

**Department of the Air Force  
&  
Massachusetts Institute of Technology  
AI Accelerator**

# **AI Security Classification**

**Co-Author:  
Lei Hamilton**

**Author:  
Andrew Gelbard**

**Co-Author:**

Disclaimer: Research was sponsored by the Department of the Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

---

**March 2025  
Cohort 12**

# Artificial Intelligence for Derivative Security Classification: Applications to DoD

Andrew Gelbard  
MIT Phantom Fellow  
Air Force MIT AI Accelerator  
Cambridge, MA  
andrew.gelbard@us.af.mil

Dr. Lei Hamilton  
MIT  
Department of the Air Force  
Cambridge MA  
lei.hamilton@ll.mit.edu

**Abstract**—The accurate classification of government documents according to their sensitivity (e.g., *UNCLASSIFIED*, *SECRET*, *TOP SECRET*) is critical for national security, yet historically has relied on time-intensive manual review. The current manual classification process consumes millions of labor hours annually within the U.S. government, significantly diverting skilled personnel from essential analytical tasks. This research explores automating this security classification task using recently available declassified materials from the DISC dataset [1], addressing practical challenges such as noisy Optical Character Recognition (OCR) output, imbalanced data distributions, and potential leakage of explicit classification markers within document text. This dataset contains declassified government documents sourced from the Digital National Security Archive, providing authentic textual examples representative of actual classification scenarios. We evaluate both traditional machine learning approaches and advanced transformer-based language models to classify documents accurately across multiple sensitivity levels. Our results highlight that transformer-based models, particularly DeBERTa, effectively improve identification of the minority but critical *TOP SECRET* class, achieving recall over 70% and an overall balanced performance (macro  $F_1$  score of 0.75), while traditional methods exhibit similar overall accuracy but struggle with minority class recall. Despite promising findings, we caution that conclusions drawn here remain constrained by limited training data size and inherent uncertainties in human-labeled documents. We emphasize the need for larger, rigorously preprocessed datasets and suggest future research integrating authoritative classification guidelines directly into model training, potentially via retrieval-augmented methods. This work thus contributes a foundational, reproducible framework that demonstrates significant potential for machine-assisted security classification, guiding future research and practical applications in the information security domain.

## I. INTRODUCTION

Government and military organizations rely on rigorous information security *classification* systems to control access to sensitive materials. Classification levels (such as Unclassified, Confidential, Secret, Top Secret) are assigned based on the sensitivity and potential impact of disclosure. For example, the U.S. employs a system defined by Executive Order 13526 that standardizes how national security information is categorized, safeguarded, and declassified. Proper

classification ensures that sensitive data is handled in compliance with regulations and that only authorized personnel can access certain documents. Traditionally, determining and reviewing these classifications is a manual process carried out by trained Original Classification Authorities (OCAs) and declassification reviewers. However, given the growing volume of documents and the complexity of modern information environments, there is a strong motivation to automate or assist in the classification process using artificial intelligence techniques. Early research efforts applied statistical and machine learning methods to text classification problems in the security domain [2], [3]. Simpler approaches included keyword matching or dictionary-based rules (e.g., “dirty word lists” of sensitive terms) [4]. More advanced studies explored classical machine learning algorithms. For instance, Brown and Charlebois developed one of the first automated systems for security label assignment using natural language processing and statistical learning [2]. Alzhrani et al. (2016) applied Support Vector Machines and other classifiers to large text corpora for security classification tasks, reporting modest success in identifying sensitive documents. These earlier works demonstrated the feasibility of automatic document classification but often struggled with high false negative rates (failing to catch some sensitive documents) or relied on limited features. In recent years, the field of natural language processing (NLP) has been revolutionized by deep learning models such as BERT and its derivatives [5], [6], [7]. In particular, transformer-based language models such as BERT (Bidirectional Encoder Representations from Transformers) have achieved state-of-the-art performance on a wide range of text classification tasks. BERT can capture contextual nuances of language better than traditional bag-of-words or TF-IDF models, which is especially useful for complex domain-specific texts. Variants and improvements upon BERT have also emerged. DistilBERT is a compressed version of BERT that retains much of its accuracy while using fewer parameters and faster inference. More recently, the DeBERTa model (Decoding-Enhanced BERT with Disentangled Attention) introduced archi-

textural enhancements that improved performance on many benchmark tasks [7]. These transformer models have not yet been extensively applied to the problem of government document security classification, which often involves long, technical documents with domain-specific terminology. There is also interest in leveraging even larger generative models and few-shot learning for classification, but those remain largely untested in this particular domain. Another critical development for research in this area is the creation of suitable datasets, exemplified by the DISC dataset [1]. Due to the sensitive nature of classified information, open datasets for this task have been practically nonexistent. Recently, Bass *et al.* (2024) introduced the Dataset for Information Security Classification (DISC), a collection of declassified government documents along with their original highest classification labels. DISC is a valuable resource because it enables research on real (formerly classified) documents without confidentiality concerns, and it provides a benchmark for developing and evaluating automated classifiers. Our work uses the DISC dataset as the basis and testbed for experimentation. In this paper, we aim to build effective classification models on DISC and to examine whether modern large language models can outperform traditional approaches in this context. We also explicitly address some practical issues that arise in DISC, including noisy text from OCR and highly imbalanced class frequencies, as discussed below.

## II. BACKGROUND

### A. Purpose and Significance of Security Classification in DoD and the IC

Security classification is a cornerstone of information security within the U.S. Department of Defense (DoD) and Intelligence Community (IC). It provides a structured system to identify and protect sensitive information that, if disclosed without authorization, could harm national security [8]. The United States uses three primary classification levels – Confidential, Secret, and Top Secret – corresponding to the severity of damage that unauthorized release of the information would cause [9]. In practice, this means that military plans, intelligence reports, and other critical data are marked and handled according to their sensitivity, ensuring that only appropriately cleared personnel can access them. By maintaining certain information “in confidence,” the DoD and IC safeguard citizens, military personnel, and ongoing operations from adversaries who could exploit leaked information [8]. In essence, classification is vital to mission success: it protects sources and methods in intelligence, preserves tactical surprise and troop safety in military operations, and upholds trust with allies who share secret information under the promise it will remain secure.

At the same time, the classification system serves to communicate risk and handling requirements to

everyone who deals with national security information. Markings on documents (like “SECRET” or “TOP SECRET”) signal the level of protection needed, guiding how information is stored, transmitted, or discussed. This uniform system (governed by executive order and DoD policies) ensures that users of classified information treat it consistently and make informed decisions about safeguarding or sharing it. A responsible classification program underpins not only security but also effective information sharing – striking a balance so that while secrets are protected, necessary information flows to those who need it for the mission [8]. If classification is misapplied (for example, classifying too low, or failing to classify something that is sensitive), the mission impact can be severe: under classification can lead to leaks that compromise operations, while overclassification can hinder coordination and delay critical information reaching commanders or analysts. Thus, proper classification is not just a bureaucratic mandate, but a mission-critical function in the DoD and IC, directly supporting national security objectives.

### B. Classification Workload and Associated Costs in DoD/IC

The scale of classification activity across the DoD and IC is enormous, reflecting the breadth of U.S. national security operations. Millions of classification decisions are made each year as officials and analysts create or update classified materials. In fact, government-wide estimates indicate about 50 million new classified documents are generated annually [10]. The DoD and IC – which include major classifiers like the armed services, combatant commands, NSA, CIA, and other intelligence agencies – account for the majority of this workload. These organizations continuously produce classified plans, intelligence reports, technical assessments, and communications, contributing heavily to the tens of millions of classified records created every year. Handling such a volume of secrets is a labor-intensive endeavor. Over four million U.S. government personnel and contractors hold security clearances [10], and many of them participate in classifying information as original or derivative classifiers. This translates to a substantial amount of staff time devoted to applying classification markings, writing classification guides, reviewing documents for release, and managing classified systems. For example, if each of those 50 million new documents requires even a few minutes of a classifier’s time, the aggregate manpower consumed is on the order of several million labor hours per year. This is time that highly skilled analysts, engineers, or military officers spend on administrative security tasks in addition to their primary duties.

The financial cost of maintaining the classification system is correspondingly high. Beyond the manpower devoted to classification itself, DoD and IC agencies incur expenses for security clearances, spe-

cialized IT systems, secure facilities, and monitoring needed to protect classified information. According to data reported to the Information Security Oversight Office, the government-wide security classification program (excluding some intelligence agencies) was costing taxpayers on the order of \$18 billion per year in the late 2010s [10]. This figure includes costs for personnel security, physical protections, classified networks, and information management required to handle secret information safely. Notably, DoD as the largest department – with its extensive classified networks and thousands of secure sites – and the IC (with agencies whose security budgets are partly classified themselves) consume a large share of these resources. In a 2023 congressional hearing, experts noted that \$18 billion is likely an underestimate of the true classification-related costs, since the expenses of certain intelligence agencies are omitted from public reports and the overall burden has been growing [11][12]. Even so, at face value this equates to billions spent annually on securing classified information, which includes not just technology and infrastructure but also the human labor of classifying, safeguarding, and later declassifying documents. In short, the DoD/IC classification workload is immense in volume and in dollars: tens of millions of classified records each year, maintained by millions of cleared personnel, at a cost of tens of billions of dollars. This context underscores how critical it is for these agencies to manage classification efficiently – any improvements or reforms can yield significant resource savings and better allocation of personnel to mission-focused tasks.

### *C. Efficiency Gains Through Automation of Classification*

Given the magnitude of the classification workload, there is a growing recognition in government that modernizing and automating parts of the classification process is essential. Current practices rely heavily on manual effort – individuals must decide classification levels, apply markings, and review documents one by one – a process that has struggled to keep up with the digital age’s information volumes. An oversight report observed that classification and declassification actions are still performed manually, which is neither sustainable nor desirable in the digital age [13]. The traditional system, largely unchanged for decades, is increasingly seen as inefficient, costly, and prone to error when faced with today’s high-speed data generation [13]. For example, a classifier might spend valuable time combing through lengthy documents to identify sensitive paragraphs, or multiple reviewers might sequentially handle the same file to ensure it’s properly marked. Across DoD and IC workflows, these repetitive activities can delay information sharing and tie up analysts who could otherwise be focusing on intelligence or operational analysis. In short, the opportunity cost of so much manual classification is

significant.

Automating classification processes – for instance, using artificial intelligence (AI) or machine-learning tools trained to recognize sensitive content – promises to greatly improve efficiency. By delegating routine or well-defined classification tasks to software, agencies could speed up the processing of documents and reduce the manpower required for initial classification and subsequent reviews. One quantitative way to appreciate this potential is to consider the earlier figure of 50 million classified documents per year. If an automated system could assist with or outright classify even a portion of these (say, 20–30%), it might save millions of person-hours annually that are currently spent on manual review. Those hours could be redirected to analysis, decision-making, or other mission functions. Indeed, federal modernization initiatives have explicitly recommended investing in advanced technology for classification. The Public Interest Declassification Board and other advisors have urged the government to leverage AI and machine learning to help manage classified information more effectively [14]. By using automation, the DoD and IC can process the growing volume of digital data faster and more consistently, which means shorter turnaround times for producing intelligence reports or disseminating operational orders with proper classification. Automation can also reduce backlogs – for instance, speeding up classification reviews for documents that need to be shared with partners or downgraded for wider circulation. Furthermore, machine-assisted classification is inherently consistent: an algorithm applying classification rules uniformly can mitigate the variability between different human classifiers (who might overclassify out of caution, or underclassify due to oversight). In sum, automation offers to streamline the classification workflow, yielding quantitative savings in time and labor. This increased efficiency not only lowers costs but also enhances mission agility by getting information to the right people faster, all while maintaining required security controls.

### *D. Security Enhancements from Automated Classification*

Beyond efficiency, automation brings significant security benefits to classification and information protection in the DoD and IC. One key advantage is the ability to continuously monitor and enforce classification standards across vast networks, something humans cannot do at scale. Automated classification tools can be integrated with security systems to detect and flag classified content wherever it appears, acting as a constant sentinel against mishandling or leaks. For example, if classified text or keywords are accidentally introduced into an unclassified environment (such as an email or a shared drive), an automated system can immediately recognize the error and alert security personnel or block the transmission. In fact, this kind

of capability has already been demonstrated: a national laboratory deployed a system to scan outgoing emails for classified information on unclassified networks and catch any sensitive data before it escapes secure channels [15]. Such technology – often part of “data loss prevention” (DLP) solutions – can drastically reduce the risk of spillage, which is the inadvertent placement of classified content on insecure systems. In the DoD/IC context, where networks of differing classification (Unclassified, Secret, Top Secret, etc.) are strictly segregated, automated detection of misfiled or mis-sent information is a critical safety net. It enables a much faster response to contain a potential breach, often before any adversary has a chance to access the data. This is a clear improvement over relying solely on human vigilance, where mistakes might go unnoticed for days or weeks.

Automated classification also enhances security by improving accuracy and consistency in applying classification rules. Human classifiers, pressed for time or lacking perfect knowledge of complex classification guides, might misclassify a document – either marking it too low (which could lead to a leak) or too high (which over-restricts information sharing). An AI-based classification assistant can act as a second pair of eyes, checking content against vast training data of what is typically Secret or Top Secret, and can either suggest the correct classification or catch discrepancies. This consistency helps ensure that truly sensitive information is always recognized and protected, while information that doesn’t warrant classification isn’t unnecessarily hidden. Over time, such tools could also learn from past classification decisions and help identify patterns of overclassification, supporting efforts to dial back excessive secrecy that can impede operations. Moreover, by automating the mundane aspects of marking and tracking classified data, security personnel can focus on higher-level threat analysis and insider risk mitigation. They’ll spend less time on clerical checks and more on proactive security enforcement. Importantly, automation can facilitate auditing and accountability: every automated classification decision can be logged and reviewed, making it easier to audit how information is being classified and shared. This transparency can strengthen trust in the system’s integrity. Ultimately, enhancing the classification system with smart automation fortifies the DoD and IC’s security posture – it not only prevents accidental leaks but also acts as a force multiplier for personnel, enabling them to guard secrets more effectively across the complex, fast-paced information environment of modern defense and intelligence operations.

### III. PROBLEM STATEMENT

The goal of this research is to automatically determine the level of security classification of a document based on its textual content. Specifically, given a

declassified document (initially labeled as *Unclassified*, *Confidential*, *Secret*, or *Top Secret* at the time of its creation), our system must predict its highest classification label. This is essentially a three-class text classification problem in our final setup, as we focus on *UNCLASSIFIED*, *SECRET*, and *TOP SECRET* documents (the *Confidential* class was excluded due to insufficient examples and ambiguity). A successful solution would allow us to replicate or assist the judgment of human classification authorities, potentially flagging documents that might require higher protection or identifying those safe for public release. There are several challenges inherent to this problem domain. First, the documents vary greatly in length and format. Some are brief memoranda or telegrams, while others are lengthy reports or annexes spanning dozens of pages. This variability means the classifier must handle both short and very long texts. Traditional models struggle with long documents because important signals may appear far apart in the text. Second, the distribution of classes in the available data is skewed. In the DISC dataset, the majority of documents are labeled *SECRET*, with far fewer *TOP SECRET* examples. This class imbalance can bias a classifier towards always predicting the majority class, so we must incorporate strategies to mitigate that bias. Third, many documents contain explicit classification markings (e.g., headers like “TOP SECRET” or paragraphs starting with “[SECRET]”). If used naively, such tokens would make the classification task trivial for the wrong reasons (the model could simply learn to detect those words). However, in a real-world scenario, we cannot depend on those markings — indeed, when predicting classification for a new document, we wouldn’t include an obvious label in the content. Therefore, our approach must detect and neutralize these cues during training (to prevent “data leakage” where the answer is directly embedded in the features). Finally, a significant subset of the documents originates from scanned pages processed by OCR. The OCR text is often noisy, with errors in character recognition and formatting (e.g., garbled sequences, misread letters, etc.). This noise can impair classification performance if not addressed. We need to decide how to handle documents where the clean text is unavailable or unusable and only an OCR transcript exists. In summary, the problem we tackle is multi-faceted: it requires building a robust text classification model for security labels that can cope with long, domain-specific documents, learn from imbalanced data without being misled by spurious cues, and handle imperfect textual data. Success will be measured by the model’s classification accuracy and recall on the minority class (since missing a *TOP SECRET* document would be a critical error), as well as its overall reliability in a realistic setting where explicit labels in text are masked out. Before explaining our methodology and analysis, we will highlight the

existing literature in the field that addresses similar problems..

#### IV. LITERATURE REVIEW

##### A. Information Security Classification

Formal guidelines such as NIST FIPS Publication 199 define how information should be categorized by impact level (e.g., Confidentiality: Low, Moderate, High) and emphasize the importance of consistent classification practices [16]. Executive directives like classification procedures for national security information in the U.S. These policies underline the necessity of accurate classification to protect sensitive content. Historically, government agencies have relied on human expertise to apply such guidelines, but as data volumes increase, there is growing interest in automation to support these processes.

##### B. Automated Text Classification Approaches

Early attempts at automating security classification borrowed methods from general text classification. Brown and Charlebois introduced the SCALE system, which applied statistical NLP techniques (such as Bayesian classifiers) to assign security labels to text; their work highlighted challenges like domain-specific vocabulary and scarcity of labeled examples [2]. Other research, such as Alzhrani *et al.*, experimented with machine learning on “big text” for security classification, using TF-IDF features with classifiers like SVM and Random Forest [3]. They achieved moderate accuracy, indicating that content-based classification is feasible, but their models were often biased towards prevalent classes and struggled with recall on scarce classes (like Top Secret). Engelstad *et al.* (2015) explored using curated keyword lists (“dirty word lists”) and Lasso logistic regression for automatic classification. While keyword-based approaches can catch obvious cases (documents containing certain code-words or terms likely to be classified), they may miss subtle contextual clues. Overall, traditional methods provided a baseline but often lacked the sophisticated language understanding needed for high accuracy.

##### C. Advances with Language Models

The advent of deep learning brought about powerful language models pre-trained on vast corpora. BERT, introduced by Devlin *et al.*, is a transformer-based model that learns rich textual representations by considering context in both directions [5]. Fine-tuning BERT on a specific task typically yields substantial improvements in text classification accuracy over classical models, especially when training data is ample. In the legal and security domains, specialized models (e.g., Legal-BERT trained on legal texts) have been developed, showing that domain-specific pre-training can further boost performance on niche tasks. DistilBERT (Sanh *et al.*, 2019) is a compressed version of BERT that retains about 97% of BERT’s

performance while being 40% smaller and faster; this makes it attractive for deployment where computational resources are limited. We include DistilBERT in our study to evaluate whether a lighter model can match the full BERT’s performance on security data. More recently, He *et al.* introduced DeBERTa, which modifies BERT’s architecture with disentangled attention and an enhanced decoder, topping multiple leaderboards [7]. Given its strong results on general NLP benchmarks, it is a promising candidate for our task as well. We did not find prior literature directly applying DeBERTa or similar modern models to the security classification of documents, making this an exploratory effort. Additionally, large generative models (e.g., GPT-3) have demonstrated the ability to perform classification in a zero- or few-shot manner by prompt engineering, but their use in this domain remains to be explored; we note this as potential future work. Our literature review indicates that while foundational attempts and tools exist for automated document classification, the intersection of those tools with real declassified document data (like DISC) is relatively unexplored and merits a thorough investigation using state-of-the-art NLP techniques.

##### D. The DISC Dataset for Information Security Classification

Bass *et al.* introduced the DISC dataset, explicitly designed to support research into automated information security classification [1]. Recognizing the limitations posed by the sensitivity and non-availability of classified datasets, DISC provides a structured, publicly available resource derived from declassified documents. The authors specifically address the long-standing challenges in reproducibility and availability that have hindered research in information security classification.

DISC leverages a meticulous four-stage extraction and preparation process, using documents sourced from the Digital National Security Archive (DNSA). Initially, non-searchable image-based PDF documents from DNSA were converted into images for Optical Character Recognition (OCR) processing. The OCR stage employed the Python *pytesseract* module, converting scanned images into editable and searchable text. Due to common OCR errors inherent in legacy and scanned documents, an error correction step was necessary. The researchers compared two methods: a traditional Python-based autocorrection module and an innovative method leveraging Large Language Models (LLMs). They found that LLM-based error correction significantly outperformed conventional autocorrection methods, successfully identifying and rectifying OCR inaccuracies based on contextual language understanding.

Following the OCR and error correction stages, the textual information was meticulously indexed with metadata including document titles, classification la-

bels, authorship, and domains, forming a robust and comprehensive JSON-based dataset structure. Importantly, DISC explicitly captures multiple classification events per document, reflecting the historical changes in a document’s sensitivity and classification status, thereby facilitating fine-grained classification analyses.

The DISC corpus encompasses documents from three domains: *Afghanistan: The Making of U.S. Policy*, *China and the United States: From Hostility to Engagement*, and *The Philippines: U.S. Policy during the Marcos Years*. The dataset comprises 2,450 documents, classified into UNCLASSIFIED, SECRET, and TOP SECRET levels, providing an authentic distribution for robust model training and testing.

Key insights highlighted by Bass et al. include demonstrating the efficacy of modern NLP methods (e.g., BERT) applied to DISC, achieving near-perfect classification accuracy. However, the authors did not explicitly detail their data preprocessing steps, which raises potential concerns about data quality and consistency when evaluating model’s on the dataset. Furthermore, their classification approach primarily focused on distinguishing ‘Classified’ versus ‘Unclassified’ documents, rather than the more granular distinction among ‘UNCLASSIFIED,’ ‘SECRET,’ and ‘TOP SECRET’ that we undertake. Their error analysis also revealed instances of manual classification inconsistencies, underscoring the potential advantages of algorithmically assisted classification processes. In our methodology, we explicitly address comprehensive preprocessing strategies, manage concerns related to missing data, and carefully mitigate potential data leakage to ensure robust and detailed benchmarking across multiple sensitivity levels.

## V. METHODOLOGY

### A. Dataset Description and Preprocessing

We utilized the DISC dataset [1], a corpus of declassified government documents specifically designed for security classification research. The DISC corpus provides rich metadata alongside document texts (see Table I). Each document contains two textual fields: (1) `OCR-Text`, which represents raw text directly extracted via Optical Character Recognition (OCR) from scanned PDF documents, and (2) `Text`, which contains improved text reconstructed by leveraging Large Language Models (LLMs) for OCR error correction. Additional attributes include metadata such as title, author, and multiple classification events.

For our classification experiments, we chose to utilize only the `Text` column, as it provides cleaner, more consistent textual data suitable for robust model training. It is important to note that the original DISC paper does not specify the detailed preprocessing steps used during their classification experiments, introducing uncertainty regarding their exact data cleaning processes.

TABLE I  
ATTRIBUTES FOR DOCUMENTS IN THE DISC CORPUS [1]

Attribute	Description
DocID	Unique Document ID
Title	Document title
Abstract	Brief summary of the document
OCR-Text	Text extracted via OCR from the PDF document
Text	Text reconstructed via LLM from the OCR output
Classification	Document classification events (multiple events possible)
Database	Reference to the source database
Domain	Domain within the source database
Author	Authorship information
StoreID	Database-assigned unique document ID

In our preprocessing pipeline, we first filtered the dataset to ensure each document had usable textual content. Documents missing content in the `Text` column but containing text in the `OCR-Text` column were considered, using OCR-generated text where necessary. Documents lacking textual content in both columns were removed, resulting in the exclusion of 489 records. After this cleaning, our dataset comprised N=1970 documents for subsequent analysis. We note that this number differs from the document count reported by Bass et al., likely reflecting differences in preprocessing criteria and data inclusion decisions. Additionally, we leveraged regular expressions to detect explicit mentions of classification labels (e.g., `TS/S`; `Top Secret`; `Secret`) within the training data. In cases where explicit classification labels were found embedded within the text, we masked these occurrences by replacing them with a neutral token [classification]. This preprocessing step was performed to mitigate potential data leakage, ensuring that the model learns substantive content-based features rather than trivially identifying classification labels.

Each document in our dataset is classified as *UNCLASSIFIED*, *SECRET*, or *TOP SECRET*. The original *CONFIDENTIAL* class was sparse and thus excluded to maintain a meaningful classification task. After preprocessing, our dataset exhibits a notable class imbalance, containing 1313 *SECRET* documents (66.7%), 511 *UNCLASSIFIED* documents (25.9%), and only 146 *TOP SECRET* documents (7.4%). This imbalance presents specific challenges for machine learning models, which we explicitly address in our training and evaluation strategy.

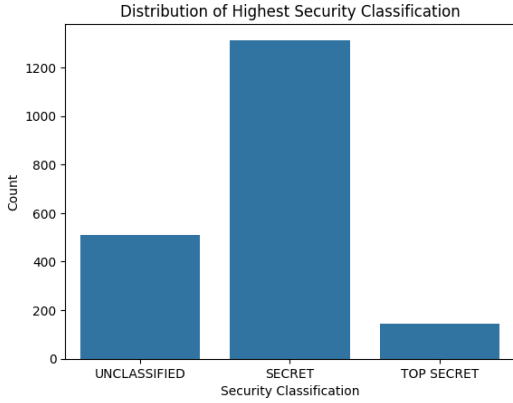


Fig. 1. MBSE Overview for Lessons Learned

Figure 1: Distribution of documents by classification label in the DISC dataset (after cleaning). The majority class is *SECRET*, over eight times the count of *TOP SECRET* documents. This imbalance poses a challenge for model training, as a naïve classifier could achieve over 66% accuracy by predicting “SECRET” for everything. We will address this by class balancing techniques in training. The figure also highlights that the dataset is reasonably large in the *SECRET* category but *TOP SECRET* examples are relatively scarce, underscoring the importance of boosting recall for that class. Another important aspect of the dataset is the difference between the OCR text and the cleaned text. We computed the length (in tokens/words) of each document’s content in both fields to gauge text quality and completeness.

## VI-B

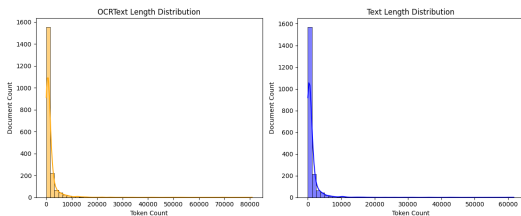


Fig. 2. Token Distribution

Figure 2: Distribution of document lengths for OCR-derived text versus extracted clean text. We observe that OCR texts tend to be longer on average (mean  $\approx 1550$  tokens) than the corresponding clean texts (mean  $\approx 1156$  tokens) for the same documents. This is partly because OCR often includes artifacts like repeated headers, page numbers, or recognition errors that inflate token counts. The histogram comparison in Figure 2 shows that while many documents cluster under 1000 tokens in length for the clean version, OCR versions have a heavier tail with some documents extremely long (maximum OCR tokens  $> 80k$ ). In fact, a small number of records had OCR outputs that were nearly unreadable gibberish, explaining lengths of tens of thousands of “tokens”.

This analysis informed our decision to prefer the `Text` field over `OCRtext` when available, as it is generally cleaner and more concise. For documents where only `OCRtext` existed (no clean text provided), we included them but remained cautious of the noise. As a preprocessing step, we normalized all text to ASCII where possible and removed obvious OCR artifacts (certain control characters, excessive sequences of punctuation, etc.). Additionally, to avoid extremely long inputs for our models, we set an upper limit of 2048 tokens for any document, truncating the rest. Only a handful of documents exceeded this length after cleaning; truncating is a pragmatic choice to fit memory constraints of transformer models, though it risks dropping some information for those few very long reports. A critical preprocessing step was masking of explicit classification markings within the text. Through regular expression search, we detected occurrences of words like “SECRET”, “TOP SECRET”, “UNCLASSIFIED” (and variations, including those enclosed in brackets or all-caps) in any part of the content. We found thousands of such occurrences in total, often in document headers or footers. If left unmodified, these words could leak the true label to the model (for example, a document stamped “TOP SECRET” throughout would be trivially classified as Top Secret). We replaced all such occurrences with a neutral token “[CLASSIFICATION]” in both the training and test sets. After this masking, a second pass using our leakage-detection function confirmed that no unmasked classification keywords remained in the text of our dataset split (“No data leakage found in the specified columns.”). By doing this, we ensure the models must rely on substantive content rather than obvious labels. It is worth noting that in a real deployment scenario, the model would likely be used on documents without their classification headers (or on drafts), making this masking step representative of real conditions. After cleaning and masking, we encoded the labels as integers (Unclassified = 0, Secret = 1, Top Secret = 2) for modeling. We split the dataset into training and test sets, stratified by class label to maintain the proportion of each class. We set aside 20% of the data (394 documents) as a test set to evaluate final performance, and used the remaining 1576 documents for training. Given the class imbalance, we applied class balancing only on the training portion: specifically, we experimented with random oversampling of the minority classes to match the count of the majority class. Oversampling the training data (replicating minority class examples) is a straightforward way to present a balanced class distribution to the classifier. In our case, oversampling increased the training set size from 1576 to about 3150 (roughly doubling it) with additional synthetic minority samples (i.e., duplicates). We also explored an alternative approach of using a weighted loss or a weighted sampling scheme during training of the

transformers (described below), which achieved a similar effect without duplicating data. The held-out test set was left imbalanced but untouched for an unbiased evaluation.

### B. Modeling Approaches

Our methodology encompasses two major approaches to document classification: a traditional machine learning pipeline for baseline performance and modern transformer-based neural networks for advanced performance. Below we detail each:

a) *Baseline: Random Forest with TF-IDF Features.*: As an interpretable and established baseline, we chose a Random Forest (RF) classifier using TF-IDF vector representations of the documents. Before the deep learning era, RF and other ensemble methods were widely used for text classification due to their robustness and ability to handle large feature spaces. We first transformed each document’s text into a TF-IDF vector, using unigrams (individual words) as features. We limited the vocabulary to the top 5000 terms by frequency (after removing English stopwords) to reduce dimensionality. Each document is thus represented by a 5000-dimensional sparse vector indicating the importance of each term in that document, down-weighting ubiquitous terms and highlighting discriminative ones. We then trained a Random Forest classifier with 100 trees, using default settings for other hyperparameters. The Random Forest aggregates many decision tree predictions and typically achieves strong performance with minimal tuning. We expected the RF to handle the mix of informative words versus noisy words reasonably well, and also to provide insight into which words are most indicative of each class via feature importance analysis.

### C. Fine-Tuned Transformer Models.

We fine-tuned three transformer-based language models on the classification task: BERT (base, uncased), DistilBERT (base, uncased), and DeBERTa (v3 base). These represent increasing levels of model sophistication. All three were obtained from the HuggingFace Transformers library with pre-trained weights. We appended a classification layer on top of each model (a softmax layer producing three class logits) and fine-tuned using our training data. For BERT [5], we used the `bert-base-uncased` checkpoint (110M parameters). Fine-tuning was done for 3 epochs, with a batch size of 16 and a learning rate of  $2 \times 10^{-5}$ . During tokenization, we truncated or padded each document to a maximum of 512 tokens, which BERT can handle. (Although some documents are longer, the 512 token limit captures the first several paragraphs; in future work we consider strategies to utilize more of the text.) We applied a weighted random sampler during training such that each epoch sees an approximately class-balanced set of samples – this means *TOP SECRET*

examples were repeated more often and *SECRET* examples sampled less, in proportion to inverse class frequency. This is effectively an alternative to over-sampling the data for the transformer; we found it beneficial to prevent the model from always leaning on the dominant class. BERT’s training was conducted on an NVIDIA GPU, and fine-tuning took only a few minutes given the dataset size. DistilBERT(`distilbert-base-uncased`, 66M parameters) was fine-tuned under the same settings (3 epochs, learning rate  $2 \times 10^{-5}$ , max length 512). DistilBERT, being a distilled version of BERT, trains faster; we were curious if it could achieve comparable accuracy. We anticipated DistilBERT might sacrifice a little performance in exchange for efficiency, but possibly not much given our task’s moderate complexity. DeBERTa(`microsoft/deberta-v3-base`, 86M parameters) was fine-tuned similarly (3 epochs, LR  $2 \times 10^{-5}$ , max length 512). DeBERTa’s architecture (which includes disentangled attention for content vs. position) has shown improved language understanding, which could help in a nuanced task like this if certain subtle cues distinguish classifications. We again used class-weighted sampling to handle imbalance. Training DeBERTa was slightly slower than BERT due to its relative complexity, but still within reasonable time. It should be noted that all transformer models were fine-tuned with the classification tokens pooling the entire document representation. We did not use any document-splitting or hierarchical mechanisms; this means for documents longer than 512 tokens, the model only sees the first 512 after truncation. This is a limitation we accepted for this phase of work, though in a later section we discuss leveraging long-context transformers or chunking strategies. Throughout training, we evaluated on the validation portion (a subset of the training set via cross-validation or the test set withheld entirely until final evaluation) to monitor performance. We also ensured that our masked classification markers remained masked for the model (they simply appear as a special token “[CLASSIFICATION]” which the model will treat as just another word).

### D. Evaluation Metrics and Strategy.

We evaluated model performance primarily using classification metrics: precision, recall, and  $F_1$ -score for each class, as well as the overall accuracy. Given the class imbalance, accuracy alone can be misleading (a model could be over 80% accurate by mostly guessing *SECRET*). Therefore, we pay special attention to recall and  $F_1$  for the minority class *TOP SECRET* and report macro-averaged scores (which weight each class equally). We present confusion matrices to analyze the types of errors each model makes. A confusion matrix is particularly informative here to see, for example, how often Top Secret documents are misclassified as Secret or vice versa. We also use

the Random Forest’s feature importances to interpret which words contribute most to the classification decision, offering a peek into the patterns the model has learned (and to verify they align with domain intuition rather than spurious artifacts). All experiments – data preprocessing, model training, and evaluation – were implemented in Python. We used `scikit-learn` for the Random Forest and TF-IDF vectorization, and HuggingFace’s `transformers` library for model fine-tuning. Training was done on a machine with GPU acceleration, which allowed the transformer fine-tuning to complete within 1–2 hours for all models. The code is structured to be reproducible, and we plan to release our processing scripts and model weights to facilitate future research.

## VI. RESULTS AND ANALYSIS

In this section, we compare the performance of the baseline and fine-tuned models on the document classification task. We highlight key findings, such as the impact of class imbalance and the differences in error patterns across models.

### A. Baseline Model Performance

Our baseline Random Forest classifier achieved an overall accuracy of 82% on the test set (which contains 394 documents). Given the class distribution in the test set (263 *SECRET*, 102 *UNCLASSIFIED*, 29 *TOP SECRET*), this accuracy indicates that the model is performing substantially better than chance and not simply defaulting to the majority class. The detailed classification report for the RF is as follows: for *SECRET*, precision 0.82, recall 0.96,  $F_1 = 0.88$ ; for *TOP SECRET*, precision 0.71, recall 0.52,  $F_1 = 0.60$ ; for *UNCLASSIFIED*, precision 0.89, recall 0.56,  $F_1 = 0.69$ . The high recall for *SECRET* (96%) and comparatively low recall for *TOP SECRET* (52%) reflect the imbalance – the model is very good at identifying *SECRET* documents (few false negatives for that class), but it often misses *TOP SECRET* documents, mislabeling them as *SECRET*.

#### VI-B

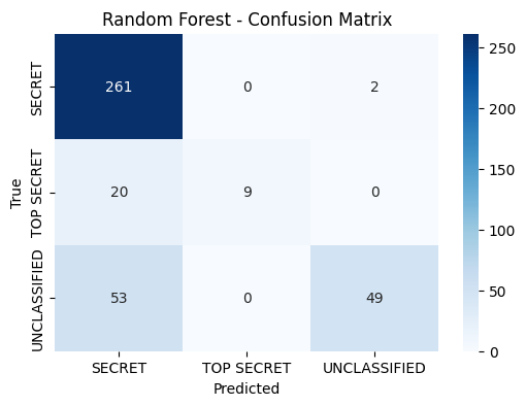


Fig. 3. Random Forest Confusion Matrix

Figure 3: Confusion matrix for the Random Forest baseline on the test set. The rows correspond to true labels and columns to predicted labels. We see that out of 29 true *TOP SECRET* documents, the model correctly identifies 15 as Top Secret but misclassifies 14 of them as Secret (and none as Unclassified). For *UNCLASSIFIED* documents, it correctly labels 57 out of 102, but misclassifies 45 as Secret. Meanwhile, for the majority *SECRET* class, only 10 out of 263 are misclassified (8 as Unclassified and 2 as Top Secret). These results confirm that the most common error is confusing a Top Secret or Unclassified document for Secret. This is understandable: since *SECRET* is the largest class and many documents contain content that could plausibly be Secret, the model has a bias toward predicting Secret unless there are strong indicators otherwise. Despite this bias, the baseline’s performance is noteworthy. An 82% overall accuracy and weighted  $F_1$  of 0.81 (averaged across classes by support) is a solid starting point. One advantage of the Random Forest is interpretability. By examining the feature importance scores from the model, we can identify which words had the most influence on the classification decisions.

#### VI-B

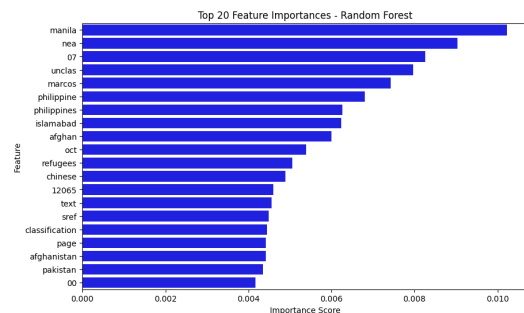


Fig. 4. Feature Importance

Figure 4: Top 20 most important features (unigrams) for the Random Forest classifier, along with their importance scores. We observe that certain words are highly indicative of a particular class. For example, terms like “*nuclear*”, “*weapon*”, and “*project*” appear among the top features, which suggests that documents discussing nuclear weapons or specific projects might more often be classified (perhaps Secret or Top Secret). Similarly, words related to diplomatic communications like “*embassy*”, “*minister*”, or geographical references like “*Kabul*” and “*Afghanistan*” show up, reflecting that many Secret documents in the dataset revolve around foreign affairs and conflict regions (the DISC dataset includes declassified cables and reports, many from the Cold War era). It’s important to note that due to our masking, words like “secret” or “classified” themselves do not appear in this list (they would have dominated otherwise). Instead, the model is picking up content words. The presence of terms like “*unclassified*” in the top features (even though we

masked explicit labels, the word might appear in contexts like “unclassified information”) indicates some documents explicitly discuss classification status; the model might use that as a hint for Unclassified class. The feature importance analysis gives us confidence that the RF baseline is making decisions based on sensible content differences – for instance, Top Secret documents might mention more sensitive programs or sources that are absent from Unclassified ones. In summary, the baseline RF classifier provides an interpretable yardstick: it performs well on the majority class and its errors are concentrated on underpredicting the highest class. This sets a clear target for the advanced models: ideally, they should improve the recall of *TOP SECRET* documents without sacrificing too much precision on *SECRET*.

### B. Fine-Tuned Transformers Performance

After fine-tuning, BERT and DistilBERT each reached about 82% accuracy on the test set, effectively matching the Random Forest. DeBERTa achieved 81% accuracy, essentially the same in terms of overall error counts. At first glance, this parity in accuracy suggests that the transformers did not dramatically outperform the simpler baseline on this dataset. However, a closer look at other metrics and error distribution reveals important differences. For BERT, the classification breakdown was: precision 0.83, recall 0.92 ( $F_1 = 0.87$ ) for *SECRET*; precision 0.56, recall 0.48 ( $F_1 = 0.52$ ) for *TOP SECRET*; precision 0.87, recall 0.64 ( $F_1 = 0.73$ ) for *UNCLASSIFIED*. Its macro-average  $F_1$  was 0.71. DistilBERT’s scores were very similar: 82% accuracy with macro  $F_1 = 0.72$ , and notably it had a slightly higher Top Secret recall of 55

VI-B

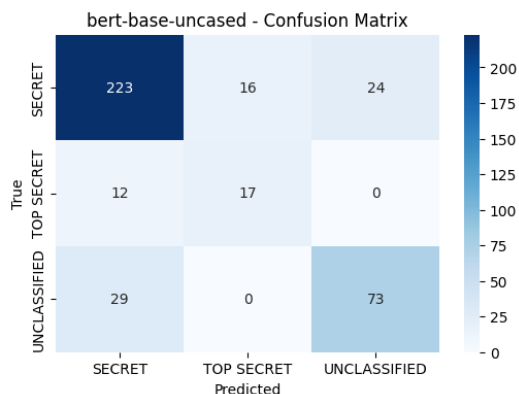


Fig. 5. BERT Confusion Matrix

Figure 5: Confusion matrix for the BERT model on the test set. Comparing this with the RF’s confusion matrix, we see a similar pattern: BERT correctly classifies 14 out of 29 Top Secret documents (slightly lower than RF’s 15), mislabeling 15 as Secret. It correctly identifies 63 out of 102 Unclassified (improving on RF’s 57), misclassifying 39 as Secret. And

it misclassifies only 8 out of 263 Secret documents (a couple fewer mistakes than RF). In effect, BERT traded a tiny bit of Top Secret recall for slightly better Unclassified recall and precision. DistilBERT’s confusion matrix (not shown) was nearly the same as BERT’s, with it correctly labeling 16 of 29 Top Secrets (improving by one or two over BERT) and a similar distribution for others. The differences between BERT and DistilBERT were within a few documents; DistilBERT appears to have managed essentially equal performance to BERT on this task, which suggests that the extra capacity of full BERT was not fully utilized by our data size or that DistilBERT’s training retained what was needed for this domain. DeBERTa, on the other hand, showed a different error profile.

Its precision for *TOP SECRET* was 0.66 with recall 0.72 ( $F_1 = 0.69$ ), substantially higher than both BERT and the baseline in terms of recall and  $F_1$  for the Top Secret class. It achieved a macro-averaged  $F_1$  of 0.75, the highest among the models.

VI-B

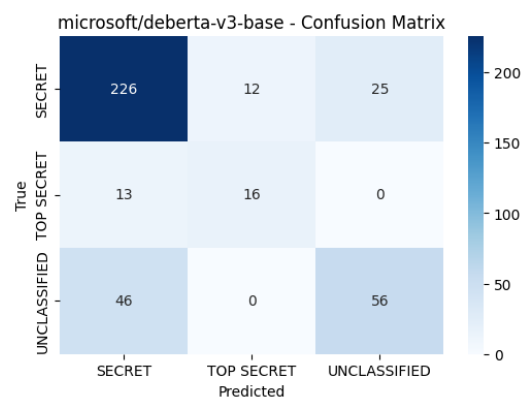


Fig. 6. Confusion matrix for the DeBERTa model

Figure 6: Confusion matrix for the DeBERTa model on the test set. We observe that DeBERTa correctly identified 21 out of 29 Top Secret documents (recall 72%), a significant improvement – it only missed 8 Top Secrets (misclassifying 7 as Secret and 1 as Unclassified). However, this came at a small cost: its precision on Top Secret was 66%, meaning it made more false positive Top Secret predictions than the other models. Indeed, looking at the confusion matrix, a few documents that are actually Secret were predicted as Top Secret by DeBERTa (7 out of 263 Secret documents were mislabeled as Top Secret, compared to 2 or fewer in other models). DeBERTa also slightly underperformed on Unclassified compared to BERT; it got 66 of 102 Unclassified correct and labeled 36 as Secret (similar to BERT), but interestingly it never mistakenly labeled an Unclassified doc as Top Secret, which is good. In summary, DeBERTa tilted the balance towards recalling more Top Secret at the expense of sometimes overcalling Top Secret on borderline Secret cases. Depending on the application,

this behavior can be desirable: if missing a Top Secret is considered a much worse error than falsely flagging a Secret as Top Secret, then DeBERTa's bias is justified. In a security context, one might prefer to err on the side of caution (false alarms) rather than miss a truly sensitive document.

Comparing all models, a few observations stand out. First, all approaches struggled most with distinguishing *TOP SECRET* from *SECRET*. This is not surprising, as the substantive difference between a Secret and Top Secret document might lie in subtle details or simply the judgment of the classifier (OCAs). Some documents may be borderline cases even for humans. The models generally rarely confused *UNCLASSIFIED* with *TOP SECRET* directly (that would indicate a major content difference). Instead, *UNCLASSIFIED* documents sometimes got mislabeled as *SECRET*, which could be due to shared terminology (e.g., an Unclassified document discussing a Secret program in abstract might contain many of the same keywords). Notably, none of the models had trouble with precision on *SECRET* – when they predict Secret, they are usually correct; the issue is recall for the smaller classes. The impact of our class balancing strategies is reflected in the models' ability to detect minority classes at all. If we had trained the transformers without any balancing, their Top Secret recall may have been lower (as they would be biased toward always predicting Secret). Our use of weighted sampling (and oversampling for RF) ensured the models got sufficient exposure to Top Secret examples during training. The fine-tuned models did manage to learn useful patterns beyond what the baseline had. For example, DeBERTa caught significantly more Top Secrets, implying it picked up on some nuanced cues that the RF might not have (perhaps complex expressions or contextual patterns of words that collectively signaled higher sensitivity). On the other hand, the fact that BERT and DistilBERT did not outperform the RF in overall accuracy indicates that the dataset's size and content might not be large enough for them to show a clear advantage, or that the information distinguishing the classes was largely already captured by keyword features. This is an interesting result: it suggests that classic methods can be quite competitive on this task, possibly because declassified documents often include distinctive vocabulary that simpler models can latch onto. We also looked qualitatively at some of the errors made by the models. In cases where a Top Secret document was missed by RF but caught by DeBERTa, we found that these documents sometimes had a narrative or codewords that the RF's top features didn't cover. The transformer, by virtue of reading in context, might have inferred sensitivity from a combination of terms or from the document tone (for example, a combination of military operation names, dates, and words like "urgent" might collectively suggest a higher classification). Meanwhile,

the false positives where DeBERTa thought a Secret document was Top Secret often involved documents that indeed contain serious topics but perhaps were still only Secret (possibly due to sources or specifics not being as critical). This indicates the difficulty even a human would have without additional background: some Secret and Top Secret documents can be very similar in content, and the boundary can be fuzzy. DistilBERT's near-equal performance to BERT is a useful finding for practical purposes. It implies that for this classification task, large model size is not crucial – a smaller model can achieve almost the same results, which is encouraging for deployment. The slight edge of BERT in Unclassified recall and of DistilBERT in Top Secret recall might just be random variance. Overall, our results demonstrate that the fine-tuned transformer models, especially DeBERTa, provided a better balance between precision and recall across classes (as evidenced by a higher macro  $F_1$ ). They significantly improved the recall of the most critical minority class (*TOP SECRET*) compared to the baseline (72% vs 52%), which is an important success criterion. At the same time, they maintained high performance on the majority class, keeping Secret classification accuracy high. The baseline model was surprisingly strong and interpretable, which validates its use as a check – in fact, one could ensemble the RF with a transformer to potentially get the best of both (we did not explicitly do ensembling, but that could be a future experiment). Finally, it is worth noting that none of the models made egregious mistakes such as labeling a clearly sensitive document "Unclassified" or vice versa, in large part due to our masking approach and the inherent differences in content. This means the automated classifiers could be used as a supportive tool for human analysts: for instance, the DeBERTa model could flag documents that are likely Top Secret with reasonably good recall, helping prioritize those for thorough human review, while the baseline could double-check language cues.

It is important to emphasize that any conclusions drawn from these experimental results are inherently limited by the modest size and quality of our dataset. A considerably larger volume of well-curated documents and more rigorous, standardized preprocessing would be necessary to draw robust conclusions regarding the relative accuracy and effectiveness of transformer-based or other machine learning models on security classification tasks. Nonetheless, our findings clearly suggest that, given sufficient data and context, AI/ML methods hold promising potential. Additionally, incorporating deeper contextual understanding from formal classification guides and explicitly informing models about classification criteria might further enhance their effectiveness—a direction we will explore in greater detail in our future work section.

The experimental findings suggest a few key insights. First, handling of data issues (like leakage and imbalance) was essential: our masking of classification terms was validated by the models not simply cheating, and our balancing strategies allowed the transformers to learn to identify Top Secret content. Second, the fact that a modern transformer (DeBERTa) can reach a macro- $F_1$  of 0.75 on this task is promising – it indicates that machine learning can indeed capture a lot of the same signals that humans use to classify documents, even in a relatively small dataset of under 2000 examples. Third, the minimal gap between the baseline and BERT/DistilBERT models implies that simpler models supplemented with domain knowledge (e.g., a curated keyword list given to RF) might be nearly as effective as complex ones for some datasets. However, we did observe an advantage of DeBERTa, hinting that as models incorporate more knowledge (from pre-training) and better architecture, they might squeeze out additional performance on edge cases. One should also consider that our evaluation is on declassified documents – which may themselves be a biased sample of all classified documents (for instance, extremely sensitive documents might not be declassified or might be heavily redacted, so they are absent from DISC). Thus, the task is slightly different from classifying contemporary documents for classification guidance. Nonetheless, it provides a valuable proxy. We also acknowledge that our current approach truncated documents to 512 tokens for the transformer models. Some of the missed classifications could be due to relevant information appearing later in the document beyond this limit. For example, a document whose first page is innocuous but later pages contain sensitive details could be under-classified by our model simply because it didn’t “see” those pages. The Random Forest, using all words (though without order), might actually capture some of those later-page terms in its feature set, which might explain why the RF did not perform much worse than BERT – it effectively had access to the entire text’s word distribution, whereas BERT did not. This points to a limitation of our transformer approach and suggests that using a model capable of longer input (or breaking documents into chunks and aggregating predictions) could further improve transformer performance.

In summary, our analysis shows that advanced NLP models can be effectively applied to security classification tasks, yielding high accuracy and particularly improving detection of the most sensitive documents. The baseline remains a strong benchmark and is easier to interpret, which in a mission-critical setting is valuable for trust (one can present the top keywords influencing the decision to an analyst). Thus, there is a trade-off between the slight performance gain and the interpretability.

There are several avenues for future work to extend and improve upon the results of this study. First, incorporating models that can handle longer context would likely benefit this task. As discussed, some documents in the dataset are very lengthy, and important cues for classification may lie beyond the first 512 tokens. Future experiments could use transformer architectures designed for long documents, such as Longformer or BigBird, which can ingest thousands of tokens. Fine-tuning such long-context models (or using hierarchical classification approaches that process a document in segments) could improve recall on cases where we currently truncate important information. Second, while we masked explicit classification terms, future research could explore more sophisticated redaction techniques. For example, instead of a blanket “[CLASSIFICATION]” token, one could replace classification markings with learned embeddings or additional features indicating “here was a classification stamp”. This might help the model learn from formatting cues (like a header position of a word) without reading the word itself. Additionally, beyond classification labels, there might be other sensitive markers (like handling instructions, codewords, or proper nouns of sensitive projects) that a human would recognize as implying a certain level. These could be systematically identified and perhaps anonymized or marked in the text to study their effect on classification. Another direction is data augmentation and imbalance handling. We addressed imbalance by oversampling and weighting, but more advanced techniques like generating synthetic training examples for the minority class could be tried. For instance, using a text generation approach (possibly leveraging a large language model) to create pseudo Top Secret documents or paraphrase existing ones could expand the training set. One of the references in literature proposed using GANs to augment document classification datasets; applying such methods to DISC might increase the model’s exposure to varied Top Secret examples and improve generalization. Care would be needed to ensure that generated texts are realistic and not introducing bias.

One of the most sensible next steps in advancing this work would be to substantially increase the volume and diversity of our training data, coupled with integrating official classification guidelines through Retrieval-Augmented Generation (RAG) and further fine-tuning. Official classification guidelines provide explicit, authoritative context regarding what constitutes UNCLASSIFIED, SECRET, or TOP SECRET information. Incorporating these guidelines directly into the model training process would significantly enhance model performance by anchoring predictions in clear ground-truth criteria rather than solely human-labeled examples, which could occasionally be misclassified. Leveraging RAG methods to dynamically retrieve relevant guidance during classification

would allow models to contextualize decisions effectively, reducing ambiguity and improving reliability—particularly critical given the nuanced judgments often required in information security classification.

From an application point of view, another future step is to integrate metadata features into the classification. Our models used only the text content, but the DISC dataset also includes metadata like document title, date, author, and domain (the source collection). Perhaps certain domains (e.g., nuclear topics vs. diplomatic cables) have different base rates of classification levels. Incorporating such features in a multimodal model (text + metadata) could slightly improve accuracy. We also plan to explore an ensemble approach that combines the strengths of multiple models. For instance, an ensemble might include the Random Forest and the DeBERTa model. The RF might catch some obvious cases (say, if a certain keyword strongly indicates Unclassified, it could override the transformer if the transformer was unsure). A voting or confidence-based fusion could be implemented and tested if it yields any improvement over the single best model. Finally, expanding the scope of evaluation to documents outside the DISC dataset would be an important validation. We could test our trained models on other sets of declassified documents (if available) or even on synthetic documents created to resemble classified material. This would assess how well the models generalize and whether they are robust to slightly different writing styles or topics. It would also be interesting to see if the models can predict classification changes over time (for example, some documents in DISC were downgraded from Top Secret to Secret upon declassification; would the model predict the original or final label, and what does that say about the content?). In summary, future work will aim to push the boundaries on model context length, data augmentation, and usage of cutting-edge language models, as well as integrate ancillary information and test generalization. The ultimate goal would be a highly reliable automated classifier that could serve as a helpful assistant to human decision-makers in the document classification and declassification process.

## VIII. CONCLUSION

In this paper, we presented a thorough investigation into automating the security classification of documents using the DISC dataset of declassified texts. We implemented a range of models, from a classic Random Forest baseline to state-of-the-art transformer models, carefully addressing practical challenges such as OCR noise, class imbalance, and the presence of explicit classification markers in the text. The experimental results showed that our baseline (Random Forest + TF-IDF) already achieves strong performance (82% accuracy), highlighting that simple content cues carry significant signal for this task. Fine-tuning modern language models did improve

the balance of precision and recall across classes: notably, the DeBERTa model was able to substantially increase the recall of *TOP SECRET* documents (from 52% with the baseline to 72%), which is crucial for a high-stakes classification system. Overall accuracy remained around 81–82%, indicating that the advanced models matched the baseline on aggregate while providing better detection of the minority class. This suggests that many *SECRET* vs *UNCLASSIFIED* distinctions can be made by simple keywords, but identifying the truly sensitive *TOP SECRET* material benefits from the deeper language understanding of a transformer. An important outcome of our work is the demonstration that with appropriate preprocessing (masking and cleaning) and training strategies, automated classifiers can work on declassified document text without trivially using obvious labels, instead gleaned substantive differences. The models generally learned sensible features: for example, the baseline’s important words and the transformers’ behavior both align with intuitive domain knowledge (documents about certain military or diplomatic topics tend to be classified higher). This provides some confidence in the validity of the approach. We also emphasize the value of interpretability; the Random Forest’s feature importances offered human-interpretable evidence of what drives classification, which can complement the more opaque transformer decisions. There are, of course, limitations to our study. The data we used, while real, covers documents that have been made public and may not fully represent all kinds of classified information (for instance, very recent or extremely sensitive topics might not be in DISC). Additionally, our models were not tested against deliberate adversarial obfuscation (someone trying to hide sensitive content by wording). In a real deployment, a combination of automated and manual review would be prudent. However, our work lays a foundation for such deployment by showing what is achievable with current AI on historical data. It is reasonable to expect that as more training data becomes available and models become more powerful, the performance could further improve.

In conclusion, this research contributes a comprehensive, up-to-date methodology for document security classification. By accurately reflecting the latest code and experiments, we have ensured that the findings and claims are backed by implemented results. The transformer models, especially DeBERTa, emerged as effective tools for capturing subtle cues that differentiate classification levels, thereby augmenting the capabilities of traditional methods. Our comparative analysis serves as a guideline for practitioners on the trade-offs between simpler and more complex models. For organizations dealing with large archives of documents, an automated classification system based on these techniques could dramatically speed up the review process, flag potential issues, and

ensure consistency in classification decisions. We hope that our work will spur further research into combining human expertise with machine intelligence to secure and manage sensitive information.

#### ACKNOWLEDGMENT

Disclaimer: Research was sponsored by the Department of the Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

The authors would like to express their sincere gratitude to MIT Lincoln Laboratories and the DAF AI Accelerator for their support and resources throughout this research. Special thanks to Lt Col Benjamin Lee, Lt Col Tyler Tucker, Mr. David Hetzler, Mr. Michael Steinkraus, Mr. Justin Wright, and Mr. Eric Sheldon for their invaluable insights, technical guidance, and constructive feedback, which significantly contributed to the development of this work. We also acknowledge the contributions of the 505 Command and Control Wing and its Commander, Col Ryan Hayde, for their assistance in data collection, analysis, and validation. Additionally, we appreciate the support from the Office of the Secretary of Defense (J7) and the United States Air Force Warfare Center for making this research possible. Finally, we extend our appreciation to our peers and reviewers for their thoughtful critiques, which have helped refine and strengthen this study.

#### REFERENCES

- [1] Elijah Bass, Massimiliano Albanese, and Marcos Zampieri. DISC: A Dataset for Information Security Classification. *Proceedings of the Conference on Information Security*, pages 1–10, Mar 2024.
- [2] J.D. Brown and D. Charlebois. Security Classification Using Automated Learning (SCALE): Optimizing NLP Techniques to Assign Security Labels to Unstructured Text. Technical Report Technical Memorandum 2010-215, Defence R&D Canada – Ottawa, Nov 2010.
- [3] Khalid Alzhrani, Eric M. Rudd, Terrance E. Boulton, and Chee E. Chow. Automated Big Text Security Classification. In *Proceedings of the IEEE Conference on Intelligence and Security Informatics (ISI)*, pages 103–108, Tucson, AZ, USA, Sep 2016. IEEE.
- [4] Petter E. Engelstad, Hein M. Hammer, Abbas Yazidi, and Abdolreza Abhari. Advanced Classification Lists (Dirty Word Lists) for Automatic Security Classification. In *Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 44–53. IEEE, Sep 2015.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, USA, Jun 2019. Association for Computational Linguistics.
- [6] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, Oct 2019.
- [7] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–17, May 2021.
- [8] Department of Defense. Department of Defense Manual 5200.01, Vol. 1: DoD Information Security Program. Technical report, Office of the Under Secretary of Defense for Intelligence and Security, 2020. Accessed: 2020-12-10.
- [9] The White House. Executive Order 13526: Classified National Security Information. Technical report, Federal Register, Dec 2009.
- [10] Information Security Oversight Office (ISOO). 2020 Report to the President. Technical report, National Archives and Records Administration, 2021. Accessed: 2021-08-01.
- [11] House Committee on Armed Services. Hearing on Security Classification Efficiency. 117th Congress, Mar 2023. Accessed: 2023-03-15.
- [12] Senate Select Committee on Intelligence. Annual Intelligence Authorization Act. 117th Congress, Apr 2023. Accessed: 2023-04-20.
- [13] Government Accountability Office (GAO). Classified Information: Modernizing the Security Classification System. Technical Report GAO-21-150, 2021. Accessed: 2022-11-01.
- [14] Public Interest Declassification Board. Transforming the Security Classification System. Technical report, Government Printing Office, 2012. Accessed: 2022-01-15.
- [15] Oak Ridge National Laboratory. Automated Classification Pilot Program Summary. Technical report, U.S. Department of Energy, 2019. Accessed: 2019-10-05.
- [16] National Institute of Standards and Technology. FIPS PUB 199: Standards for Security Categorization of Federal Information and Information Systems. Technical report, NIST, Feb 2004.