocabulary distributions, and syntactic patterns generated by ChatGPT and Gemini form a robust boundary that naturally encompasses the text generated by Claude. Conversely, Gemini proved to be the most challenging out-of-distribution architecture to detect. When Gemini text was withheld, Logistic Regression and Random Forest fell to 61.33% and 62%, yielding recalls of just 23% and 24%. This drop indicates that Gemini Possesses unique structural or lexical signatures that are entirely unaccounted for in a training set consisting strictly of ChatGPT and Claude chapters.

An examination of architectural resilience highlights the Naive Bayesian Classifier as being particularly robust in this domain. In sharp contrast to its peers, Naive Bayes did not collapse into a highly asymmetric confusion matrix when challenged by unseen data. When ChatGPT was withheld, Naive Bayes achieved a balanced error distribution resulting in a 79.89% test accuracy, and it maintained an exceptional 92.67% accuracy on Gemini with an 88.00% recall. Because the Naive Bayesian Classifier relies on independent probabilistic word frequencies rather than geometric hyperplanes or hard binary decision trees, it appears uniquely capable of leaning on broader, high-level vocabulary characteristics that remain uniform across all LLM text architectures, making it highly effective at zero-shot AI author detection.

The Neural Network also exhibited compelling generalization, trailing just behind NBC. It proved

exceptionally strong during the Claude-withheld iteration, retaining a 91.00% recall. However, when confronted with the complex stylistic shifts of an unseen Gemini model, its recall dipped to 49.00% as it fell prey to the dominant false-negative trend. Logistic Regression and SVM matched each other almost perfectly in their behavior, highlighting their shared reliance on linear separation in high-dimensional TF-IDF space. Both models proved reasonably competent at identifying unseen Claude and ChatGPT texts but were thoroughly blind to Gemini, where their linear hyperplanes misclassified the vast majority of AI texts as human.

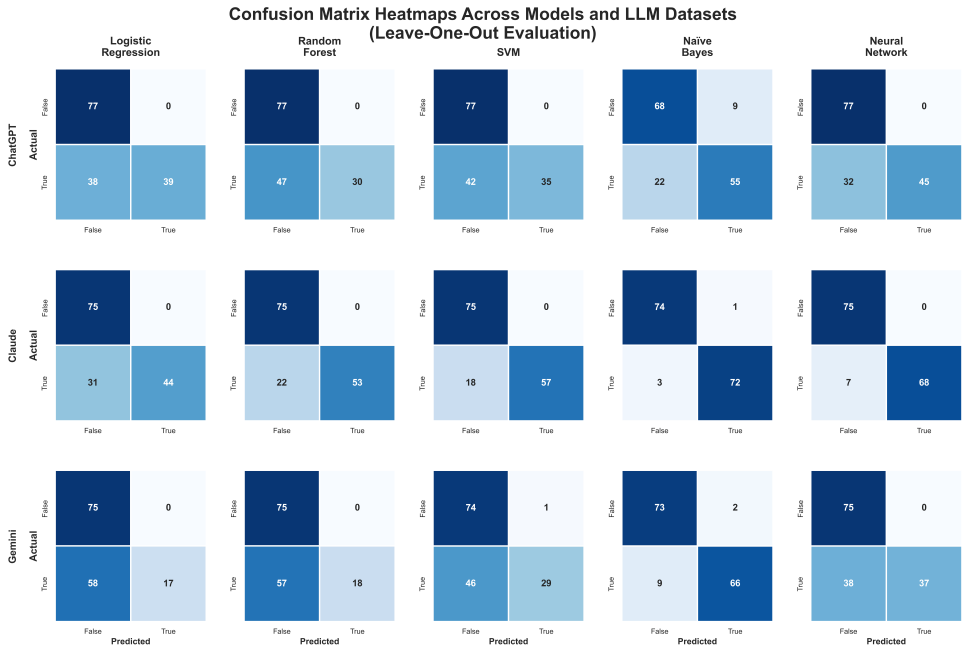

Figure 5: Confusion Matrices Visualizing the Classification Results across Classification Models and Unseen LLM

The confusion matrices visualizations in the above figure reinforce the trends observed in Table 11. Across all classifiers except the Naive Bayesian Classifier, the matrices exhibit a pronounced imbalance characterized by extremely low false-positive rates and substantially elevated false-negative counts. Visually, the upper-right quadrants are almost entirely empty, while the lower-left quadrants dominate the error distribution. This indicates that most models adopted an overly conservative decision strategy under Leave-One-Out conditions, strongly favoring classification into the human class whenever uncertainty arose from an unseen LLM distribution. This behavior is particularly evident in Logistic Regression, Random Forest, and SVM. While these models preserved near-perfect specificity across most experiments, they struggled severely to identify AI-generated chapters from withheld architectures. The Gemini-withheld condition produced the clearest collapse, where Logistic Regression and Random Forest misclassified 58 and 57 AI chapters respectively as human text. In contrast, the Claude-withheld matrices remained comparatively balanced across all models, suggesting that Claude-generated text shares stronger stylistic overlap with ChatGPT and Gemini outputs.

Ultimately, the LOO experiment underlines a foundational challenge for AI text detection systems. Although standard machine learning models can easily separate human text from a closed, known set of AI training data, their real-world utility is heavily throttled by an inherent architectural vulnerability to new generator models. The structural variance between outputs from architectures like ChatGPT and Gemini demonstrate that AI style is not a monolith, but rather a moving target that requires probabilistic

or highly generalized modeling frameworks to ensure long-term, cross-model classification efficacy.

5 Conclusion

This thesis sets out to evaluate the efficacy of multiple machine learning and deep learning classification methods in distinguishing authentic human-authored novels from AI generated literary imitations. Through the construction of a carefully controlled dataset consisting of canonical novels paired with stylistically constrained imitations generated by ChatGPT, Claude, and Gemini, this research established a rigorous benchmark for AI text detection within long-form creative writing. Across all experiments, the results demonstrate that AI-generated literary text contains identifiable statistical and contextual characteristics that remain detectable through both classical machine learning methods and modern transformer architectures. However, the findings also reveal that the effectiveness, robustness, and generalizability of these models vary substantially depending on document size, corpus structure, and exposure to unseen LLM architectures.

The most significant contribution of this thesis is the systematic exploration of the inverse relationship between document size and corpus size within a fixed textual dataset. While prior research in AI text detection often evaluates models at a single arbitrary granularity, this study instead demonstrated that document segmentation fundamentally alters classifier behavior. By restructuring the same fixed corpus into chapter, page, and paragraph sized documents, this research isolated the trade-off between informational richness per sample and the quantity of training instances available to the model. The findings reveal that increasing corpus size by reducing document size does not universally improve classification performance. Instead, the impact is highly dependent on the architecture of the classifier itself.

The results showed that Random Forest, Naive Bayesian Classifier, the Neural Network, and RoBERTa all experienced overall degradation as document size decreased. These models benefited more from the richer contextual information contain within larger chapter-length documents than from the statistical advantages associated with a larger corpus. In particular, Random Forest demonstrated an extreme dependence on larger contextual windows, falling from perfect chapter-level accuracy to approximately baseline performance on smaller document sizes. Similarly, the Neural Network and Naive Bayesian Classifier exhibited progressive decreases in accuracy as document size contracted, indicating that many of the lexical and stylistic patterns they relied upon became diluted once contextual continuity was reduced.

In contrast, Logistic Regression and Support Vector Machines demonstrated a markedly different relationship with corpus structure. Both models produced their most accurate results with the page document size and performed better on page and paragraph size than on the chapter sized data. This finding suggests that these simpler linear models benefited substantially from the increased number of available training examples generated through segmentation. However, this improvement was not unlimited. The paragraph-level results indicate that excessive reduction in the document size eventually removes too much contextual information even for models that benefit from a larger corpus. Consequently, this thesis demonstrates that there exists an optimal balance between document size and corpus size that differs across architectures rather than a universally superior segmentation strategy.

This contribution is particularly significant because it provides empirical evidence that dataset construction itself acts as a major determinant of AI text detection performance. The findings suggest that future detection systems cannot simply maximize dataset size without considering the informational

density of individual documents. In applied applications, this insight has important implications for moderation systems, educational integrity tools, and literary authenticity verification, where classifiers may encounter documents ranging from isolated paragraphs to complete manuscripts. The results of this thesis indicate that classifier selection should be informed not only by overall accuracy, but also by the expected granularity of the textual inputs.

A second major contribution of this research is the extensive analysis of cross-model generalization through the Leave-One-Out framework. While many AI detection studies evaluate performance using data generated by the same models present during training, this thesis specifically examined whether classifiers are capable of identifying fundamentally unseen AI architectures. This represents a substantially more realistic approximation of real-world deployment, where newly released LLMs may not yet be represented within existing training datasets.

The LOO experiments revealed a critical limitation of many conventional machine learning classifiers. Logistic Regression, Random Forest, Support Vector Machines, and even the Neural Network demonstrated severe false negative bias when evaluated against unseen LLMs. Rather than incorrectly labeling human writing as AI-generated, these models overwhelmingly misclassified unseen AI generated text as authentic human literature. This pattern demonstrates that many classifiers are not truly learning universal characteristics of the AI imitations, but are instead partially overfitting to the stylistic signatures of the specific models included in training. The asymmetry between withheld LLMs further reinforces this conclusion. Claude consistently proved to be the easiest unseen architecture to detect, while Gemini was significantly more difficult across nearly all classifiers. This finding suggests that modern LLMs do not produce a singular AI style, but instead generate distinct lexical and structural profiles that vary meaningfully between models. The implication is substantial: AI text detection is not a static binary classification problem, but rather a continuously evolving adversarial environment in which new models may evade detectors trained exclusively on prior architectures.

Among all evaluated methods, the Naive Bayesian Classifier emerged as the most robust conventional machine learning framework for zero-shot AI detection. Unlike the geometric decision boundaries employed by Logistic Regression and SVM or the partitioned features spaces of Random Forest, NBC maintained comparatively balanced confusion matrices and strong recall even on unseen LLMs. This suggests that probabilistic frequency based modeling may capture broader linguistic characteristics shared across AI generated text more effectively than methods dependent on fixed hyperplane separators. The Neural Network also demonstrated comparatively strong generalization, although it remained vulnerable to the major false negative results found from when Gemini was unseen. These findings contribute important insight into the future design of AI detection systems. Specifically, they indicate that strong in-distribution accuracy alone is insufficient for evaluating practical effectiveness. A classifier capable of perfectly identifying known generators may still fail catastrophically against newly developed architectures. Consequently, future AI detection research must increasingly prioritize cross-model robustness and out-of-distribution evaluation rather than relying solely on closed-set benchmarks.

Beyond the primary contributions regarding document segmentation and LOO evaluation, this thesis also demonstrated several broader findings regarding AI-generated literary text. First, the consistently high performance achieved by multiple models at chapter-level classification confirms that modern LLM-generated novels retain detectable statistical fingerprints despite sophisticated prompting and stylistic imitation constraints. Even when instructed to imitate canonical authors while preserving narrative settings and character structures, the generated text remained distinguishable from authentic literature. Second, the experiments attributing text to specific LLMs demonstrated that individual models possess distinct stylistic signatures that persist even within constrained literary generation tasks. The near-

perfect attribution performance achieved by several classifiers suggests that AI-generated prose retains measurable architectural identity despite thematic overlap.

RoBERTa consistently emerged as the strongest overall model throughout the experiments, maintaining exceptional performance across nearly all document sizes. Its transformer architecture and pretrained contextual representations appear substantially more robust to shrinking document context than traditional TF-IDF-based classifiers. Nevertheless, even RoBERTa experienced measurable degradation at the paragraph level, indicating that severe reductions in contextual information remain challenging even for advanced contextual transformers.

Taken together, the results of this thesis demonstrate that AI-generated literary text detection is both highly achievable and fundamentally constrained by issues of generalization and document structure. The work contributes a comprehensive benchmark framework spanning multiple document granularities, multiple classifier architectures, multiple LLM generators, repeated random-seed validation, and zero-shot leave-one-out testing. Most importantly, it establishes that document segmentation and cross-model robustness are not secondary implementation details, but foundational variables that directly determine classifier efficacy.

There are many avenues for expansion available from this research. Some next steps for this project could be exploring sentence or full novel document sizes, utilizing different embedding methods, increasing the number of novels included in the dataset, and increasing the number of LLMs utilized. Alternative avenues of expansion are utilizing hybrid systems that combine probabilistic lexical analysis with contextual transformers to improve out-of-distribution robustness. Finally, as generative AI systems continue to evolve, broad scope evaluation across successive generations of LLMs will become increasingly necessary to determine whether universal indicators of AI-generated writing truly exist.

Ultimately, this thesis demonstrates that the challenge of AI text detection extends beyond merely separating human and machine-authored prose. Instead, the problem encompasses issues of scalability, contextual dependency, architectural generalization, and the evolving diversity of generative models themselves. By systematically analyzing these dimensions, this research provides both a practical benchmark and a conceptual framework for future studies seeking to preserve authenticity, attribution, and trust within literary and creative domains increasingly shaped by artificial intelligence.

6 Appendix

6.1 AI Prompts

- "I am doing a project that needs you to create an imitation chapter of [Novel Title] by [Author]. An average chapter length of [Novel Title] is [Words in selected chapter] words, so your chapter should have a word length between from [words $-$ 0.1 $\times$ words] words and [words $+$ 0.1 $\times$ words] words (plus or minus 10%). This should be in the style of [Author] and character and setting names should generally be from the story."
- Repeat the prompt and carry this out again. I am doing a project that needs you to create an imitation chapter of [Novel Title] by [Author]. An average chapter length of [Novel Title] is [Words in selected chapter] words, so your chapter should have a word length between from [words $-$ 0.1 $\times$ words] words and [words $+$ 0.1 $\times$ words] words (plus or minus 10%). This should be in the style of [Author] and character and setting names should generally be from the story.
- That is the same chapter. Repeat the prompt and carry this out again. I am doing a project that needs you to create an imitation chapter of [Novel Title] by [Author]. An average chapter length of [Novel Title] is [Words in selected chapter] words, so your chapter should have a word length between from [words $-$ 0.1 $\times$ words] words and [words $+$ 0.1 $\times$ words] words (plus or minus 10%). This should be in the style of [Author] and character and setting names should generally be from the story.

6.2 AI Contribution Statement

Throughout this project generative AI was used in several ways. ChatGPT, Claude, and Gemini were used in the data creation process. This was a crucial point in the project. ChatGPT was also used for code assistance. The most significant use was to help with bug fixes and errors that were not resolved after an extended attempt. The other use was with quickly compiling visualization, but those were further refined and standardized individually for the final report. The final use of ChatGPT and Claude in this project is to review my thesis and point out weaknesses and structural issues. ScopusAI was additionally used in the search for supporting evidence and to better understand the topics. All derivations, analysis, and conclusions are the authors own work.

7 List of References

7.1 Data Sources

[2] Jane Austen. *Pride and Prejudice*. `https://www.gutenberg.org/ebooks/1342`. Project Gutenberg, 1813.

[6] Lewis Carroll. *Alice's Adventures in Wonderland*. `https://www.gutenberg.org/ebooks/11`. Project Gutenberg, 1865.

[9] Charles Dickens. *Hard Times*. `https://www.gutenberg.org/ebooks/786`. Project Gutenberg, 1854.

[10] Arthur Conan Doyle. *A Study in Scarlet*. `https://www.gutenberg.org/ebooks/244`. Project Gutenberg, 1887.

[21] Mary Wollstonecraft Shelley. *Frankenstein; or, the modern prometheus*. https://www.gutenberg.org/ebooks/84. Project Gutenberg, 1818.

[22] Robert Louis Stevenson. *The Strange Case Of Dr. Jekyll And Mr. Hyde*. https://www.gutenberg.org/ebooks/43. Project Gutenberg, 1886.

[23] Robert Louis Stevenson. *Treasure Island*. https://www.gutenberg.org/ebooks/120. Project Gutenberg, 1883.

[24] Mark Twain. *The Adventures of Tom Sawyer*. https://www.gutenberg.org/ebooks/74. Project Gutenberg, 1876.

7.2 AI Tools

[1] Anthropic. *Claude Sonnet 4.6 AI language model*. https://claude.ai. Used for AI-assisted text generation and analysis between April 2 and April 16, 2026. The free subscription was used. 2026.

[12] Google. *Gemini 3 Flash AI language model*. https://gemini.google.com/app. Used for AI-assisted text generation and analysis between April 2 and April 9, 2026. The free subscription was used. 2026.

[18] OpenAI. *ChatGPT GPT-5.3 AI language model*. https://chatgpt.com. Used for AI-assisted text generation and analysis between April 2 and April 8, 2026. The free subscription was used. 2026.

7.3 Scholarly References

[3] David Baidoo-anu and Leticia Owusu Ansah. "Education in the Era of Generative Artificial Intelligence (AI): Understanding the Potential Benefits of ChatGPT in Promoting Teaching and Learning". In: *Journal of AI* 7.1 (2023), pp. 52–62. DOI: 10.61969/jai.1337500. URL: https://izlik.org/JA84JM98SM.

[5] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

[7] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine Learning* 20.3 (Sept. 1995), pp. 273–297. ISSN: 1573-0565. DOI: 10.1007/BF00994018. URL: https://doi.org/10.1007/BF00994018.

[8] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[13] Andreas Lindholm et al. *Machine Learning - A First Course for Engineers and Scientists*. Cambridge University Press, 2022. URL: https://smlbook.org.

[14] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL]. URL: https://arxiv.org/abs/1907.11692.

[15] Alfira Makhmutova et al. “Testing the Limits: Evaluating AI Detectors’ Accuracy and the Impact of Obfuscation Techniques on AI-Generated Text”. In: *Journal of Advances in Information Technology* 17.3 (2026), pp. 438–449. DOI: 10.12720/jait.17.3.438-449.

[17] Andrew Mccallum and Kamal Nigam. “A Comparison of Event Models for Naive Bayes Text Classification”. In: *Work Learn Text Categ* 752 (May 2001).

[26] Yue Zhang and Zhiyang Teng. *Natural Language Processing. a Machine Learning Perspective.* Cambridge University Press, 2021.

7.4 Technical Tools and Software

[4] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit*. https://www.nltk.org/book/. O’Reilly Media Inc., 2009.

[16] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[19] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[20] Guido van Rossum and Jelke de Boer. “Interactively testing remote servers using the Python programming language”. In: *CWI Quarterly* 4.4 (Dec. 1991), pp. 283–304.

[25] Thomas Wolf et al. “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.