HAW HAMBURG

BACHELORARBEIT

BERT-based Spam Email Classification: Exploring Model Effectiveness Across Different Datasets

vorgelegt am 16. 09. 2025

Anna Van

Erstprüferin: Prof. Dr. Larissa Putzar
Zweitprüferin: Prof. Dr. Sabine Schumann

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Department Medientechnik
Finkenau 35
20081 Hamburg

Abstract

Machine learning and transformers have become increasingly involved in modern applications, particularly for text generation and classification. This work presents the current state of art text classifiers with transformers, with a focus on email spam detection. It will evaluate and compare the performance across a selection of BERT-Transformer models: The base model BERT and specialized models RoBERTa and DistilBERT. Each of them is trained on datasets Ling-Spam, SpamAssassin and Enron-Spam as well as on a combined dataset, to examine how dataset and model architecture influence the classification performance. The study highlights the differences in efficiency, generalization and stability among the models, showing how lighter models like DistilBERT excel on smaller datasets while BERT benefits from larger, more complex data. It further underscores how model choice should consider dataset size, complexity and deployment priorities. These results offer guidance for selecting and implementing spam filters and emphasize the trade-offs between cost and performance priorities.

Abstrakt

Maschinelles Lernen und Transformer haben in modernen Anwendungen zunehmen an Bedeutung gewonnen, insbesondere im Bereich der Textgenerierung und -klassifikation. Diese Arbeite befasst sich mit dem aktuellen Stand der Technik von Textklassifikatoren, mit einem Fokus auf Transformern und der Erkennung von E-Mail-Spam. Dabei werden eine Auswahl von BERT-basierten Transformermodelle untersucht und verglichen: das Basismodell BERT und die spezialisierten Varianten RoBERTa und DistilBERT. Jedes dieser Modelle wird auf den Datensätzen Ling-Spam, SpamAssassin und Enron-Spam sowie einem kombinierten Datensatz trainiert, um zu analysieren, wie Datensatz und Modellarchitektur die Leistung bei der Klassifizierung beeinflussen. Die Studie hebt die Unterschiede in Effizienz, Generalisierungsfähigkeit und Stabilität der Modelle hervor und zeigt, das leichtere Modelle wie DistilBERT auf kleineren Datensätzen bessere Leistungen erbringen, während BERT bei größeren und komplexeren Daten überzeugt. Dadurch wird unterstrichen, dass die Modellwahl die Größe und Komplexität des Datensatzes sowie die Anforderungen an den Einsatz berücksichtigt werden muss. Die Ergebnisse liefern Orientierung bei der Auswahl und Implementierung von Spamfiltern und verdeutlichen die Abwägung zwischen Kosten- und Leistungsprioritäten.

Table of Contents

List of Figures

List of Abbreviations

LLM	Large Language Model
GPT	Generative Pretrained Transformer
BERT	Bidirectional Encoder Representations from Transformers
BART	Bidirectional and Autoregressive Techniques
NLP	Natural Language Processing
LLC	Limited Liability Company

1 Introduction

1.1 Motivation

Nowadays, people all over the world have an email address as a point of personal connection to the internet. It serves as a tool for communication, information exchange, and registration for digital services. In many ways, an email address functions as the digital equivalent of an address to your physical mailbox. Anyone who knows it can send messages directly to the recipient.

This accessibility is a point of weakness that can be exploited by malicious actors. Unwanted advertisements, fraudulent messages, phishing or malware – spam emails represent a serious cybersecurity risk with the potential to cause damages to individuals and organizations. Attempts to access confidential information through deceptive messages have been made since the early days of the internet. The end user has consistently been the target of cyberattacks which is considered the weakest link in the chain of security in the online space. Thus, development of effective spam detection systems has been a decade old endeavor.

As of recent, Artificial Intelligence (AI) has begun affecting the digital landscape arising with both opportunities as well as new risks. With the rapid development of this technology, the efforts to regulate them are running behind. A defining moment that raised public awareness of the growth of this sector was the release of ChatGPT in November 2022. LLMs became part of people's everyday with ChatGPT reporting 100 million monthly active users shortly after their launch proving the societal impact of this technology. In the context of cybersecurity, this must become a new point of examination. What dangers spawn along with the opportunities from this technology is something that needs to be explored. The same tool that enables more natural human-computer interaction can be misused to generate increasingly sophisticated malicious content.

As traditional spam detection may fall behind in the climate of where attacks also grow more sophisticated, it is necessary to adapt with growing urgency and develop countermeasures against spam and phishing. This bachelor thesis aims to explore how modern language models, specifically transformer-based architectures, can be leveraged to strengthen spam and phishing detection. By evaluating and comparing the performance of selected models across multiple datasets, this work contributes to the ongoing effort to secure the most widely used, and most exploited communication channel of the digital age.

1.2 Structure

The foundation provides a theoretical background to understand the implementation and analysis. The thesis begins with outlining text classification and introduces transformers models. Next, are the evaluation metrics to define how a model's performance is assessed. After that, the risk of spam through phishing attacks and large language models are addressed. Followed by the key principles of a dataset and considerations to data quality.

Then the implementation will be detailed, which describes the practical setup of the study. Starting off with what tools were used, this includes environment, and important libraries. Then, the used datasets are presented: Ling-Spam, SpamAssassin and the combined dataset. After that the chosen model's BERT and RoBERTa and DistilBERT are explained. And lastly the methods applied during training.

The results report the findings of each model and their performance across multiple datasets. The evaluation then serves to compare the performances and discusses what factors influenced the results. Finally, the conclusion which summarizes the findings and draws practical and theoretical implications, outlines limitations and potential directions for future research.

2 Foundations

2.1 Text Classification

Text Classification is one of the most explored fields in NLP, used in a wide range of applications such as sentiment analysis and question answering. The idea behind it is to predict the meaning of the words in the context they have been given. Early approaches include using algorithms such as logistic regression and Naïve Bayes. The common type of text classification process can be broken down into four phases: Feature extraction, dimension reductions, classifier selection and evaluations (**Figure 2.1**).

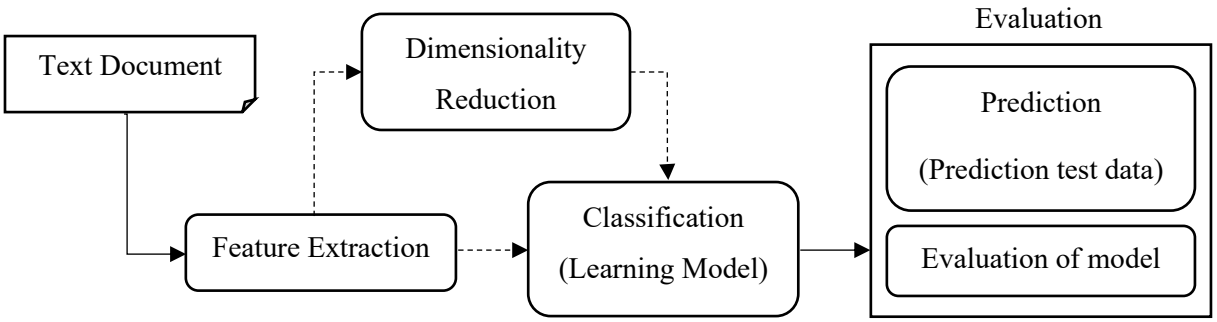

Figure 2.1: Text Classification Pipeline (based on Kamran Kowsari, 2019)

As text classification is used on texts and documents, the first step is to convert this unstructured data into structured data. This for one means removing unnecessary data. Afterwards, feature extraction methods will be used such as Term Frequency-Inverse Document Frequency (TF-IDF), Term Frequency (TF), Word2Vec, and Global Vectors for Word Representation (GloVe). The application of these methods results in structured, vectorized data that classifier models can further deal with.

Dimensionality Reduction is a step that described the process of reducing the number of dimensions while preserving the important information. This is an optional but recommended step to deal with the high processing times that occur when dealing with complex and large amounts of data. It results in fewer data to process that is focused on contextually higher valued themes. Common techniques include Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and non-negative matrix factorization (NMF) (Kamran Kowsari, 2020).

As mentioned before early primitive classification techniques include algorithms such as Naïve Bayes. Their performance has been largely overshadowed by today's use of deep learning models. Deep learning models are generally more advantageous, being able to deal with large datasets and complex relationships within the data.

2.2 Transformer Architecture

Transformers are a type of deep learning architecture that has been introduced in Vaswani et al., 2023. They are used for text classification tasks. Their strengths lie especially in their contextual awareness and being able to deal with longer and complex sentences. Additionally, they have been extensively pretrained and can be further trained for specific tasks.

Today they have found widespread use with the likes of GPT and BERT. Their advantage in allowing for easy adaptation by fine-tuning has made them a popular tool over traditional text classification algorithms. Especially in the field of NLP it has established itself as a trustworthy tool. The usage of transformers is time saving and has proven to achieve better results for the resulting models. (Lewis Tunstall, 2022)

There are different configurations of transformers with the main three being: Encoder-only (e.g. BERT), decoder-only (e.g. GPT) or encoder-decoder (e.g. BART). The functionality behind transformers architecture is the added attention mechanism which is the key-point that enables their increased contextual understanding. The Encoder-decoder architecture was designed with the intention of translation, meaning understanding text as well as generating text. Transformers with a focus in classifying text without generating text do not use a decoder. Therefore, it should be noted that the complete encoder-decoder architecture (**Figure 2.2**) is not prevalent in all transformers. (Ashish Vaswani, 2023)

Another significant functionality in transformers is the ability for computations to run parallelly. This differs from recurrent or convolutional neural networks that run sequentially. This prevents bottlenecks, speeds up the training process and allows scalability to larger datasets. Additionally, longer sequences of information will be less likely to suffer from information loss. This is achieved through self-attention computation. First an input is given with a variable-length sequence and is passed to the transformer. There it will be processed by an encoder. The encoder splits the input into evenly sized vectors and passes it one by one to the decoder. The decoder takes each bit and processes them in the order it's been passed on. For each piece of information that is processed by the encoder there will be a hidden state that gets passed on to the decoder right away. The hidden state contains information about the weight of the passed-on vectors. In this case weight describes the importance in the context of all information given. This allows for deeper contextual understanding. (Lewis Tunstall, 2022)

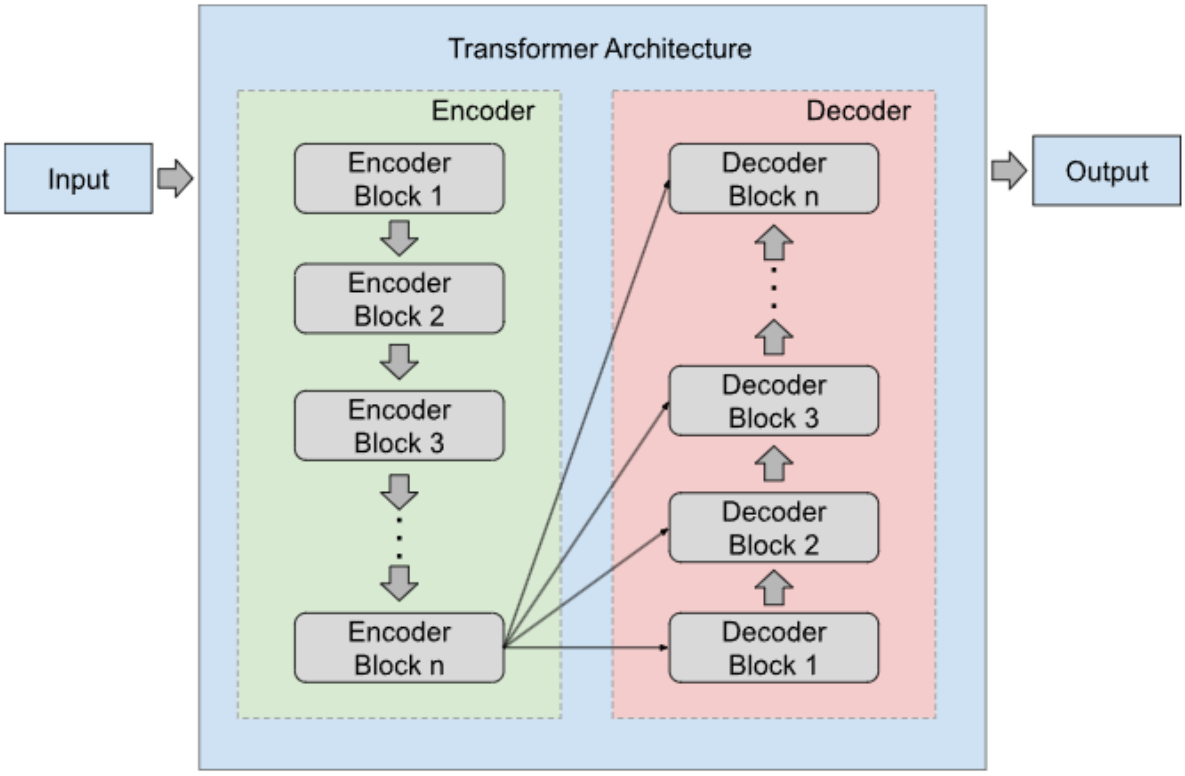

Figure 2.2: Transformer's Encoder-Decoder Architecture

2.3 Evaluation Metrics

One of the most common ways to evaluate the performance of a classification models is to analyze the predictions made by the model. Predictions are what the learning model outputs when classifying the input data. Based on their correctness those will be considered "true positive" for a correctly classified positive label, "true negative" for a correctly classified negative classes, "false positive" for a negative label that has incorrectly been classified as positive and "false negative" for a mistakenly predicted positive label. Widely used metrics when dealing with the evaluation are accuracy, precision, recall, training and validation loss with additional metrics such as F1-score offering further insight into the models' performance.

The **accuracy** is calculated from the number of correct predictions over the total of all predictions. One of the straightforward metrics that are generally intuitive and helpful – however, it is not as reliable when dealing with imbalanced datasets.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Predicitions}$$

The number of correctly predicted positives over all positives is described as **precision**. High precision indicates that if the model predicts the positive class, it is generally correct. This may matter when false positives are hugely detrimental to the use case of the model.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Next is **recall** which looks very similar to precision. It described the true positives over all positives and false negatives. It is therefore more telling of false negatives and useful if these are the ones prioritized to be detected.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

F1-score (or F_1-score) is a metric that balances recall and precision. It is useful if there should be a single measure of performance instead of looking at precision and recall individually. It more often finds use when there's class imbalance. F_1-score is otherwise known as $F_ß$-score while setting ß=1 resulting in equal weights of recall and precision. (Kamran Kowsari, 2020)

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Training Loss is a value that measures how well the model is learning the training data. A constant decrease in training loss is to be expected, and the model's training needs to be investigated should there be great deviation. The **validation loss** is a helpful metric that measures the model's prediction errors when working with new data. It works by letting the model predict labels of a different set of data than the initial training data – it is commonly described as test of validation data. Ideally the validation loss should continuously decrease along with the training loss. However, if that is not the case and there is an increase in validation loss then it shows how the model is overfitting.

2.4 Phishing Attacks and LLMs

One of the weakest links in Cybersecurity is the "Human Factor". The strategies to target those vulnerabilities are what are referred to as Social Engineering. It's defined by the attacker's impersonation of trusted individuals such as representatives of reputable organizations to manipulate the victims into revealing confidential information or visiting fraudulent websites. (Informationstechnik, n.d.)

After gaining access to one's e-mail account, attackers may use the compromised account to target contacts connected to the account. These days, this method of impersonating and gaining access through

trusted contact lists is prevalent not only per e-mail but through other communication platforms such as Discord.

The early methods of phishing attempts were done by sending a large number of low-quality emails to many people. The e-mails were generally not convincing to most people and resulted in a low number of people becoming victims of serious consequences. While this method is still in use, other strategies emerged with the goal to target select individuals. One being individualized attacks which more often yield a higher return and are less likely to be recognized as a phishing attempt. Utilization of LLMs can improve the previously mentioned high volume attacks and lead to more convincing phishing attacks with higher rates of success. LLMs can be used to generate or change existing mails content to sound more convincing. The sophistication of phishing e-mails will make it increasingly difficult to detect before they cause harm. Along with the technology that is steadily evolving, grows the urgency to create countermeasures to effectively deal with it. (Lewis Tunstall, 2022)

The result of a study investigating the effectiveness of spam recognition with LLM modified emails show that 73,7% of previously spam classified emails were falsely classified as legitimate after being LLM modified. The accessibility and cost-efficiency of LLMs, combined with their ease of use, underscore the growing need to confront the risks they may present. (Malte Josten, 2024)

With just a text message, an image or a voice memo, generative AI can be trained to imitate a person's likeness. With the quality of generative AI improving there is now a growing concern for human perception being unable to distinguish from human or AI generated content. (Sarah Barrington, 2025)

2.5 Datasets

Datasets are a vital part of training a model. It gives the model direction of what to look for in future data. Based on the intent of the creation of the model and its future use-case, one must create or choose an existing suitable dataset. This means ensuring relevancy to the goal - investigating if the data used is representative of data encountered in its use case later. With the large amount of existing data to deal with comes difficulties like data quality assessment, data management, data democratization and data provenance.

The key principles to a good dataset are quantity, quality and balance. A sufficient size of training data is necessary to provide enough representative data for the model to recognize patterns and telling characteristics to differentiate the different classes. This helps the model develop the ability to deal with real-life data and its complexity as well as variability after its training. Additionally, the larger number of examples help prevent overfitting. Overfitting is a problem that occurs when the model becomes too adapted to the training set and cannot perform well on other sets of data. In this case, further noise and specific details that are not generally common are memorized. This may show through a gap of performance when comparing the training and the validation set. A possible cause of overfitting can be due to an excessive number of epochs in training (Caruana et al., 2000).

Quality of a dataset is ensured by its cleanliness, accuracy, consistency, completeness, and relevance. Cleanliness and accuracy are attributed to the correct and cohesive labeling of the data, with corrupt or unfitting data being removed. In this case, relevance can refer to whether the datasets context is befitting of the intended real application and to the freshness of data. The latter point being especially meaningful in departments like cybersecurity and news. The condition of the datasets used during training correlates to the model's performance with poor quality of data leading to unreliable models.

Balance refers to the proportional representation of all data categories within the dataset. The balance of the dataset during training directly impacts the overall performance of the model. Class imbalance or an imbalanced dataset means that one class within the samples is far fewer than the other. In such cases, the model training on this set of data will be more biased towards the majority class, misclassifying the minority class more often. (Sophie Henning, 2023)

2.6 Data Quality

There is an array of factors to look out for when ensuring the quality of data being used for the model. First is correct and cohesive labelling. Mislabeled data is detrimental to the training process and may lead to further errors in its use case. Additionally, of importance is making sure that the data is up to par with representational occurrences in this aspect as well. An example would be, if the quality of images in a classification use case is pixelated and colorless in application, then such quality is to be used for training as well. Further deciding factors are the age of data which may not be overcome with fine-tuning, suggesting a need to train models from the startup with new data to reach best results. Should the requirement for the data or the models intended application change, so will the assessment of data quality. An additional factor is to make sure that repeated data is appropriately dealt with. Depending on the nature of the dataset, duplicates may result in a biased model after training. On the other hand, removing them could also be destructive and lead to inaccurate representation of real-life data. There is not one correct answer that fits every case and adds more stress to gain an understanding of why duplicates are there in the first place. (Zige Wang, 2024)

While evaluating the quality of data aspects to be considered are challenges with toxicity and bias. Both aspects lead to data that may result in contextual bias in classification later. This is especially relevant in language research and necessitates defining criteria and analysis of models. (Jack W. Rae, 2021)

Data management deals with the assessment of data quality during the selection of data and with the behind the scenes of data usage. This is where one of the biggest concerns during the collection of data is handled. Which is the topic of privacy and how to ensure the privacy of the users whose data is being used. Other factors that must be considered are combination and utilization strategies along with the evaluation of the chosen strategies. (Zige Wang, 2024).

3 Implementation

This chapter describes the tools which encompass the environment and the libraries, then the datasets, the models and the methods used for the implementation.

3.1 Tools

Python is a flexible programming language that has found much popularity in the context of data analysis and machine learning. Its simplicity makes it attractive for prototyping and research-oriented projects. The strong community backing has solidified Python in machine learning with many open-source libraries that provide functionality in data processing, model training and evaluation. In this thesis it allows the integration of datasets, model training pipelines and the evaluation metrics.

Kaggle is a platform that enables users to experiment with machine learning and data science by granting access to publicly available datasets, models and notebooks and allows the publishing of one's own creations. Their platform offers free access to their resources while allowing external ones to be imported. Furthermore, it grants access to their GPUs and TPUs for all registered users, making machine learning more accessible for users that do not have high performing devices at home.

The **Transformer** library grants access to different BERT tokenizers and models. This library is maintained by HuggingFace**.** It is an open-source platform that is renowned for its extensive transformer library with many works focused on NLP. They host an ever-increasing library of resources to use for training and fine-tuning models. This tool was used to adapt pre-trained models specifically for spam email classification.

PyTorch is an open-source deep learning framework providing infrastructure to build, train and deploy neural networks. It grants access to tools such as AdamW which is an optimizer that is commonly used with BERT-like models that stabilizes training and prevents weights from growing too large. The synergy with the HuggingFace Transformer library makes PyTorch preferable over other frameworks. It is the underlying framework for this thesis project on top of which the other libraries are built on.

Scikit-learn is an open-source machine learning library that offers tools for predictive data analysis. It is built using other Python libraries as a base like NumPy, ScipPy and Matplotlib. It allows access to sklearn.metrics which was used for computing performance metrics.

NumPy is a Python library that allows scientific computing containing tools that allow for mathematical functions.

Pandas is another Python library that provides data analysis tools. It was used to deal with the datasets specifically to load them in and merge them.

Tqdm is a library that can visualize the training process through a progress bar amongst other options, additionally it can predict the remaining training time.

Matplotlib is a library that offers different ways of visualizing data.

3.2 Dataset Selection

This chapter aims to outline the selection of datasets that were for training the models focused on spam detection. It further details the differences in intent during their creation. The following datasets have been collected during times when data privacy was not considered, and many now available versions are modified and differ from the time of their creation. Besides their background they all are of different contexts (**Figure 3.1**) and sizes (**Figure 3.2**). As the aim is to explore spam detection with junk mail, the selection of datasets encompasses messages with content focused on spam and legitimate emails.

Collecting emails from freely available newsgroups and mailing lists from public archives is exactly what has been done for the creation of the **Ling-Spam** (Ion Androutsopoulos, 2000) dataset which mainly contained legitimate mail with academic background and a bias to linguistic interests such as job postings and research opportunities. The collection contains 2893 total messages with 2412 being legitimate and 481 spam mail leading to a quite unbalanced dataset. The dataset includes the emails message content, the label of ham or spam as well as the subject line. The spam mails in this dataset are collected from the mailbox of one singular user. Because of its small size and imbalanced representation in data, its simplicity makes it a useful benchmark for testing model behavior on smaller datasets.

One of the most famous datasets for spam and ham mail is the **SpamAssassin** (Mason, 2006) dataset which includes ham data collected mainly from public mailing lists. The spam mails were received through spam traps and through public submissions from a variety of users. This dataset contributed to the creation of the open-source spam filter Apache SpamAssassin. It encompasses a total of about 7.000 messages with a roughly even split of ham and spam content and a slight bias towards ham. Within the dataset there is a distinction in the folders between ‘easy’ ham and ‘hard’ ham which pertains to the ease of telling the ham apart from spam. Further it has folders named ‘easy_ham_2’ and ‘spam2’ indicating later added data. This dataset includes more of the email’s information including the email header, subjects, date sent and senders information as well as the emails message itself. There is no separation of all the information, and it has been all saved into the text content. In this paper, a version of SpamAssassin is used that excludes the ‘hard_ham’ folder as well as emails from the other folders that that were not encoded in UTF-8. This reduction is meaningful to the total and leads to a cut of about 2.000 rows of data as well as a larger imbalance. The initial ratio from 56:44 moved to 65:35 with a stronger bias towards ham. The topics in the ham collection of this dataset contain technical topics such as talks about python and bug fixings. This dataset is slightly larger than Ling-Spam but still somewhat imbalanced with spam accounting for 31% of the data. It has more varied email sources and therefore a larger diversity in its data, making it more realistic setting than linguistic focused emails.

The Enron Corpus (or Enron Email Dataset) consists of approximately 600.000 mails from 150 employees. All these e-mails and personal data were published during the legal investigation of Enron Corporation. Currently available versions of the dataset have been modified by removing personal information and attachments to respect the employee's privacy. One of those versions is the **Enron-Spam** (Vangelis Metsis, 2006) dataset. It is a modified version of the original Enron Corpus containing only about 31.000 entries – a reduction to only 1/20 of the initial dataset - with a close to 50:50 split of both spam and ham mail. Ham mails were taken from six employees while spam messages were collected from four different sources, including the SpamAssassin corpus. They were mainly gathered through spam traps and one singular user's inbox of spam. The reliability from this data comes from the fact that the ham mail is sourced from real inboxes with spam mail being added and labeled as such. This eliminates the possibility of incorrect labeling. On the other hand, while close to a real inbox it does not capture the full realism, as the spam mail has been artificially added. The aim during the creation of this smaller version of the Enron Corpus was to simulate a more individualized mailbox to allow the training of a more personalized spam filter. It's a dataset that has been in use for training spam mail for a substantial amount of time dating back to 2006. It comes with the downsides that it is not representative of current data. This dataset is comparably larger to the other datasets, so it will be interesting how well they would adapt to larger amount of data. Additionally, with the almost even distribution of spam and ham, it might reduce the bias that would arise from imbalanced datasets.

Lastly, to further investigate model performance with more diverse conditions, the three datasets were merging into a single collection that contains more than 40.000 entries. While a grand portion of the data consists of Enron-Spam's collection, the inclusion of the other datasets will allow for more diverse samples of ham and spam. The introduction of variety in writing style, content and spam and ham characteristics will make it the most challenging for model generalization. Further references to this dataset will be as **combined** dataset.

Ling-Spam	Consists of 2.400 legitimate and about 500 spam mails, collected via a mailing list about linguistics.
SpamAssassin	Collection with around 6.000 entries. Ham from volunteer data from a variety of users.
Enron-Spam	Contains 32.000 mail with a real-life cooperate background, taken from selected employees**.**
Combined	Created by merging Ling-Spam, SpamAssassin and Enron-Spam into one diverse dataset with more than 40.000 entries

Figure 3.1: Overview of Datasets for Phishing Detection

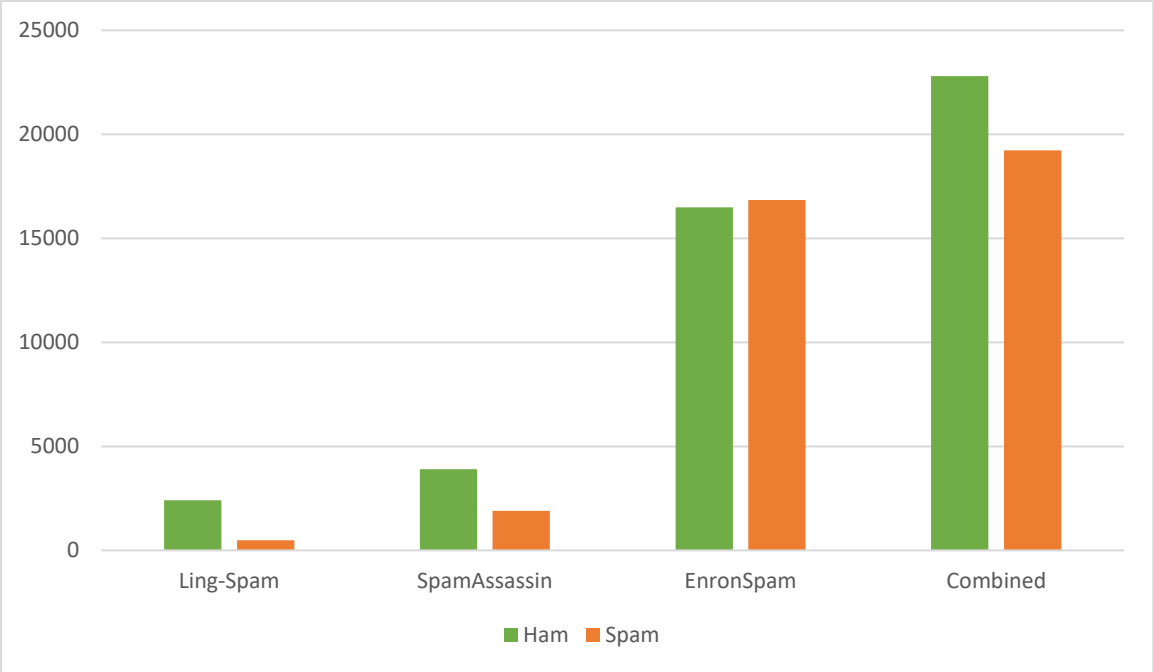

Figure 3.2: Ham and Spam Distribution of the Datasets

3.3 Models

One of the models chosen for this experiment is **BERT** (Jacob Devlin, 2018) - a multilayer bidirectional transformer model specialized in working with unprocessed text. The publishing of this model was a turning point as standard conditional language models at that time were only able to be trained left-to-right or right-to-left. BERT introduced bidirectional pre-training through two unsupervised tasks. The first one is called Masked Language Models (MLM) where a random selection of the input tokens is masked, and the model is tasked to predict them. Next, is the Next Sentence Prediction (NSP) allowing BERT to understand sentences in relation to each other. This training process allowed BERT to outperform existing task-specific architectures in the field of NLP and erased the need for high complexity in fine-tuning.

DistilBERT (Victor Sanh, 2020) is a compact Transformer model developed through the process of knowledge distillation of the original BERT model. The process entails training a smaller student model to learn the behavior of one larger teacher model. The architecture generally remains the same as BERTs. Its conception was a response to the practical limitations of BERT's large size. The runtime is designed to be significantly more efficient with the size of BERT being reduced by 40% and the speed being accelerated by 60%, all while retaining at least 95% of its performance. Like BERT, DistilBERT is primarily trained for NLP focused tasks and supports further fine-tuning for specific applications. Its key advantage lies in its suitability to be used in environments with resource-limitations and real-world scenarios where computational efficiency, and responsiveness are prioritized rather than results with full complexity.

RoBERTa is an acronym meaning Robustly Optimized BERT Pretraining Approach (Yinhan Liu, 2019). It is a version of BERT that was developed after the discovery that BERT is significantly undertrained and is an attempt to optimize the pretraining process. One of the key differences lies in the discard of the NSP process, the reason being that it did not contribute to downstream performance and could hurt the model's performance. Additionally, it employs larger batch sizes and longer training times which requires a larger number of resources. Dynamic masking is used which leads to the reduction of overfitting. The results are an improved performance with a higher accuracy on a wide range of NLP tasks.

3.4 Methods

To explore the effectiveness of the selected models - BERT-base, DistilBERT and RoBERTa - training was completed with the datasets Enron-Spam, SpamAssassin and Ling-Spam as well as the combined dataset. The datasets were hosted on Kaggle and were imported with pandas. Each dataset was standardized into two columns containing message data and a binary label (0 = ham, 1 = spam). Since transformers have their own preprocessing steps for cleaning text, further preprocessing was kept minimal with null value removal to not cause further problems with unclean data. Then the combined dataset was created by merging them together. After the preparation of all the datasets one dataset can be chosen to be used for training. Then, the message and label data are separated, the label data is cast into the necessary datatype, and the text data is tokenized for further usage. While BERT and DistilBERT are compatible with the BERT tokenizer, RoBERTa uses its own model-specific tokenizer which were all available through the transformer library.

The data is split into training and validation subsets, with a standard 75:25 ratio applied across all experiments. Typically, with a larger corpus of data, a smaller split towards validation data would be preferred. However, in this case the amount of data used is rather modest with a total of fewer than 50.000 samples per training session.. The training was set for 10 epochs, and the learning rate to 2e-5 using the AdamW optimizer in combination with a learning rate scheduler.

An initial implementation using a Keras-based pipelines resulted in a function written with the aid of ChatGPT to evaluate the model, its function to calculate the validation loss has been utilized. While the Keras setup yielded usable results for BERT, extending this framework to other BERT-models proved technically limiting. Therefore, the decision was made to switch to the HuggingFace Transformers library and PyTorch, which offers broader support for pretrained transformer architectures. An additional helper function was implemented to compute evaluation metrics - accuracy, precision, recall and F1 score- using the scikit-learn library. Those metrics will be used to assess the model's performance. The training is visualized with a progress bar that provies real-time updates and elapsed time. Each tokenized input along with the corresponding labels are passed to the model to train after which the metrics calculated and displayed at the end of the epoch.

4 Results

The following chapter presents the results obtained from training each model and dataset combination. For readability, the reported values in the text are rounded and the exact metrics are in the chapter **Appendix**.

4.1 BERT

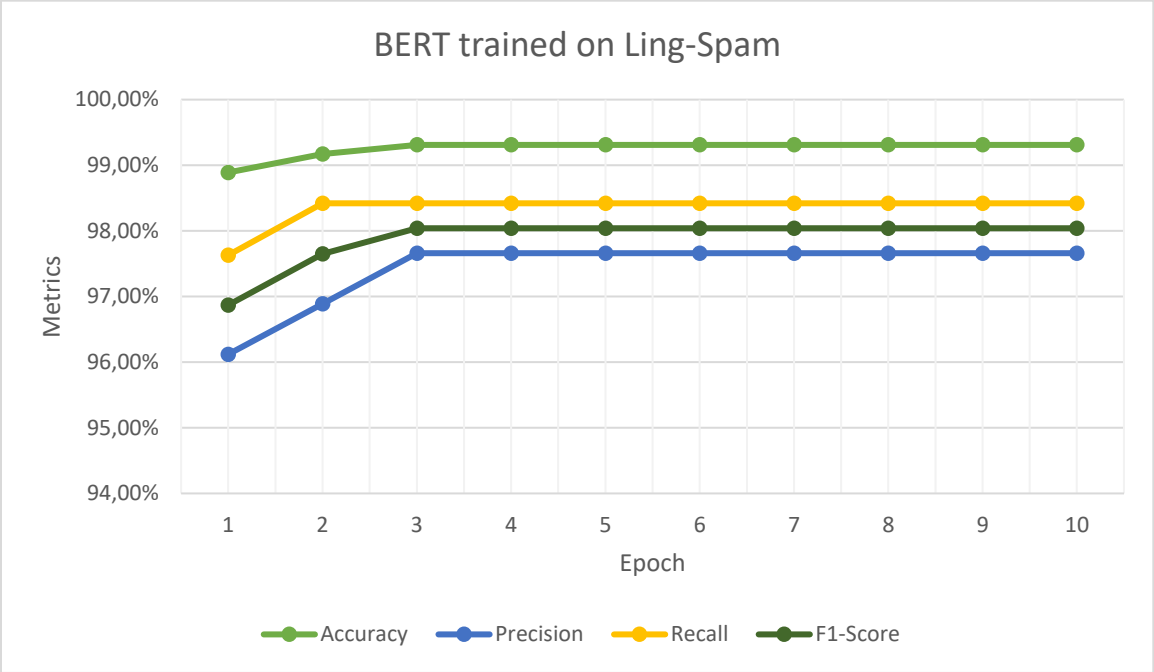

Figure 4.1: BERT's results with Ling-Spam: Accuracy, precision, recall and F1 score across epochs

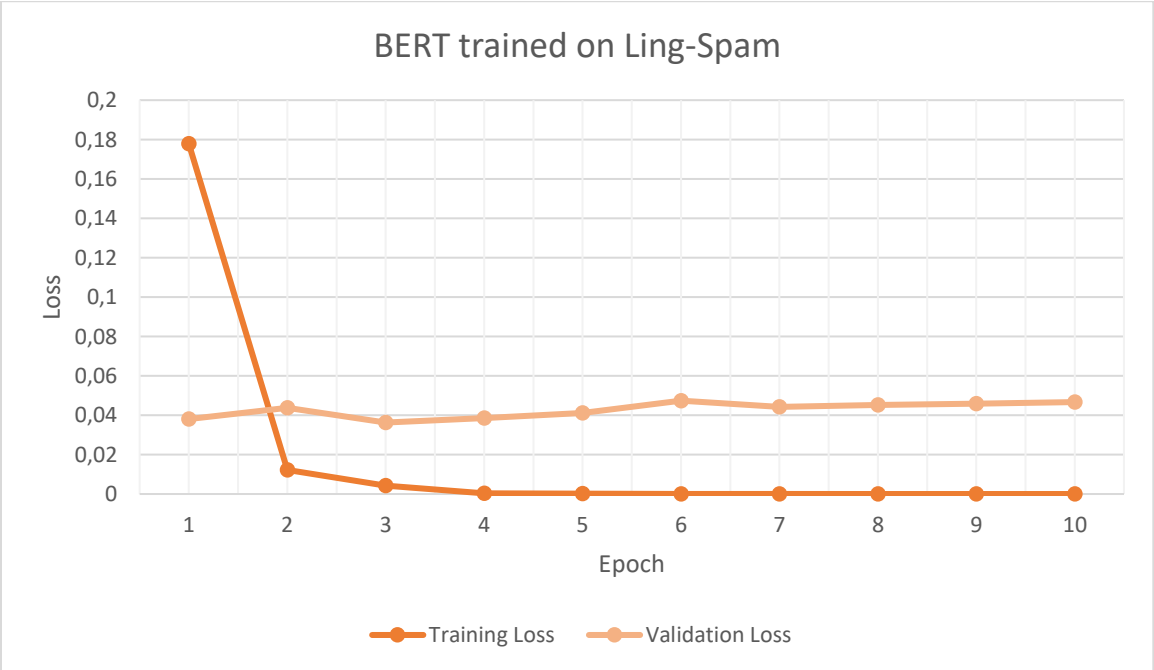

Figure 4.2: BERT's results with Ling-Spam: Training and validation loss across epochs

These are the results of the BERT-model training with various datasets, starting off with Ling-Spam.

The initial results from Ling-Spam show that the model performs strongly from the beginning, with the accuracy starting off at approximately 99% (**Figure 4.1**). The training loss and validation loss as well as the F1 score steadily improved until this peak, which was reached in Epoch 3 (**Figure 4.2**). Beyond this point, training loss continued to decrease and eventually reach zero, while validation loss showed little to no improvement. The metrics plateau after Epoch 3 and show no more change. The training time was a total of 9 minutes and 30 seconds.

The metrics point to the model classifying close to all emails correctly and few errors. Only 2% of spam labelled emails are incorrectly labeled and less than 2% of spam is not identified. The F1 scores prove a balanced performance with no meaningful bias towards either label. The model achieves its peak performance in Epoch 3 and therefore quickly learns to separate spam and ham. Further training after the third epoch seems unnecessary as the results do not show meaningful improvement. The training loss is at almost zero by Epoch 4 which explains why there is no change in the metrics after training further. In this case training time and resources can be reduced if the training is limited to three epochs. The plateau suggests that the model has reached its peak with the given dataset. Further improvement would be possible through expanding the size of the dataset it is training on.

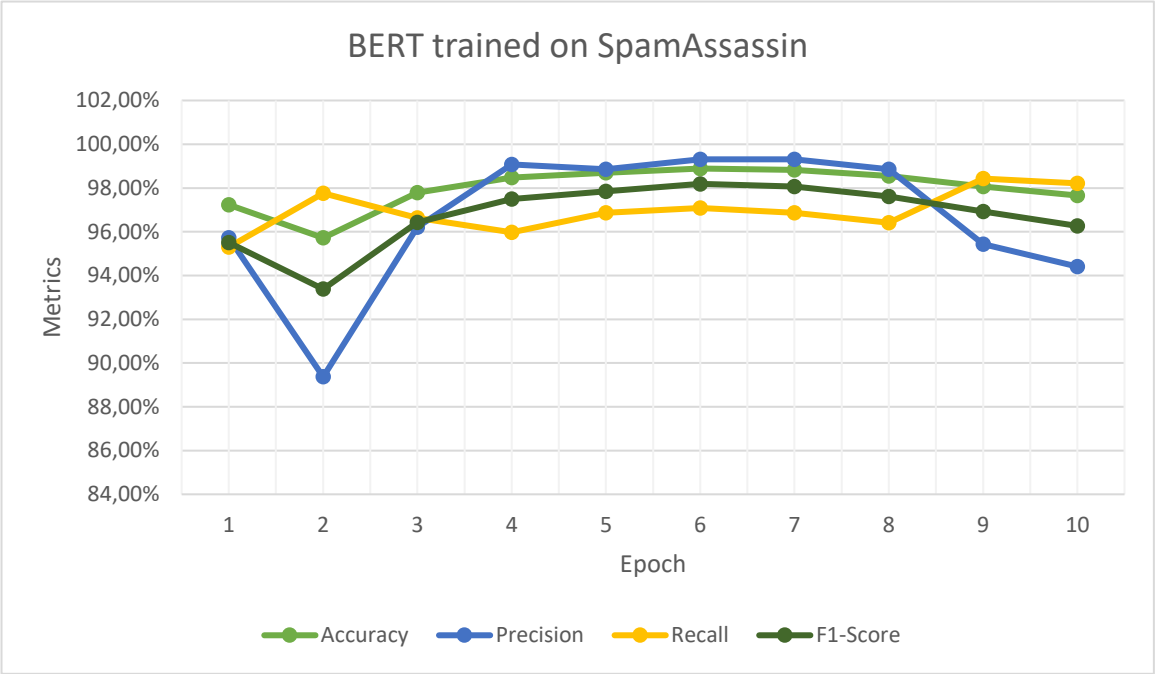

Figure 4.3: BERT's results with SpamAssassin: Accuracy, precision, recall and F1 score across epochs

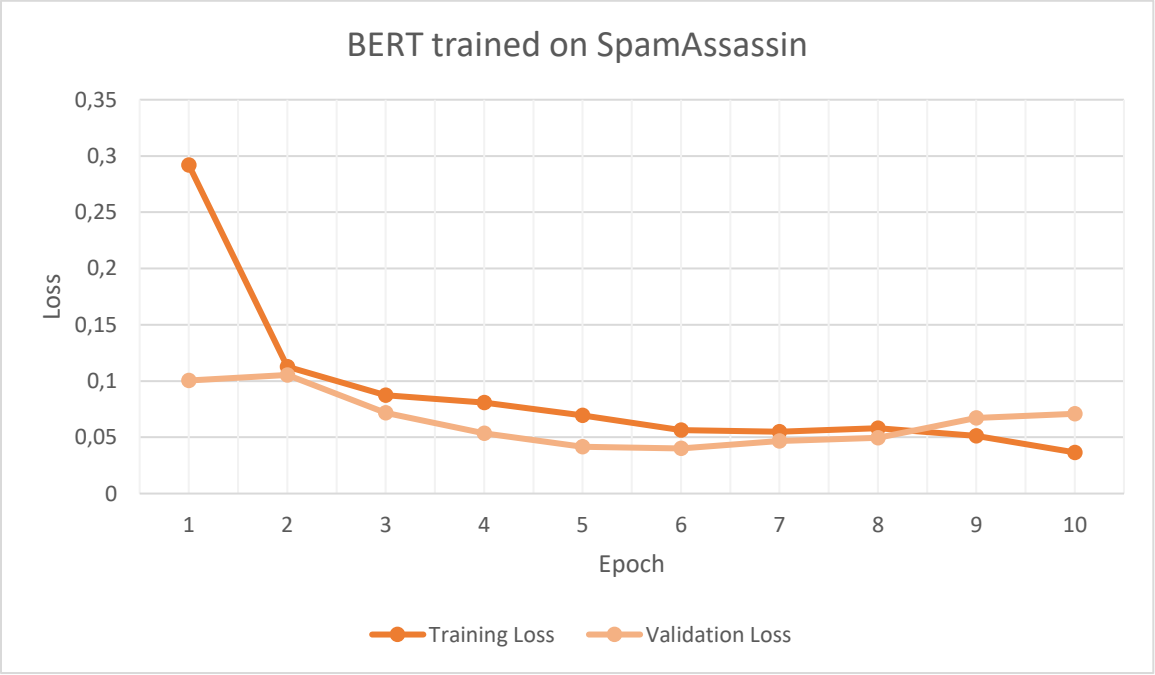

Figure 4.4: BERT's results with SpamAssassin: Training and validation loss across epochs

Next, we have BERT training results with the SpamAssassin dataset. Initial epochs show high accuracy and recall with precision seeing a drastic drop in the second epoch (**Figure 4.3**). Afterwards, from Epoch 3 to 6 the scores see a positive development with accuracy peaking at close to 99% and precision and recall being both high and achieving a good balance shown through the F1 score. In the later training from Epoch 7 on, metrics start being inconsistent. Accuracy, precision and F1-score decrease, and recall goes back up. The training loss steadily declines until the very end reaching close to zero while the validation loss drops initially until Epoch 6 but then rises again after (**Figure 4.4**). The training duration was a total of 19 minutes and 7 seconds.

The early behavior is akin to a paranoid filter - it preferred to classify as spam which is reflected in the high recall and lower precision score. In the middle of the training the model achieved a good balance and consistent results. At this stage, the metrics are stable with minimal change, marking a clear contrast to the early training behavior. Notable is how during the middle epochs precision consistently exceeded the recall. This hints at a slight bias for not flagging spam, making false negatives (missed spam) more likely than false positives (incorrectly labelled ham). The development of the metrics after Epoch 6 shows an uptrend of misclassification for both ham and spam. Telling, is the drop in training loss while there is an increase in validation loss – that is a typical indicator of the model overfitting. Therefore, the best stopping point for the training would be at Epoch 6, beyond which the overfitting negatively impacts the model's performance.

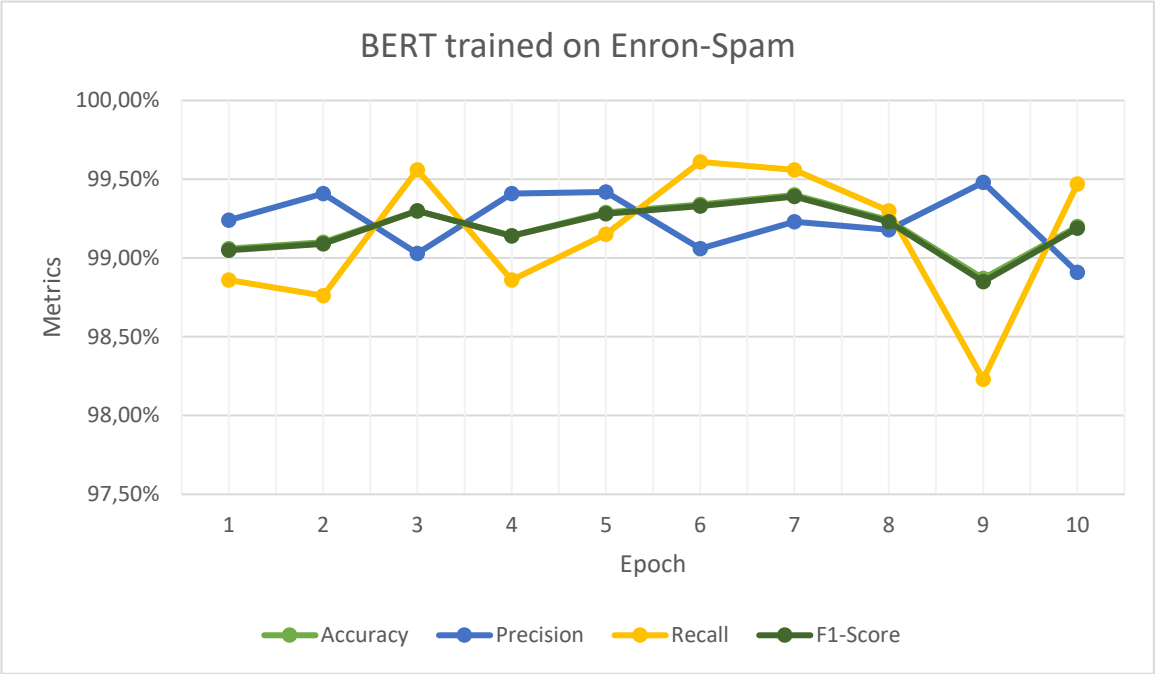

Figure 4.5: BERT's results with Enron-Spam: Accuracy, precision, recall and F1 score across epochs

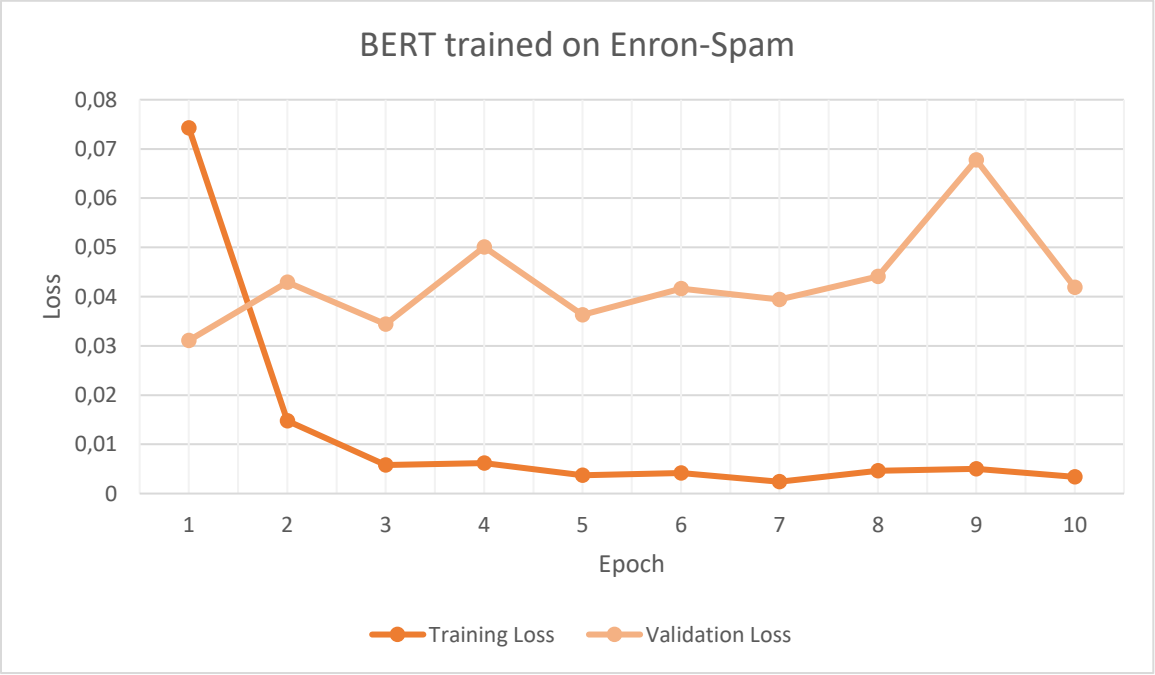

Figure 4.6: BERT's results with Enron-Spam: Training and validation loss across epochs

The training of BERT on the Enron-Spam dataset took 1 hour and 42 minutes. Firstly, notable is that the graph for accuracy and F1 score in **Figure 4.5** are nearly overlapping so that the accuracy curve is barely visible. The model showed a strong start with accuracy reaching above 99% and other metrics similarly high in the first epoch. There is improvement in all metrics besides recall in Epoch 2. After that, metric values fluctuated across epochs. Recall and precision alternated in dominance in Epoch 3, 4, 6, 8 and 10 but the F1 score showed rather stable improvement. The best overall results are in Epoch 7 where the model reaches the highest accuracy and F1 score. Afterwards, there seems to be slight instability as validation loss is going up and accuracy takes a dip (**Figure 4.6**). While precision reaches its peak in Epoch 9, recall on the other hand has reached its lowest score so far. Validation loss has a big spike too but calms down in the next epoch. Overall, the metrics remain high until the end. While the graph makes the metrics to be very inconsistent, the actual numbers are quite close to each other with most of the fluctuations not more meaningful than about 1%.

The results indicate that BERT deals with this dataset very well with an overall low number of false predictions. It reaches high precision and simultaneously high recall, resulting in a model that has almost no spam or ham mail falsely classified. Stability in performance is achieved after at least Epoch 3 and BERT should, for optimal performance, be trained until Epoch 7. Further training leads to signs of inconsistency and chances of overfitting.

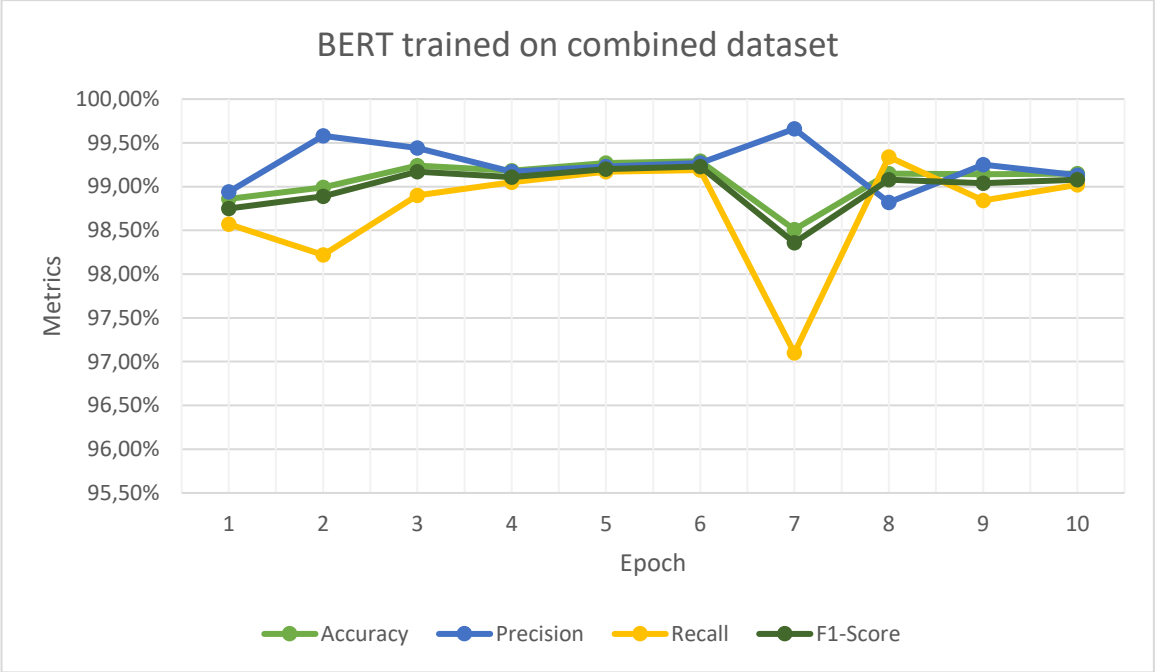

Figure 4.7: BERT's results with combined dataset: Accuracy, precision, recall and F1 score across epochs

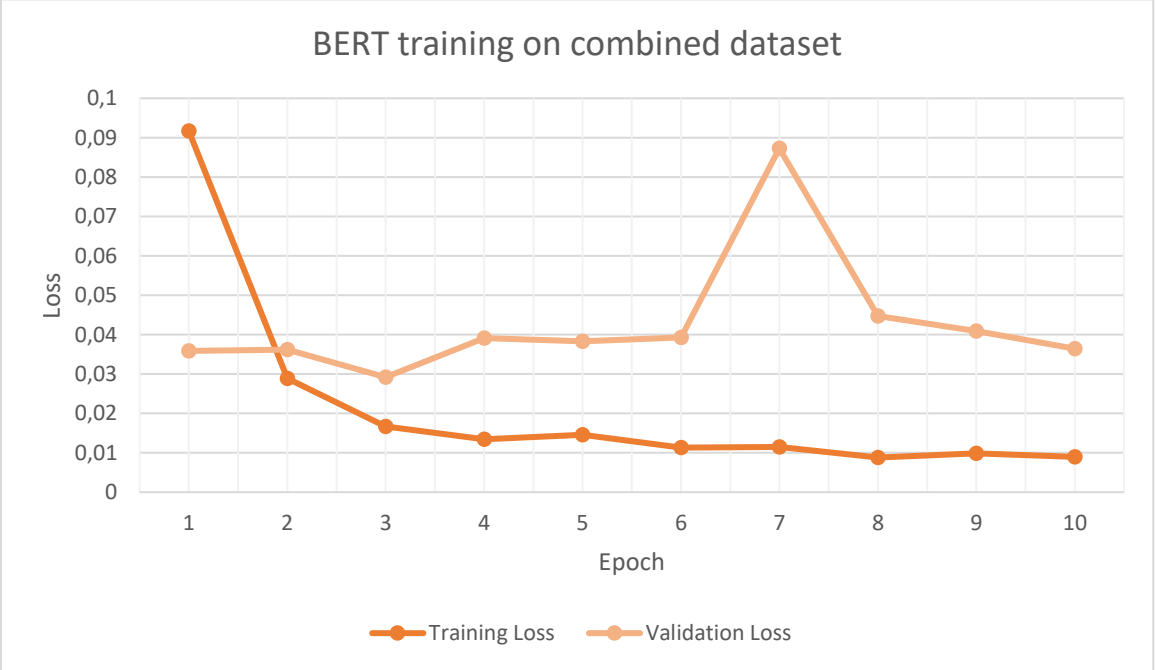

Figure 4.8: BERT's results with combined dataset: Training and validation loss across epochs

BERT completed training with the combined dataset in approximately 2 hours and 2. Already in Epoch 1, the model achieved strong results with balanced precision and recall values which point to a solid baseline performance without the tendency to over-flag or under-flag (**Figure 4.7**). Training loss starts higher than validation loss but is generally low (**Figure 4.8**). In Epoch 2 and 3 the performance stabilizes, the F1-score improves, and the training and validation loss converge. From epoch 3 to 4, validation loss has a sudden increase, but other metrics show no sign of instability. From then onwards, the metrics steadily improve until Epoch 6 where they reach their peak in F1 score. Epoch 7 represents an outlier with a sudden surge in validation loss, precision peaking and recall sharply declining - this imbalance leads to the F1 score also lowering. After that, the scores become more stabilized and follow earlier tendencies again, training and validation loss decrease while F1 score rises, and accuracy hover around the same numbers. It finishes strong with all metrics remaining near 99%.

Overall, the training results show that BERT was able to learn spam cues across the datasets quickly. The steady decline in training loss promises good adaptation to the dataset. The gap between training loss and slightly higher validation loss is healthy and indicates that the model is not purely memorizing the training set. Although the validation loss spiked in Epoch 7, the model recovers swiftly. This behavior is due to the diversity of the combined dataset preventing the model from overfitting and collapsing. The last results in Epoch 10 showed a chance of further training, leading to slight improvement in the results. However, as the training loss is already extremely low any change to the metrics would not be very meaningful as they are already remarkably high. Minor fluctuations in performance would be attributed to borderline spam/ham cases rather than weaknesses in the model itself.

4.2 DistilBERT

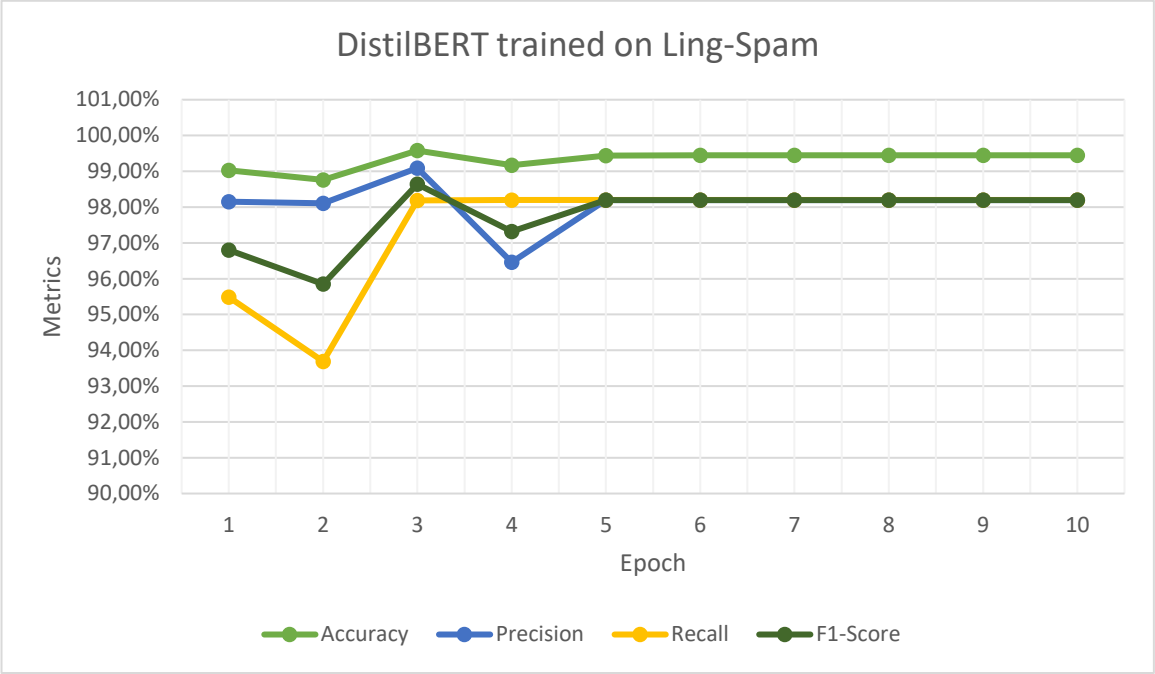

Figure 4.9:DistilBERT trained on Ling-Spam: Accuracy, precision, recall and F1 score across epochs

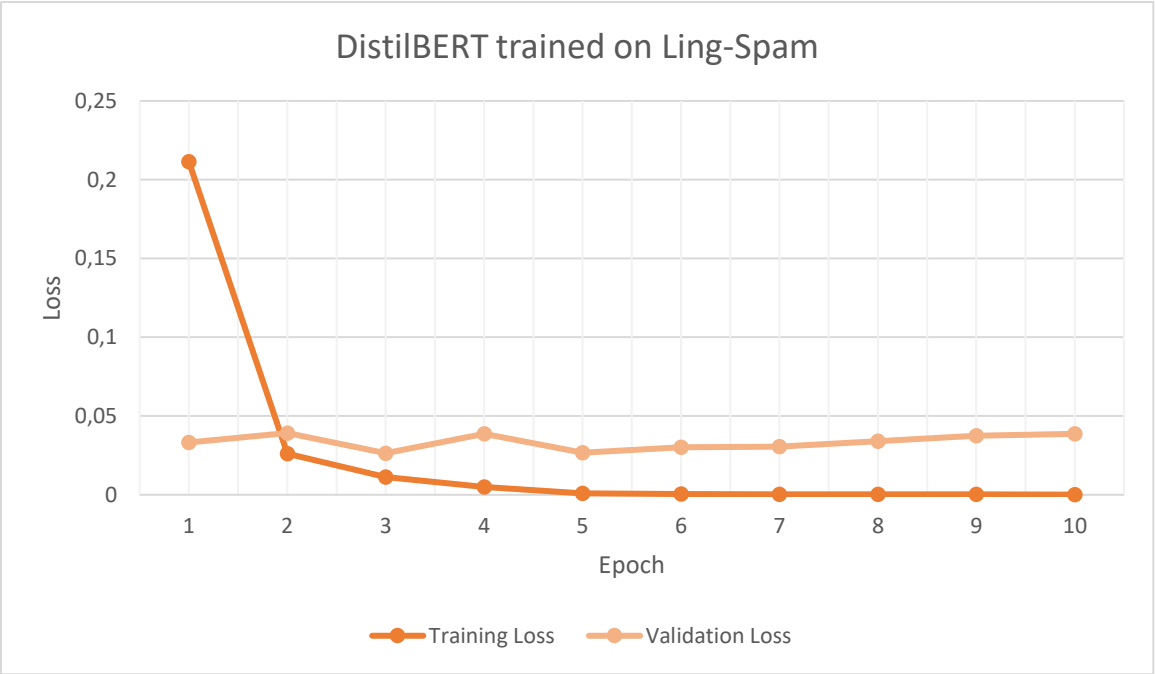

Figure 4.10: DistilBERT trained on Ling-Spam: Training and validation loss across epochs

These are the results of DistilBERT when training with Ling-Spam. The training took 4 minutes and 23 seconds to complete. First epoch results show high accuracy and slightly high training loss with a comparably low validation loss (**Figure 4.10**). Precision is higher than recall so there is a bias from Epoch 1 to 3 (**Figure 4.9**). Next, in Epoch 2 there is a big drop in training loss and precision, and all metrics seem a little weaker than before. Further in the Epochs 3 to 5 the training loss falls even lower with validation loss continuing to fluctuate. While precision and recall also fluctuate they both never drop significantly. Accuracy and F1 peak in the third epoch and generally remain stable with only slight variation. In Epoch 6 and onwards the training loss becomes extremely low, and validation loss has an upward trajectory ending higher than at the beginning. The other metrics, accuracy, precision, recall and F1 plateau after Epoch 5.

DistilBERT reaches peak performance in Epoch 3 with high accuracy and F1 score. The model achieves a very good balance of precision and recall during training early on. Further training does not result in meaningful gain and is detrimental to the model's performance with increased signs of overfitting shown through the continuous decrease in training loss and increase in validation loss. The stability or plateau in the metrics reflects the ease of which the model adapts the dataset and given the lack of change in the metrics it can be concluded that the maximum performance of the model and dataset combination has been reached within the training time.

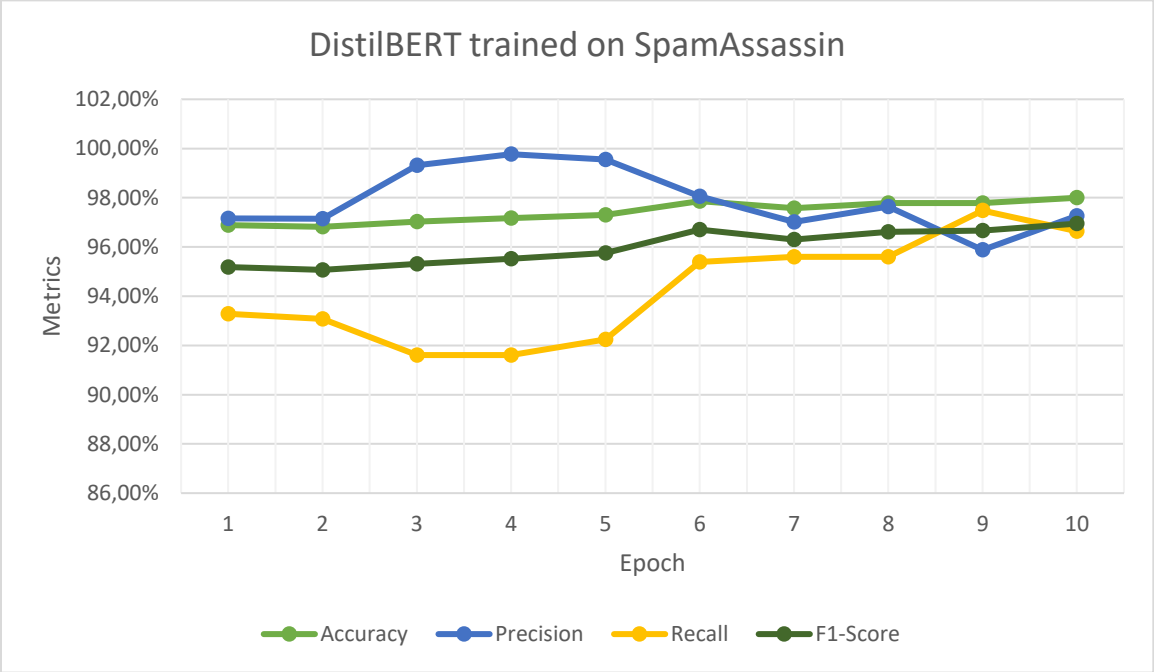

Figure 4.11: DistilBERT results on SpamAssassin: Accuracy, precision, recall and F1 score across epochs

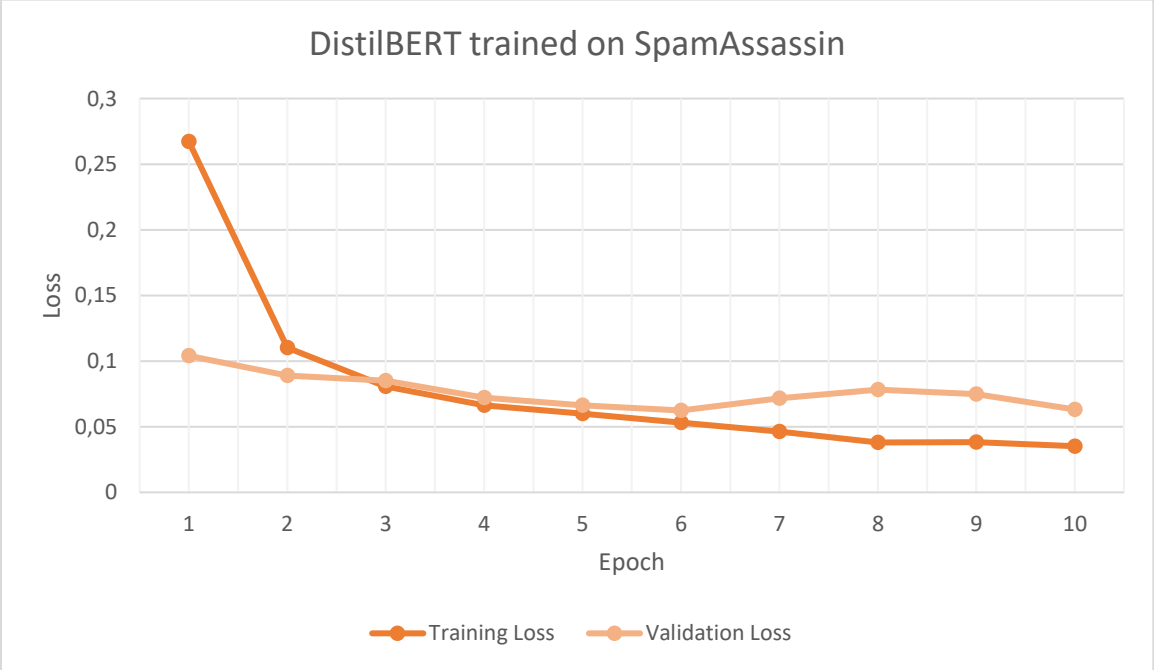

Figure 4.12: DistilBERT on SpamAssassin: Training and validation loss across epochs

The training of DistilBERT on SpamAssassin concluded in 8 minutes and 45 seconds. To start off, training loss is relatively high, but validation loss is already comparably low (**Figure 4.12**). The accuracy also seems rather high, and precision seems throughout favored over recall (**Figure 4.11**). Immediately after, both loss metrics drop while accuracy, precision and recall remain comparable to Epoch 1. In Epoch 3 and 5 the losses further decline, and accuracy rises. The precision has a sudden increase to above 99% while recall is even lower than before. There is an extremely diverging trend between them but the F1 score remains rather stable and even has an upward trend. In Epoch 6, training loss and validation loss continue their downwards decent and accuracy its upward projection. And precision and recall achieved greater balance and have converged closer to each other with the highest F1. Later, Epochs 7 and 8 show a rise in validation loss and a slight dip in F1 score. Other metrics remain similarly consistent to earlier behavior until Epoch 9. That is where recall overtakes precision, and the validation loss decreases. In the last Epoch precision passes recall again, meanwhile accuracy and F1 score continue to show an uptrend.

It is reflected in the training results with SpamAssassin, that DistilBERT can reliably filter for spam with a peak accuracy of 98% and a good balance of precision and recall. There is a steady improvement in the model's performance throughout the training until the very end. Especially in Epoch 6 the balance of precision and recall shows notable improvement from previous epochs. The validation loss fluctuates but doesn't show a trend of divergence as it lowers again, pointing to a very stable performance with slight signs of overfitting.

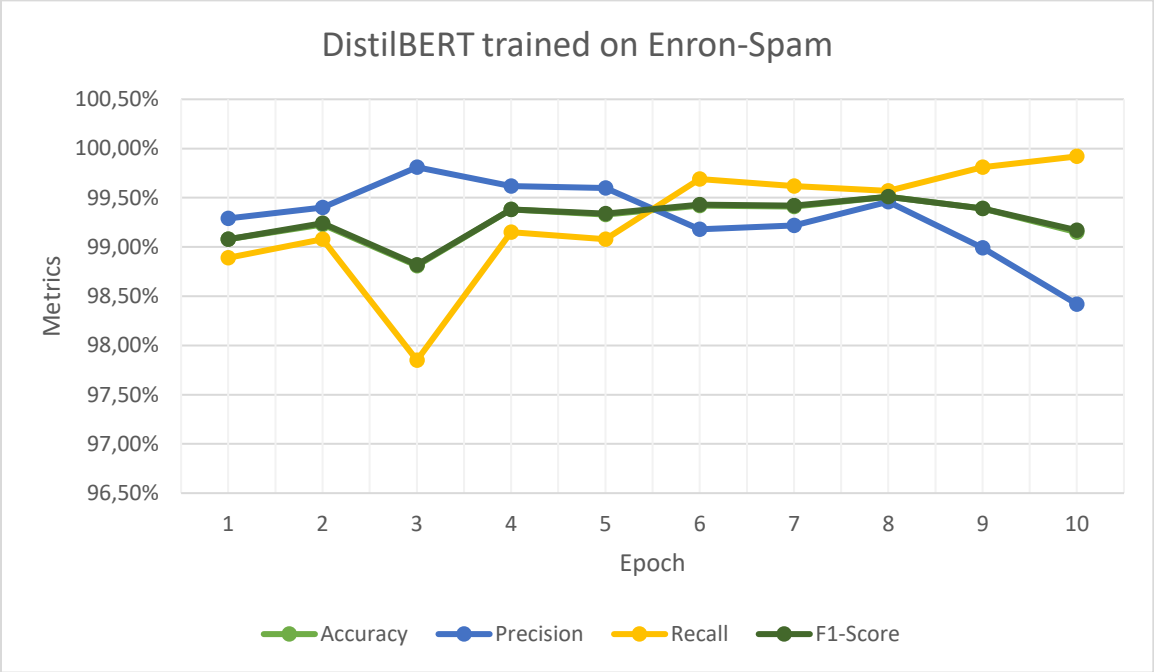

Figure 4.13: DistilBERT on Enron-Spam: Accuracy, precision, recall and F1 score across epochs

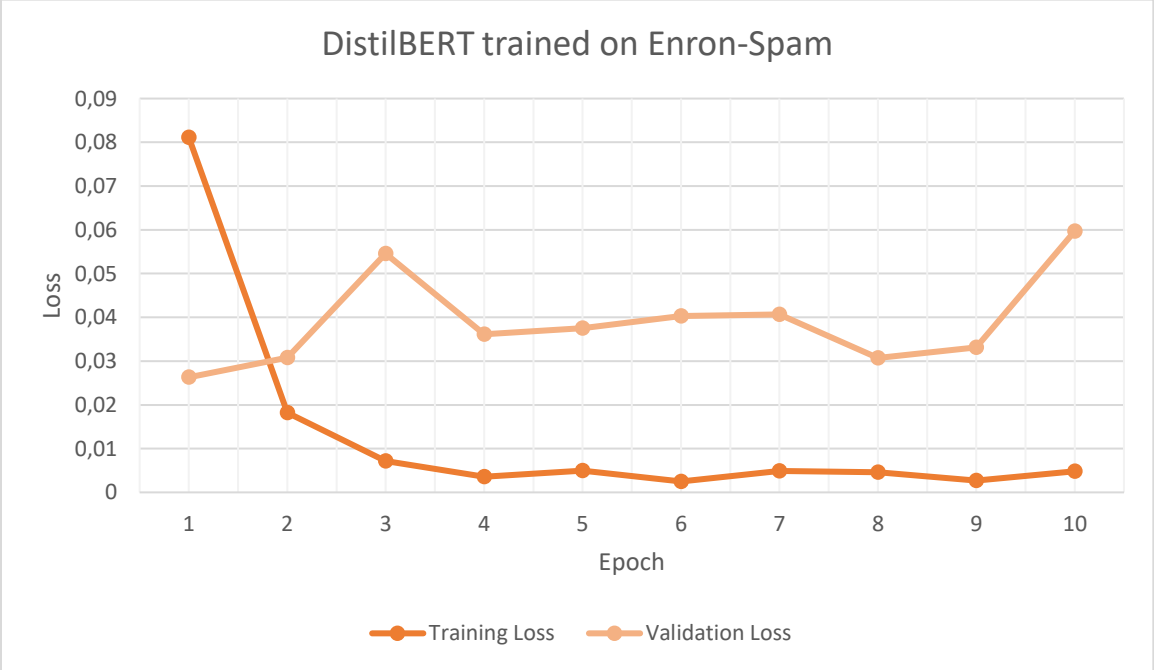

Figure 4.14: DistilBERT on Enron-Spam: Training and validation loss across epochs

DistilBERT completed training on Enron-Spam in 49 minutes and 25 seconds. In the first epoch the model surpasses 99% on accuracy, precision and F1 score (**Figure 4.13**). The early epochs results show there is a slight imbalance of favoring precision over recall until Epoch 5. From Epoch 6 onwards, recall surpassed precision, and further training led to greater convergence between the two metrics. The peak of this model's performance is achieved in Epoch 8, with its highest accuracy and F1 score. After that, recall and precision extremely diverge and accuracy along with F1 score decrease. Throughout the training, the lowest validation loss occurred in Epoch 1 after which it showed continuous increase (**Figure 4.14**). The only exception was in Epoch 8 where validation loss briefly decreased. Additional spikes in validation loss occurred in Epoch 3 and Epoch 10.

These training results show overall good metrics with a different performance depending on the training time. Early epochs have a slightly higher precision than recall with Epoch 5 being the best Epoch for precision. The peak performance occurs later in Epoch 8 after some stability. However, further training beyond that seems extremely detrimental to the model's performance with the greatest jump in validation loss and divergence of recall and precision. The validation loss almost continuously increases throughout the training which showed that the model has difficulties with generalizability despite training loss decreasing.

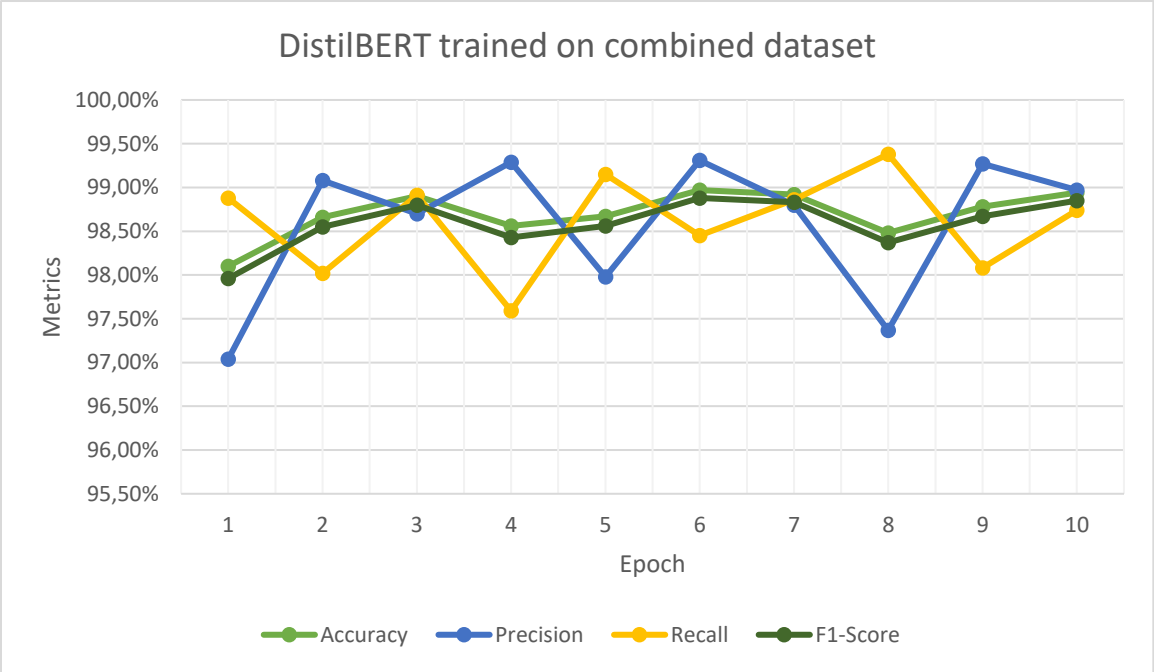

Figure 4.15: DistilBERT trained on combined dataset: Accuracy, precision, recall and F1 score across epochs

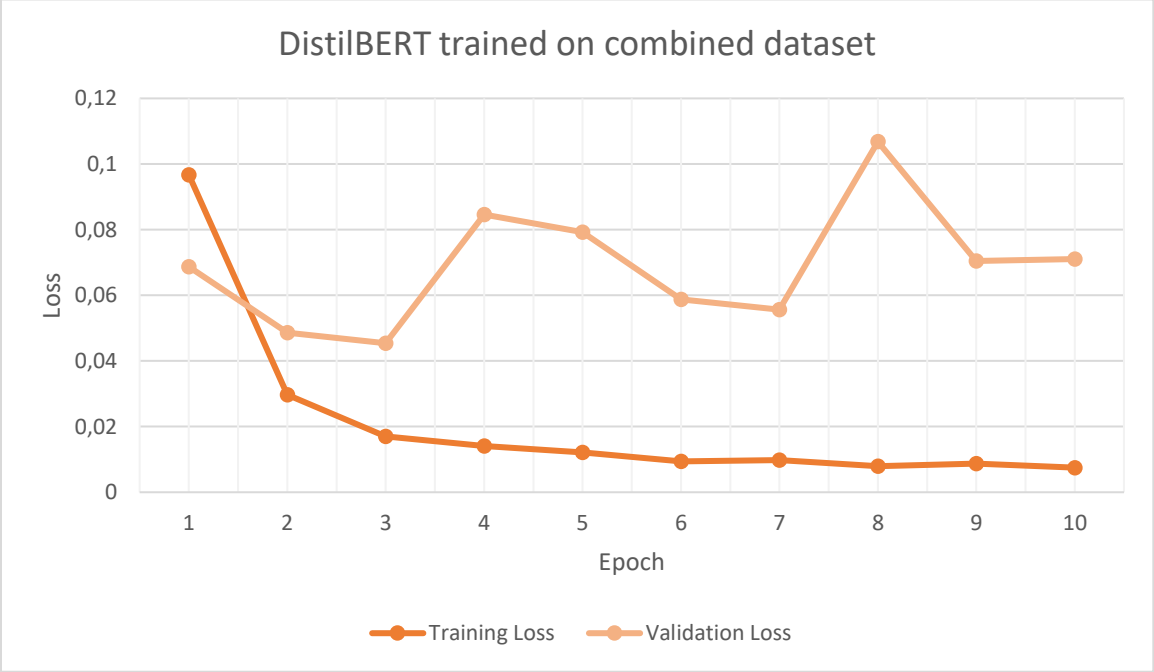

Figure 4.16: DistilBERT trained on combined dataset: training and validation loss across epochs

Training DistilBERT with the merged dataset time was a process that took 1 hour and 5 minutes. It starts in Epoch 1 with recall exceeding precision - however, precision surpasses it in the next epoch (**Figure 4.15**). This alternating pattern persists until the end of the training. All metrics show a continuous improvement until Epoch 4, where metrics take a dip. There is a sudden jump in validation loss (**Figure 4.16**), although the model recovers from instability and shows another upwards trend until Epoch 6 where it reaches its peak performance with the highest accuracy and F1 score while slightly favoring precision. Epoch 8 shows big variance with validation loss peaking once again. Later epochs show no more meaningful improvement while regaining stability.

The model was able to reach accuracy of almost 99% within 3 epochs along with stable precision and recall around 98%-99%. Throughout the training there is variation in precision or recall being favored which shows further tuning here could lead to a more decisive bias depending on the desired performance. Solely by observing

Figure 4.15 (a) There is a pattern of instability and improvement which could indicate that further training after a moment of instability could yield greater results. However, the model struggled to maintain a steady performance as further training led to greater validation loss. Within the constraints of this training the peak of this model was in epoch 6.

4.3 RoBERTa

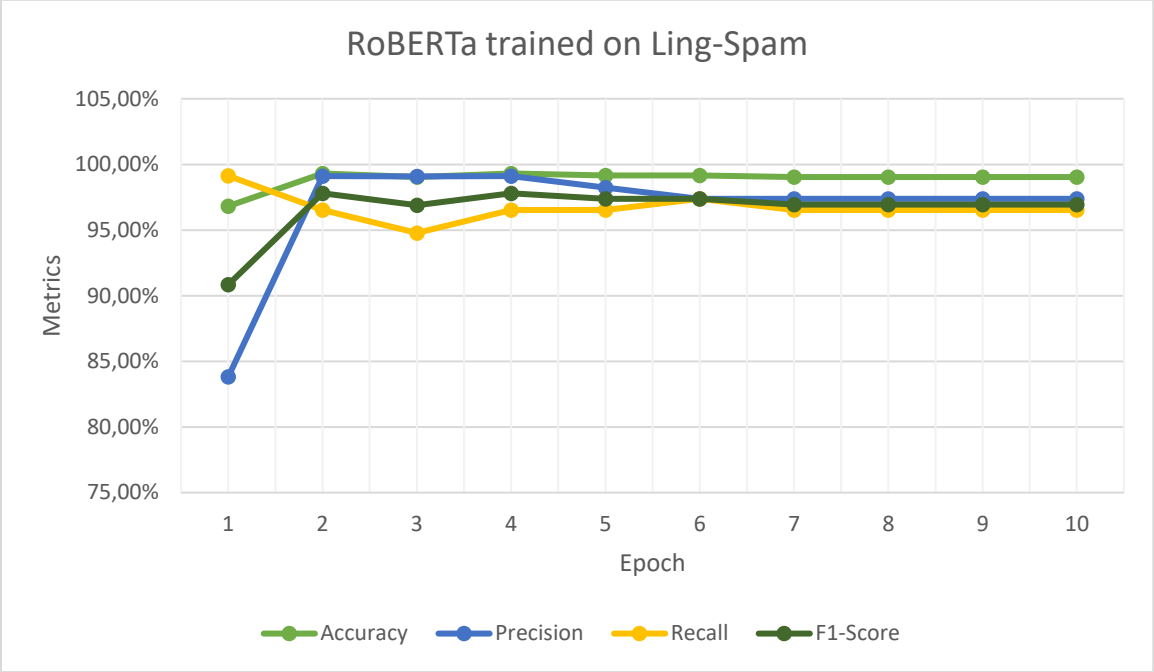

Figure 4.17: RoBERTa trained on Ling-Spam: Accuracy, precision, recall and F1 score across epochs

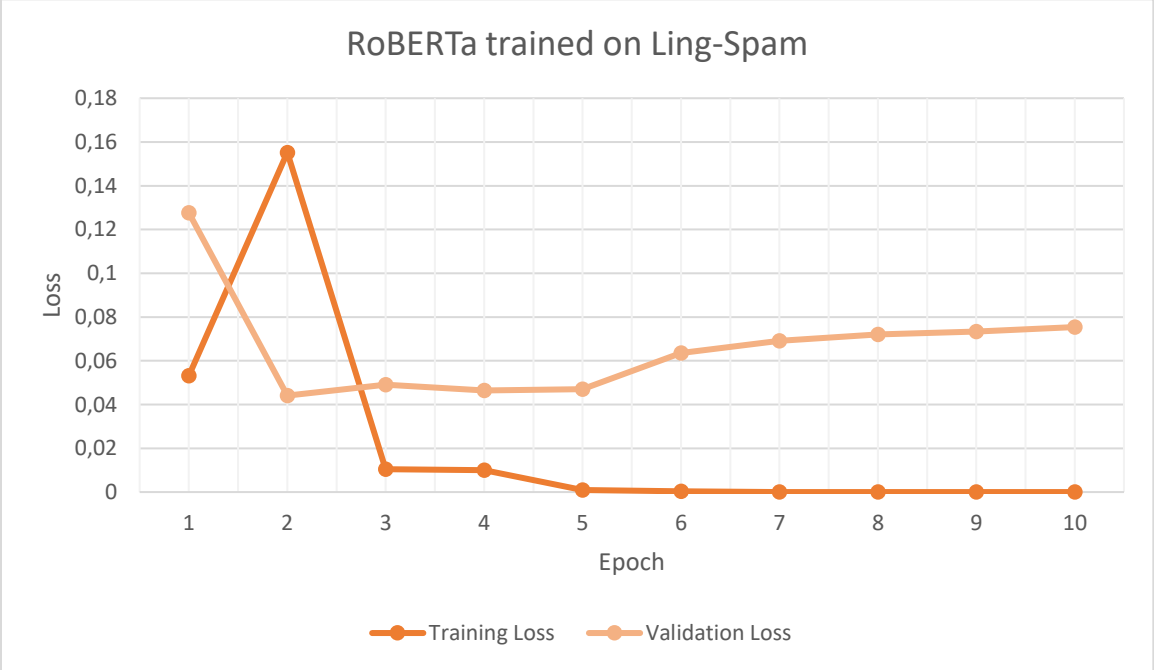

Figure 4.18: RoBERTa trained on Ling-Spam: Training and validation loss across epochs

RoBERTa completed training on the Ling-Spam dataset in 9 minutes and 20 seconds. In the first epoch, accuracy reached 97% with recall notably high and precision is comparatively low (**Figure 4.17**). The following epoch there is a sharp improvement, with accuracy and precision rising to 99% while recall dropped slightly below precision. Besides that, the model balances precision and recall well and the F1 score peaks with 98%. After that, training loss continuously approaches zero while validation starts to rise until the very end of the training (**Figure 4.18**). The other metrics falter in Epoch 3 but stabilize after. Until Epoch 6 the metrics converge and precision and recall meet. Epoch 6 is also when validation loss starts rising while training loss is approaching zero. Later epochs show no more change in the metrics; they plateau with the last stand from Epoch 7.

The plateauing of all metrics after Epoch 7 along with the divergence of training and validation loss indicates that the model has fully learned the patterns of Ling-Spam. These early signs of overfitting imply that the model's capacity was underutilized on this relatively small dataset. The model reaches its peak performance early in Epoch 2 where validation loss was the lowest and all other metrics were at their strongest. RoBERTa does not benefit from extended training with Ling-Spam and an early stopping after Epoch 2 would ensure the best possible performance metrics. In the end, RoBERTa displayed a good balance between precision and recall, although precision remained higher than recall. The bias implies a greater risk of letting spam slip into the inbox which makes the results less favorable for spam filtering.

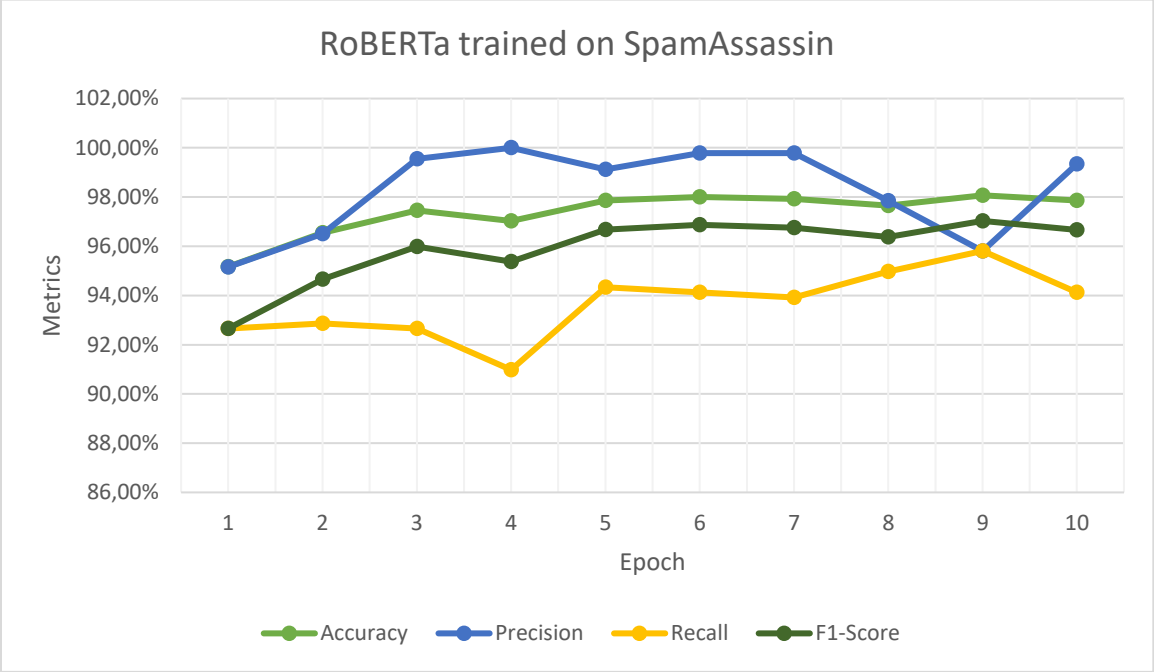

Figure 4.19: RoBERTa trained on SpamAssassin: Accuracy, precision, recall and F1 score across epochs

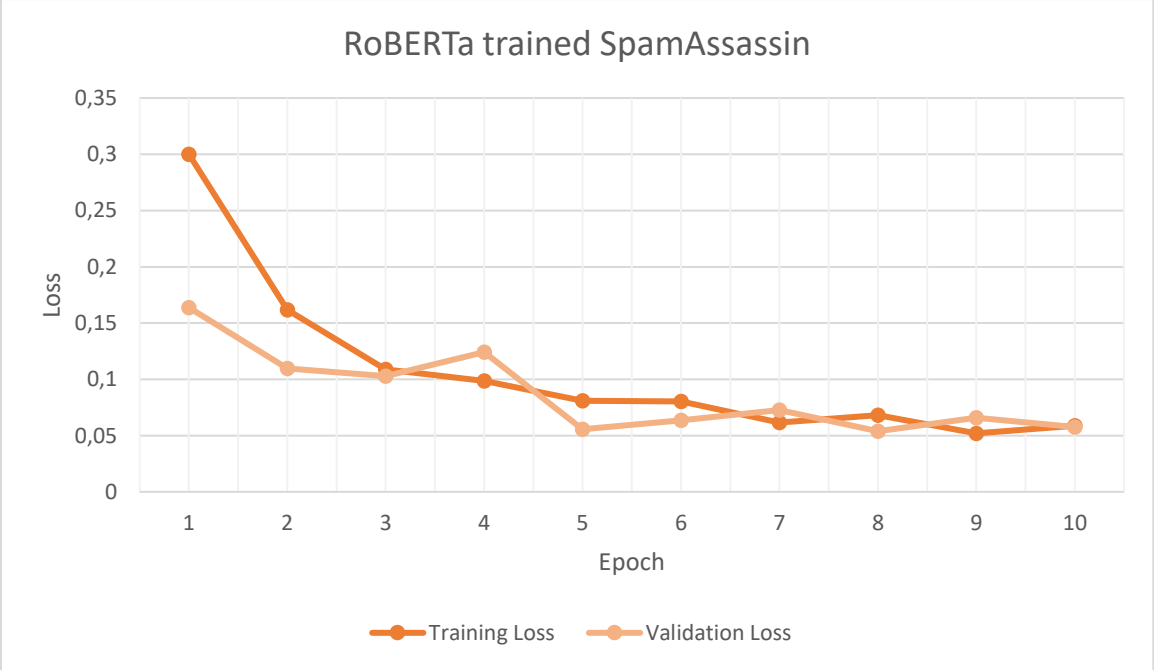

Figure 4.20: RoBERTa trained on SpamAssassin: Training and validation loss across epochs

RoBERTa trained on SpamAssassin for 19 minutes and 30 seconds. In Epoch 1 the model already achieved strong results with 95% accuracy and precision, while recall and F1 score fall slightly behind (**Figure 4.19**). At this point, the training loss and validation loss are a little high but that is to be expected of the first Epoch (**Figure 4.20**). In the next Epochs 2 and 3, almost all metrics see improvement. Only recall shows little to no improvement, while in contrast precision is extremely high at 99%. In Epoch 4 the accuracy slips down, and precision reaches 100%, meaning no false positives at all. Recall on the other hand falls to its all-time low 91%. At the same time, validation loss has unexpectedly increased. Further training improves accuracy and F1 score, precision and recall converge slowly although the bias for precision remains. Validation loss dropped below training loss in Epoch 5 and 6 and 8 before rising again. Epoch 8 marked a shift, there is a setback in accuracy and precision with an improvement in recall, the F1 score did not show improvement – however in Epoch 9, the metrics reach another peak for accuracy with 98% and for recall as well. This epoch also has the best F1 score so far. The training finished in Epoch 10 with precision surpassing recall once again and an overall dip in performance.

The model was able to improve well within the first three epochs. Early epochs extremely emphasized precision while neglecting recall, which would lead to the misclassification of many spam emails as ham. Later epochs rebalanced performance and only Epoch 9 reached no bias in either. These results show RoBERTa's tendency to prioritize precision. The validation loss seems to fluctuate slightly but doesn't show an upward trend which usually signifies overfitting. The instability could signify its difficulty in generalizing. The best performance was achieved in Epoch 9.

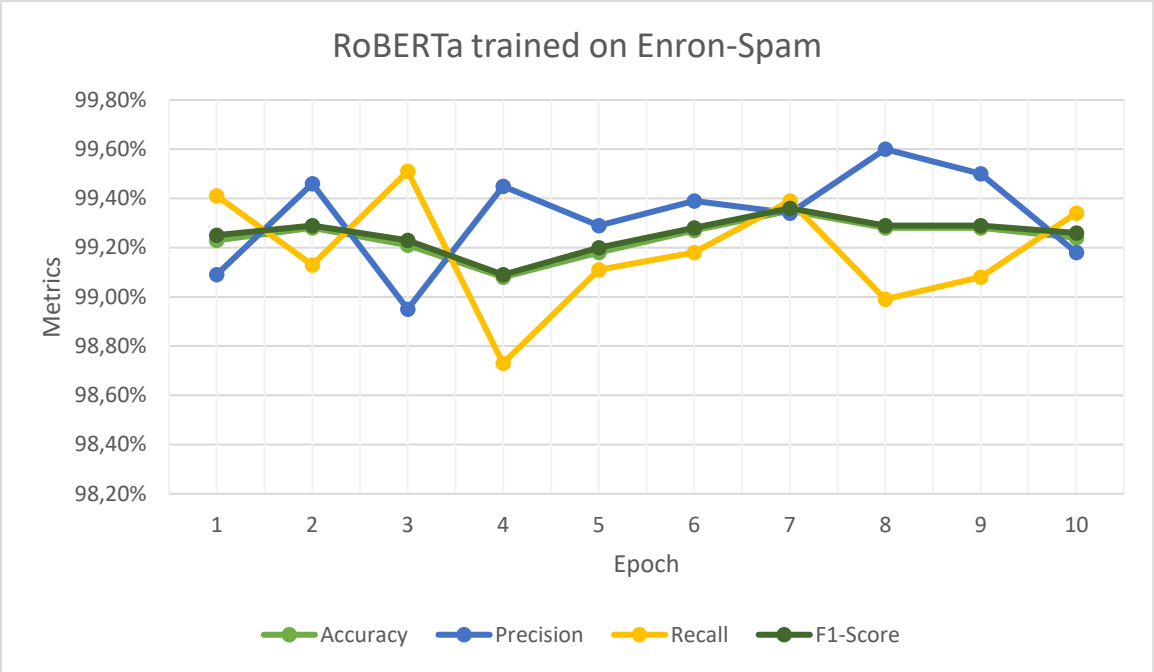

Figure 4.21: RoBERTa trained on Enron-Spam: Accuracy, precision, recall and F1 score across epochs

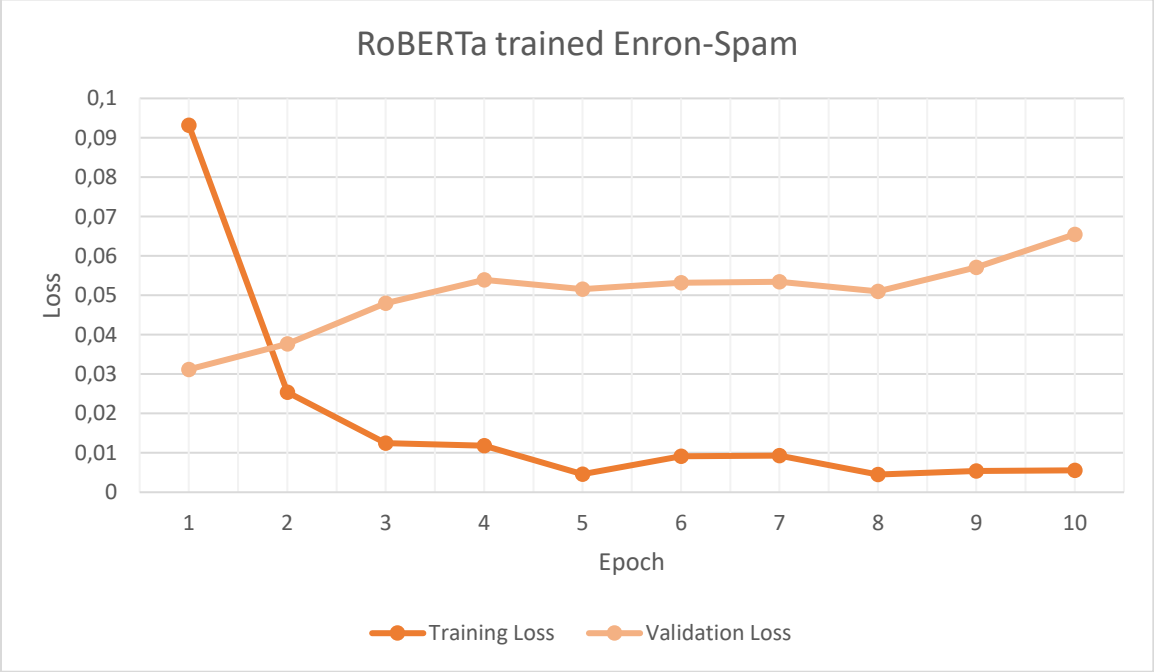

Figure 4.22: RoBERTa trained on Enron-Spam: Training and validation loss across epochs

The RoBERTa on Enron-Spam training lasted for 1 hour and 35 minutes. It starts off with strong metrics across the board – high accuracy and F1 score, well balanced precision and recall with all metrics above 99% (**Figure 4.21**). There is slight improvement in the second epoch in accuracy and F1 score but precision now overtook recall. Then in Epoch 3 and 4, accuracy and precision fall while precision and recall seem to diverge and alternate in dominance. Meanwhile validation loss has been rising although training loss was falling (**Figure 4.22**). Afterwards, in Epoch 5 all the metrics improved for a moment. While the trajectory in the metrics continues in Epoch 6, the validation loss rises once again. Further in Epoch 7 all metrics converge. This is where the best scores are achieved although the Loss graph does not show much change. The Epochs from 8 to 10 show a slow rise in validation loss, while other metrics continue to diverge and then converge again.

The model learns the dataset fast, showing very good results in the first epoch. Throughout the whole training the F1 score was consistently strong, and metrics were generally high with little exceptions. This indicates that RoBERTa was able to learn the Enron-Spam dataset well. The alternating bias between recall and precision – even with longer periods of favoring precision – imply that the model did not settle into a balance. With the model's capacity it is likely to have overfit Enron-Spams small quirks, it picked up on subtle patterns that ended up not generalizing well. This behavior shows instability, and the unreliable filtering performance is problematic for real-life applications. The peak performance was reached in Epoch 7 with the highest accuracy and F1 score which would be the optimal point to stop training. The results demonstrate that continuing training after Epoch 7 leads to even more overfitting and causes further harm to generalization.

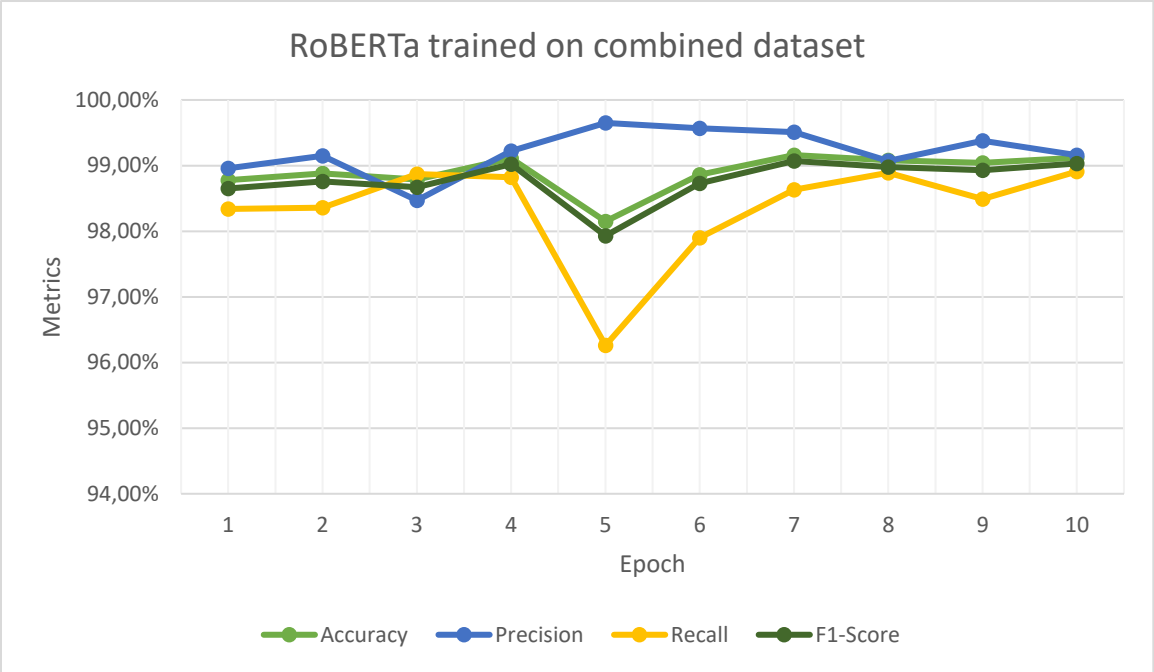

Figure 4.23: RoBERTa trained on combined dataset: Accuracy, precision, recall and F1 score across epochs

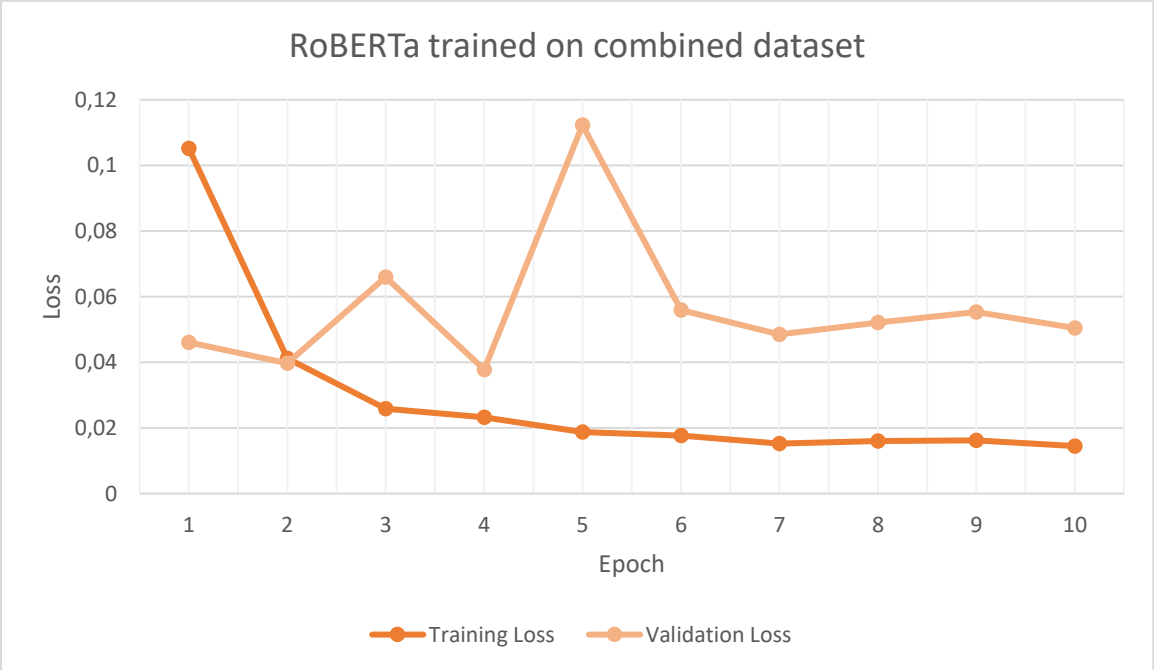

Figure 4.24: RoBERTa trained on combined dataset: Training and validation loss across epochs

RoBERTa trained for 2 hours and 19 minutes on the combined dataset. Initial training loss is slightly high with validation loss being comparatively low (**Figure 4.24**), while other metrics perform well close to 99%. Second epoch leads to a big drop in training loss and shows slight improvement in all other metrics (**Figure 4.23**). Afterwards, in Epoch 3 validation loss has an unexpected increase. The other metrics converge and overall perform slightly worse. However, the model recovers in Epoch 4 - validation loss drops back down, accuracy and F1 both reach a new peak. Afterwards in Epoch 5 there is another spike in validation loss and other metrics worsen. Precision and recall diverge extremely and accuracy and F1 score drop. Additionally, both accuracy and recall fall to their lowest value in the training. Following Epoch 6 all metrics recover, and precision and recall gradually converge again until Epoch 8. At this point, validation loss also rises again but so does training loss. After that, Epoch 9 shows inconsistency again with loss rising and metrics worsening. In the final epoch, the model improves slightly, stabilizing once more.

Overall, RoBERTa attained high results early on and maintains them throughout the training. Apart from Epoch 5 the metrics seem rather high with an accuracy of around 99% and precision, recall and F1 score consistently high as well. The instability that occurred may be attributed to the diversity of the combined dataset. The fluctuation suggests that the model over-adapted to patterns from one dataset segment which led to the decrease of generalization. However, the ability to recover after each fluctuation demonstrates the model's robustness. Best overall results were achieved at Epoch 7 although Epoch 4 also had similarly good results, where validation loss is very low and F1 peak the highest. The metrics hover around 99% after Epoch 7 suggest that the model has reached its full potential with the training and would not improve much with further training.

5 Evaluation

In this evaluation, I compare the performance of BERT, DistilBERT and RoBERTa on multiple spam datasets. Points to consider are the effectiveness and generalizability in spam detection. Each chosen model represents a different point of interest during their conceptualization. Therefore, it is expected that the models will perform differently. The results of BERT should serve as a baseline for performance with DistilBERT expected to have a meaningful decrease in its running time with slightly worse overall results and RoBERTa's results being noticeably better.

For the sake of unbalanced datasets and their real-world applications such as here: Spam filtering where the occurrence of spam is usually smaller than ham - accuracy alone is not a reliable metric. For the intended application, precision and recall become more important and the balance between them which is shown through the F1 score. In this context high recall means that few spam mails go undetected. Low recall therefore is interpreted as the model struggling to correctly label spam emails. High precision signifies a low number of false alarms, so real mails aren't incorrectly labeled as spam often. In turn, low precision indicated that the model often incorrectly flags ham as spam. For an average user, spam making it into the inbox is less detrimental than losing important emails in the spam folder. On the other hand, security would be increased if spam was more harshly filtered. To fulfill the needs of the user and the standards of a secure and trustworthy spam filter, a balance between those two metrics would be ideal. F1 score will be the most prioritized metric along with the consideration of their bias towards either precision or recall.

Our first dataset to look at is Ling-Spam. When looking at the peak performance of each model the metrics are very comparable. All models reach their peak performance results within 3 epochs and have an accuracy of over 99%. DistilBERT scored the highest in accuracy while BERT and RoBERTa surprisingly have the same accuracy. Next with precision, RoBERTa performed the best followed by DistilBERT and lastly BERT. Recall had their order reversed and lastly the highest F1 score was achieved by DistilBERT with BERT second and RoBERTa last.

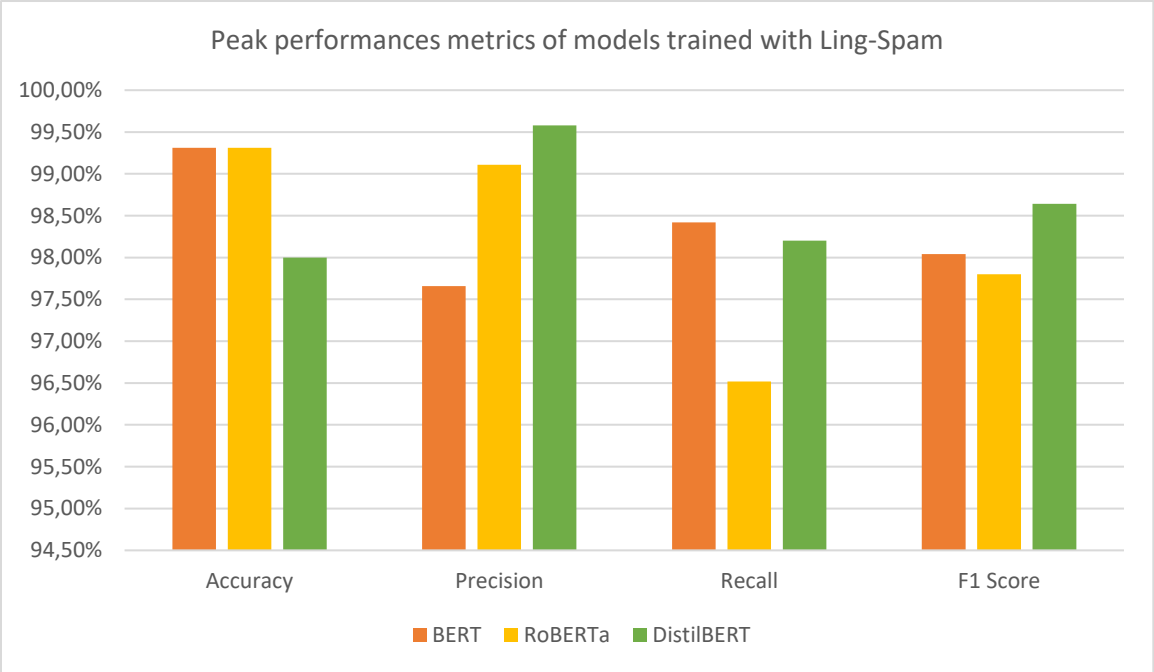

Figure 5.1: Peak performances metrics of all models trained using Ling-Spam

Ling-Spam is a relatively small dataset with not enough complexity to push the performance of every model further. This is shown in all their plateaued performances. BERT hit its limitations the earliest in Epoch 3. While it shows that BERT was able to learn quickly, it reached its highest capacity and showed no signs of improvement. RoBERTa reached its peak even earlier at Epoch 2 and plateaued the latest in Epoch 7 with slightly worse results than its peak. It reflects its capacity to pick up on the most obvious spam and ham patterns very quickly and tried to adapt longer than the other models. DistilBERT oscillated around the same performance metrics after its peak but ultimately plateaued Epoch 5 onwards. It's a lighter model with fewer parameters making it less prone to overfitting on smaller datasets, explaining the slightly higher peak performance over the other two. All the models showed signs of overfitting - learning too many specific quirks that were not very generalizable. Their training losses were close to zero in the middle of the training already and therefore could not improve anymore.

Because of Ling-Spam's size it was foreseeable that the models were able to learn the dataset quickly. Not only that, but it was also mainly the reason for the lack of further improvement in prolonged training. For this specific dataset BERT, DistilBERT and RoBERTa performed similarly with all their metrics reaching a plateau. Their difference becomes apparent when it comes to the precision and recall as BERT is the only model that favored recall in its peak performance. RoBERTa's strength – the ability to deal well with complex data – could not be capitalized on with this dataset and task. In the end, DistilBERT performed best with the highest F1 score (**Figure 5.1**). Notable is despite those results its accuracy is also the lowest of the three. The imbalance of the dataset is likely to be the cause. Despite this, for the spam and ham classification task with Ling-Spam's data DistilBERT performed most efficiently – able

to complete the training ins half of the time and achieving the best balance between the two prioritized metrics.

The next training with the dataset SpamAssassin resulted in on average lower results than Ling-Spam. Across all metrics BERT performed the best in Epoch 6 with highest accuracy and F1 score. After its peak, all metrics slightly decline until the end of the training. RoBERTa's performance trails behind with fluctuating growth and its best performance achieved in Epoch 9. Then closely after comes DistilBERT, with a steady growth of learning and performance that peaked in the last epoch.

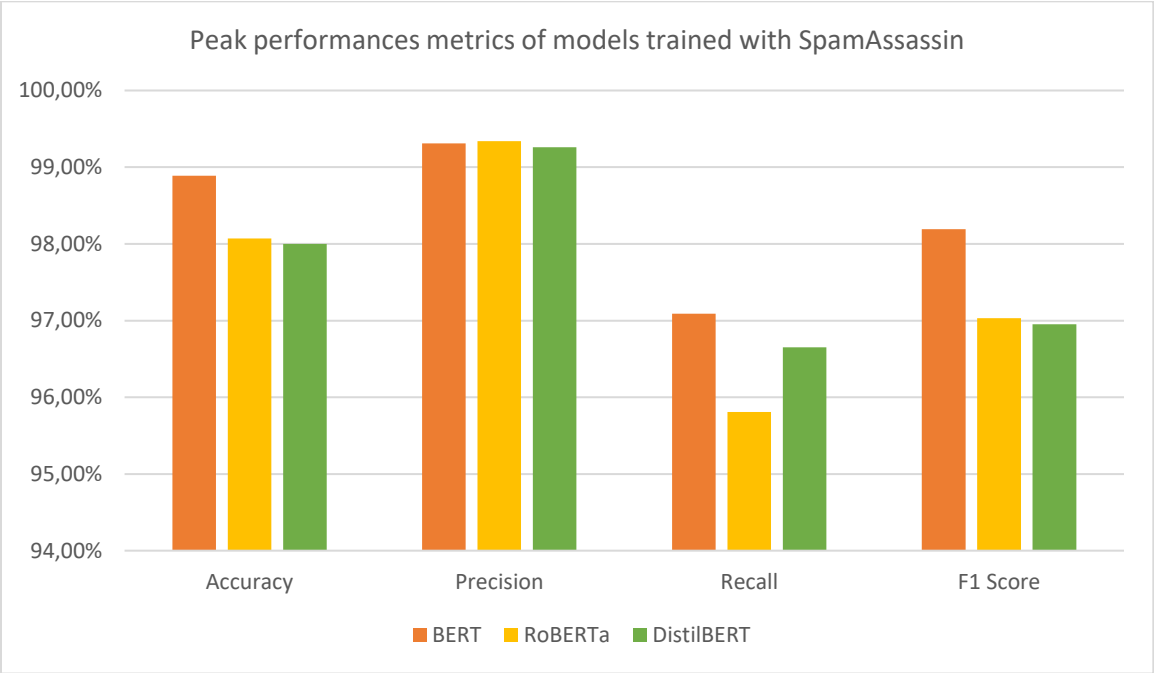

Figure 5.2: Peak performances metrics of all models trained using SpamAssassin

When considering only peak performances, DistilBERT achieved the lowest results among the three models. However, its learning curve was consistent and steadily improving (**Figure 4.12**). This suggests that with additional training epochs, DistilBERT could potentially achieve even better results. It is in that sense a competitive and promising result, especially compared to RoBERTa. RoBERTa reached its peak results after what looks like a sudden convergence and immediate divergence after (**Figure 4.19**). The training in that sense does not seem comparatively stable although the peak performance metrics are similar. BERT's curve (**Figure 4.3**) is comparably more stable looking although similarly expected to get worse results with further training. Despite the large capacity of RoBERTa, it once again failed to outperform the other models, indicating difficulty in generalizing on this dataset.

These results for the SpamAssassin dataset show flexibility in the optimal choice of model depending on the goal. The imbalance of the dataset can be reflected in all the model's bias towards precision. Because the minority class is spam, this led to the models making more conservative spam predictions.

If priority is maximizing performance, then BERT would be the most suitable option, achieving the highest accuracy and F1 score during its training (

Figure 5.2). However, if efficiency is the priority, then DistilBERT provides a better trade-off – delivering comparable but slightly worse results but saving on computational cost.

Next, there are the training results using Enron-Spam which are very high with each model peaking at over 99% across all metrics. DistilBERT started off with strong results and showed improvement until its peak at Epoch 8 (**Figure 4.13**). Similar behavior was showcased by BERT although it achieved its peak slightly earlier in Epoch 7. RoBERTa started with initially higher metrics but also peaked at Epoch 7. Both BERT and DistilBERT ultimately surpassed it in final performance.

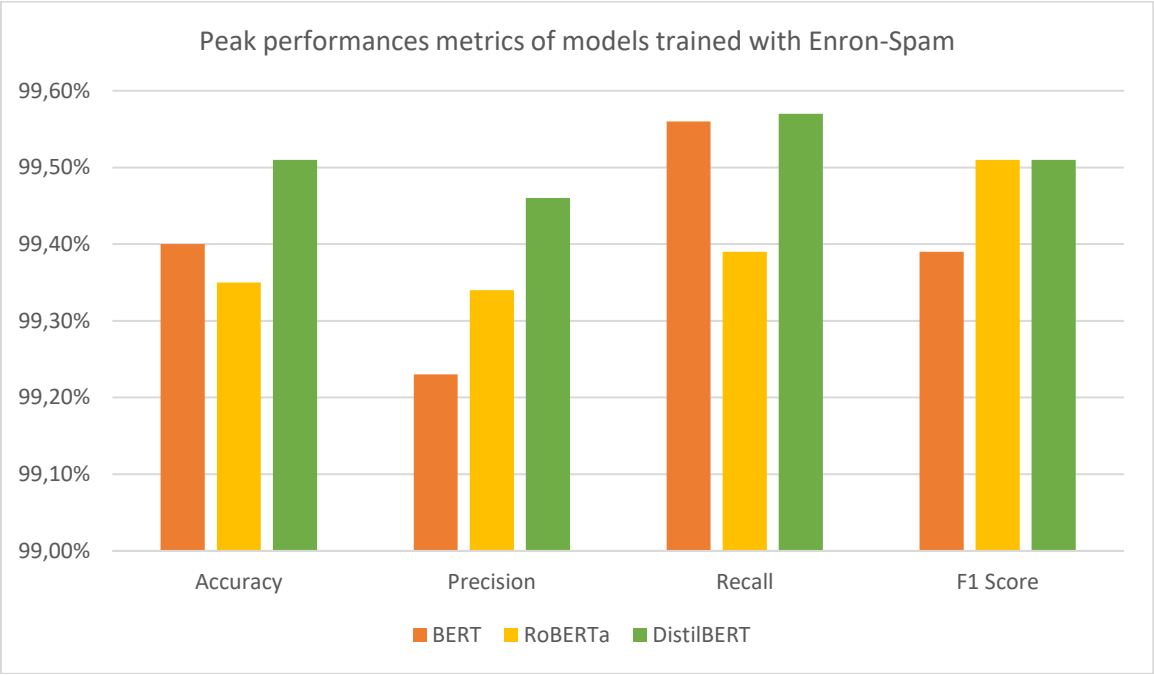

Figure 5.3: Peak performances metrics of all models trained with Enron-Spam

DistilBERT surprisingly outperformed both BERT and RoBERTa on the Enron-Spam dataset. Despite fewer parameters it was able to capture spam and ham patterns very effectively. BERT and RoBERTa both showed volatility in their bias towards precision or recall while DistilBERT had in initial epochs preferred precision and in the later ones recall. RoBERTa's F1 score is only slightly behind DistilBERT. Its pretraining - which would be helpful for more complex data – offered little advantage with the Enron-Spam dataset. Another noteworthy aspect is that all models achieved stronger recall than precision, indicating a lower tendency for predicting false. In the end the training with DistilBERT led to the most balanced and effective model for this dataset, BERT was more polarized in its bias and RoBERTa overall achieved lower metrics.

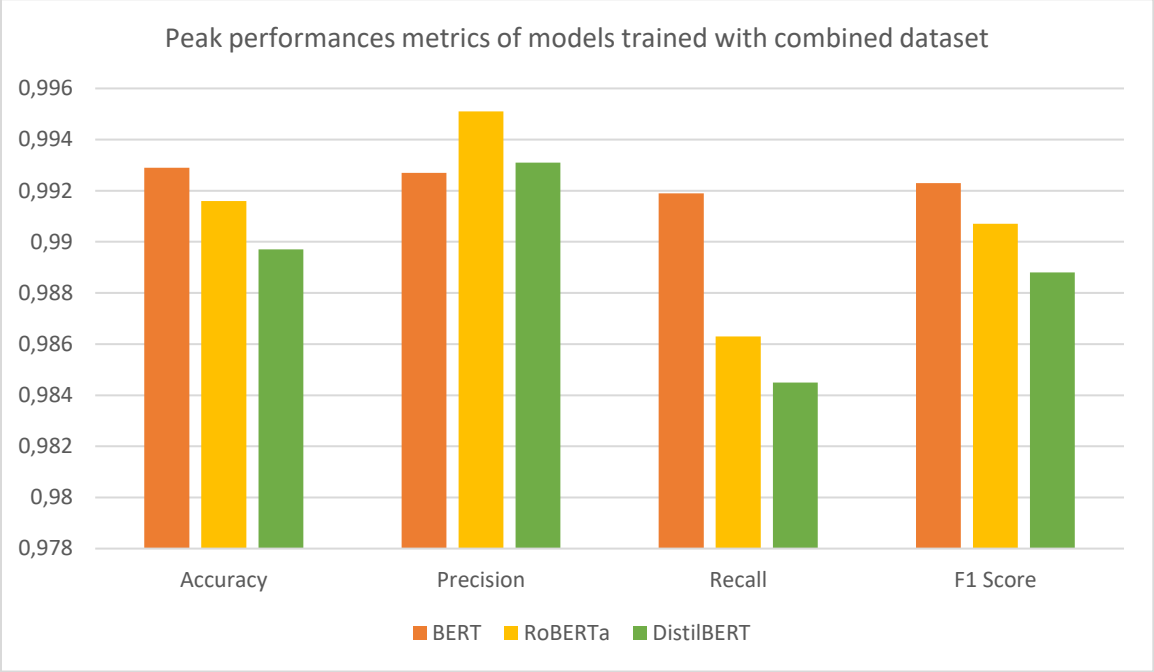

Figure 5.4: Peak performances metrics of all models trained with the combined dataset

BERT achieved the highest F1 score and accuracy when training on the combined dataset in Epoch 6. Its training curve (**Figure 4.7**) looked extremely stable compared to DistilBERT's which seemed to struggle as seen from the alternating precision and recall peaks (**Figure 4.15**). There is a slight bias towards precision although it is not as drastic as RoBERTa or DistilBERT (**Figure 5.4**). Similar to the results with previous datasets, RoBERTa performed strongly, however less stable and with a lower peak than BERT, but it did outperform DistilBERT. Although DistilBERT ranks third in F1 score and accuracy at its peak, it did only need half the time of its competitors. For maximizing the performance metrics BERT is the best choice. Same as earlier, DistilBERT proves itself as a good alternative should efficiency be more of a concern. RoBERTa underperforms next to the base BERT model although it was expected to have the best performance with the now increased size of data and complexity of data.

To summarize, DistilBERT performed best when training with small to medium dataset with less complexity, excelling on Ling-Spam, SpamAssassin and Enron-Spam when used separately. Its smaller size and reduced parameter count allowed it to generalize well without overfitting. Across these datasets, DistilBERT showed both tendencies towards higher recall and precision, showcasing flexibility in its ability to be fine-tuned.

On the other hand, BERT performed strongest with the combined dataset which had a larger amount of data and a more diverse collection of ham and spam. This wider range of data provided the model with enough complexity to fully leverage its high capacity. BERT especially then was able to have a very

good balance of precision and recall, necessitating a sufficient amount of data before such performance can be achieved.

Meanwhile, RoBERTa demonstrated the most inconsistent results. While it could keep up with the other models, especially when dealing with larger collections like Enron-Spam and the combined dataset, its development throughout the training was more volatile. The sensitivity and inconsistency can be traced back to its large capacity, it was likely learning each datasets quirk too well. On smaller datasets it underperformed compared to BERT and DistilBERT. In this way, RoBERTa performed contrary to expectation, not being able to outperform the other models as initially thought. The results indicate that the optimizations and larger capacity of RoBERTa are not suited for the simple classification of spam and ham. It goes further to show that there is a misalignment between the pretraining and given task.

Lastly, the results highlight the importance of matching model complexity to dataset size and task requirement. While DistilBERT proved most efficient and very stable with smaller dataset, BERT was able to reach its full potential best with larger and more varied data. RoBERTa's underperformance and instability shows that more advanced and resource-heavy models do not guarantee better results for spam detection tasks. In the end all results were very high and differ by small percentages.

It should be noted that these results are based on singular training runs per model. Variability through random data splits and training dynamics between runs may influence performance. Therefore, while the observed behavior is useful insight, they should not be interpreted as absolute performance rankings.

6 Conclusion

This thesis aimed to evaluate the effectiveness of BERT-based models for spam email classification across multiple datasets of varying size and complexity. As spam emails continues to pose a cybersecurity threat to users – ranging from mild annoyance to major threat that could lead to malware and phishing – efforts to work on countermeasures remain crucial. By comparing BERT, DistilBERT and RoBERTa, this work highlights how the difference in their capacity and efficiency influence spam classification performance.

The results show that DistilBERT demonstrates strong capabilities in spam detection, despite having fewer parameters than BERT and RoBERTa. It performed the best on Ling-Spam and Enron-Spam, where lighter architecture allowed for better generalization and reduced overfitting risk. BERT displayed its potential on larger, more diverse datasets including SpamAssassin and the combined dataset. It achieved strongest overall performance due to its higher capacity and ability to leverage richer data. RoBERTa, which was expected to outperform both models with its optimized pretraining, exhibited inconsistent and volatile results. Its advantage did not translate to better results, suggesting a mismatch of its pretraining and this task or these datasets. Across all datasets, models achieved high performance, but their bias towards recall or precision varied depending on dataset size, class balance or complexity.

These findings emphasize that model selection for spam detection must consider the characteristics of the dataset and the deployment requirements. If computational efficiency and fast training times are of importance, then DistilBERT would be able to fulfill the task with a worthwhile trade-off between cost and performance. On the other hand, if the goal is to maximize performance on large and complex datasets, then BERT would be a more reliable option. While RoBERTa is theoretically the most powerful model, the results demonstrate that higher capacity alone does not translate to superior performance. These findings further highlight that mismatch between model pretraining and task requirement can reduce effectiveness.

The results of this thesis need to be considered alongside its limitations. One of which being that the variance across multiple runs is left unexplored and should be included in future work to provide statistically more telling comparisons. Furthermore, the datasets used, while standard in spam classification research, were created more than ten years ago. The data is therefore not representative of today's circulating emails. Training on a dataset that is more representative of today's spam could shed light on how spam evolved and if the models need to be more sophisticated to keep up these high-performance metrics. Additionally, as this study was restricted to three transformer-based models, extending the comparison to additional architecture could provide more insight as well.

In conclusion, BERT-based models are effective tools for spam classification, but their suitability depends on dataset characteristics and deployment requirements. Understanding the trade-offs between

model capacity and dataset complexity is essential for benchmarking and to developing a practical, reliable and high-performing defense against the evolving landscape of spam and phishing mail.

Bibliography

Ashish Vaswani, N. S. (2023, 08 02). *Attention Is All You Need*. Retrieved September 11, 2025, from arXiv:1706.03762 [cs.CL]

Informationstechnik, B. f. (n.d.). *Social Engineering - the "Human Factor"*. Retrieved September 11, 2025, from BSI – Bundesamt für Sicherheit in der Informationstechnik: https://www.bsi.bund.de/EN/Themen/Verbraucherinnen-und-Verbraucher/Cyber-Sicherheitslage/Methoden-der-Cyber-Kriminalitaet/Social-Engineering/social-engineering_node.html

Ion Androutsopoulos, J. K. (2000, 06 07). *An Evaluation of Naive Bayesian Anti-Spam Filtering.* Retrieved September 11, 2025, from arXiv:cs/0006013 [cs.CL]

Jack W. Rae, S. B. (2021, 12 08). *Scaling Language Models: Methods, Analysis & Insights from Training Gopher*. Retrieved September 11, 2025, from arXiv:2112.11446 [cs.CL]

Jacob Devlin, M.-W. C. (2018, 10 11). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Retrieved September 11, 2025, from arXiv:1810.04805 [cs.CL]

Kaggle. (n.d.). *How To Use Kaggle*. Retrieved September 11, 2025, from Kaggle: https://www.kaggle.com/docs

Kamran Kowsari, K. J. (2020, 05 20). *Text Classification Algorithms: A Survey*. Retrieved September 11, 2025, from arXiv:1904.08067 [cs.LG]

Lewis Tunstall, L. v. (2022). *Natural Language Processing with Transformers, Revised Edition.* O'Reilly Media, Inc.

Malte Josten, T. W. (2024, 08 26). *Investigating the Effectiveness of Bayesian Spam Filters in Detecting LLM-modified Spam Mails*. Retrieved September 11, 2025, from arXiv:2408.14293 [cs.CR]

Mason, J. (2006, 01 31). *Apache SpamAssassin*. Retrieved September 11, 2025, from spamassassin.apache.org: https://spamassassin.apache.org/old/publiccorpus/

Rich Caruana, S. L. (2001,05) *Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping.* Retrieved September 11,2025, from ResearchGate: https://www.researchgate.net/publication/2326149_Overfitting_in_Neural_Nets_Backpropagation_Conjugate_Gradient_and_Early_Stopping

Sarah Barrington, E. A. (2025, 01 25). *People are poorly equipped to detect AI-powered voice clones*. Retrieved September 11, 2025, from arXiv:2410.03791 [cs.HC]

Sophie Henning, W. B. (2023, 02 22). *A Survey of Methods for Addressing Class Imbalance in Deep-Learning Based Natural Language Processing*. Retrieved September 11, 2025, from arXiv: 2210.04675 [cs.CL]

Vangelis Metsis, I. A. (2006, 07 27). *Spam Filtering with Naive Bayes – Which Naive Bayes?* Retrieved September 11, 2025, from Athens University of Economics and Business (AUEB): https://www2.aueb.gr/users/ion/docs/ceas2006_paper.pdf

Victor Sanh, L. D. (2020, 05 01). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. Retrieved September 11, 2025, from arXiv:1910.01108 [cs.CL]

Yinhan Liu, M. O. (2019, 06 26). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. Retrieved September 11, 2025, from arxiv:1907.11692 [cs.CL]

Zige Wang, W. Z. (2024, 08 02). *Data Management For Training Large Language Models: A Survey*. Retrieved September 11, 2025, from arXiv:2312.01700 [cs.CL]

Appendix

Tables of evaluated experiment results

BERT trained on Ling-Spam

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1-Score
1	0,1778	0,038	0,9889	0,9612	0,9763	0,9687
2	0,0122	0,0438	0,9917	0,9689	0,9842	0,9765
3	0,0043	0,0363	0,9931	0,9766	0,9842	0,9804
4	0,0003	0,0385	0,9931	0,9766	0,9842	0,9804
5	0,0002	0,0411	0,9931	0,9766	0,9842	0,9804
6	0,0001	0,0474	0,9931	0,9766	0,9842	0,9804
7	0,0001	0,0443	0,9931	0,9766	0,9842	0,9804
8	0	0,0453	0,9931	0,9766	0,9842	0,9804
9	0	0,0458	0,9931	0,9766	0,9842	0,9804
10	0	0,0467	0,9931	0,9766	0,9842	0,9804

BERT trained on SpamAssassin

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1-Score
1	0,2919	0,1005	0,9724	0,9573	0,9530	0,9551
2	0,1127	0,1054	0,9572	0,8937	0,9776	0,9338
3	0,0873	0,0717	0,9779	0,9621	0,9664	0,9643
4	0,0809	0,0534	0,9848	0,9908	0,9597	0,975
5	0,0694	0,0416	0,9869	0,9886	0,9687	0,9785
6	0,0563	0,0401	0,9889	0,9931	0,9709	0,9819
7	0,0549	0,0467	0,9883	0,9931	0,9687	0,9807
8	0,0580	0,0496	0,9855	0,9885	0,9642	0,9762
9	0,0512	0,0673	0,9807	0,9544	0,9843	0,9692
10	0,0366	0,0710	0,9765	0,9441	0,9821	0,9627

BERT trained on Enron-Spam

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1-Score
1	0, 0743	0,0311	0,9906	0,9924	0,9886	0,9905
2	0, 0148	0,0429	0,9910	0,9941	0,9876	0,9909
3	0,0058	0,0344	0,9930	0,9903	0,9956	0,9930
4	0,0062	0,0501	0,9914	0,9941	0,9886	0,9914
5	0,0037	0,0363	0,9929	0,9942	0,9915	0,9928
6	0,0042	0,0416	0,9934	0,9906	0,9961	0,9933
7	0,0024	0,0394	0,9940	0,9923	0,9956	0,9939
8	0,0046	0,0441	0,9924	0,9918	0,9930	0,9923
9	0,0050	0,0678	0,9887	0,9948	0,9823	0,9885
10	0,0034	0,0419	0,9920	0,9891	0,9947	0,9919

BERT trained on combined dataset

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1-Score
1	0,0917	0,0359	0,9886	0,9894	0,9857	0,9875
2	0,0289	0,0362	0,9899	0,9958	0,9822	0,9889
3	0,0167	0,0292	0,9924	0,9944	0,9890	0,9917
4	0,0134	0,0391	0,9918	0,9917	0,9905	0,9911
5	0,0146	0,0383	0,9927	0,9923	0,9917	0,9920
6	0,0113	0,0393	0,9929	0,9927	0,9919	0,9923
7	0,0115	0,0873	0,9851	0,9966	0,9710	0,9836
8	0,0088	0,0447	0,9915	0,9882	0,9934	0,9908
9	0,0099	0,0409	0,9914	0,9925	0,9884	0,9904
10	0,0090	0,0364	0,9915	0,9913	0,9902	0,9908

DistilBERT trained on Ling-Spam

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1-Score
1	0,2115	0,0332	0,9903	0,9815	0,9549	0,9680
2	0,0262	0,0392	0,9876	0,9811	0,9369	0,9585
3	0,0112	0,0264	0,9958	0,9909	0,9819	0,9864
4	0,0050	0,0386	0,9917	0,9646	0,9820	0,9732
5	0,0009	0,0268	0,9944	0,9820	0,9820	0,9820
6	0,0005	0,0301	0,9945	0,9820	0,9820	0,9820
7	0,0003	0,0305	0,9945	0,9820	0,9820	0,9820
8	0,0002	0,0340	0,9945	0,9820	0,9820	0,9820
9	0,0002	0,0374	0,9945	0,9820	0,9820	0,9820
10	0,0001	0,0386	0,9945	0,9820	0,9820	0,9820

DistilBERT trained on SpamAssassin

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1-Score
1	0,2672	0,1040	0,9689	0,9716	0,9329	0,9518
2	0,1102	0,0891	0,9682	0,9715	0,9308	0,9507
3	0,0807	0,0852	0,9703	0,9932	0,9161	0,9531
4	0,0663	0,0721	0,9717	0,9977	0,9161	0,9552
5	0,0600	0,0663	0,9731	0,9955	0,9224	0,9576
6	0,0531	0,0624	0,9786	0,9806	0,9539	0,9670
7	0,0464	0,0717	0,9758	0,9702	0,9560	0,9630
8	0,0381	0,0782	0,9779	0,9764	0,9560	0,9661
9	0,0382	0,0748	0,9779	0,9588	0,9748	0,9667
10	0,0352	0,0631	0,9800	0,9726	0,9664	0,9695

DistilBERT on Enron-Spam

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1
1	0,0811	0,0263	0,9908	0,9929	0,9889	0,9908
2	0,0182	0,0308	0,9923	0,9940	0,9908	0,9924
3	0,0072	0,0546	0,9881	0,9981	0,9785	0,9882
4	0,0036	0,0361	0,9938	0,9962	0,9915	0,9938
5	0,0050	0,0375	0,9933	0,9960	0,9908	0,9934
6	0,0025	0,0403	0,9942	0,9918	0,9969	0,9943
7	0,0049	0,0407	0,9941	0,9922	0,9962	0,9942
8	0,0046	0,0307	0,9951	0,9946	0,9957	0,9951
9	0,0027	0,0331	0,9939	0,9899	0,9981	0,9939
10	0,0048	0,0597	0,9915	0,9842	0,9992	0,9917

DistilBERT on combined dataset

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1-Score
1	0,0967	0,0687	0,9810	0,9704	0,9888	0,9796
2	0,0297	0,0486	0,9866	0,9908	0,9802	0,9855
3	0,0170	0,0454	0,9890	0,9870	0,9891	0,9880
4	0,0141	0,0846	0,9856	0,9929	0,9759	0,9843
5	0,0121	0,0792	0,9867	0,9798	0,9915	0,9856
6	0,0094	0,0588	0,9897	0,9931	0,9845	0,9888
7	0,0098	0,0556	0,9892	0,9880	0,9886	0,9883
8	0,0079	0,1068	0,9848	0,9737	0,9938	0,9837
9	0,0087	0,0705	0,9878	0,9927	0,9808	0,9867
10	0,0075	0,0710	0,9894	0,9897	0,9874	0,9885

RoBERTa on Ling-Spam

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1-Score
1	0,0531	0,1276	0,9682	0,8382	0,9913	0,9084
2	0,1552	0,0441	0,9931	0,9911	0,9652	0,9780
3	0,0105	0,0491	0,9903	0,9909	0,9478	0,9689
4	0,0100	0,0464	0,9931	0,9911	0,9652	0,9780
5	0,0009	0,0470	0,9917	0,9823	0,9652	0,9737
6	0,0003	0,0635	0,9917	0,9739	0,9739	0,9739
7	0,0000	0,0691	0,9903	0,9737	0,9652	0,9694
8	0,0000	0,0721	0,9903	0,9737	0,9652	0,9694
9	0,0000	0,0734	0,9903	0,9737	0,9652	0,9694
10	0,0000	0,0754	0,9903	0,9737	0,9652	0,9694

RoBERTa on SpamAssassin

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1-Score
1	0,3001	0,1638	0,9517	0,9516	0,9266	0,9266
2	0,1616	0,1097	0,9655	0,9651	0,9287	0,9466
3	0,1088	0,1029	0,9745	0,9955	0,9266	0,9598
4	0,0985	0,1243	0,9703	1	0,9098	0,9538
5	0,0810	0,0556	0,9786	0,9912	0,9434	0,9667
6	0,0804	0,0637	0,9800	0,9978	0,9413	0,9687
7	0,0617	0,0728	0,9793	0,9978	0,9392	0,9676
8	0,0680	0,0539	0,9765	0,9784	0,9497	0,9638
9	0,0520	0,0659	0,9807	0,9581	0,9581	0,9703
10	0,0588	0,0576	0,9786	0,9934	0,9413	0,9666

RoBERTa on Enron-Spam

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1-Score
1	0,0932	0,0312	0,9923	0,9909	0,9941	0,9925
2	0,0254	0,0377	0,9928	0,9946	0,9913	0,9929
3	0,0125	0,0480	0,9921	0,9895	0,9951	0,9923
4	0,0118	0,0539	0,9908	0,9945	0,9873	0,9909
5	0,0046	0,0516	0,9918	0,9929	0,9911	0,9920
6	0,0091	0,0532	0,9927	0,9939	0,9918	0,9928
7	0,0093	0,0534	0,9935	0,9934	0,9939	0,9936
8	0,0045	0,0510	0,9928	0,9960	0,9899	0,9929
9	0,0054	0,0571	0,9928	0,9950	0,9908	0,9929
10	0,0056	0,0655	0,9924	0,9918	0,9934	0,9926

RoBERTa on combined dataset

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1-Score
1	0,1052	0,0461	0,9878	0,9896	0,9834	0,9865
2	0,0412	0,0397	0,9888	0,9915	0,9836	0,9876
3	0,0259	0,0660	0,9879	0,9847	0,9887	0,9867
4	0,0232	0,0378	0,9911	0,9922	0,9882	0,9902
5	0,0188	0,1123	0,9815	0,9965	0,9626	0,9793
6	0,0177	0,0559	0,9886	0,9957	0,9790	0,9873
7	0,0152	0,0485	0,9916	0,9951	0,9863	0,9907
8	0,0160	0,0521	0,9908	0,9907	0,9889	0,9898
9	0,0162	0,0553	0,9904	0,9938	0,9849	0,9893
10	0,0145	0,0505	0,9912	0,9916	0,9891	0,9903

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit mit dem Titel:

BERT-based Spam Email Classification:
Exploring Model Effectiveness Across Different Datasets

selbständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich wörtlich aus der Literatur oder aus anderen Quellen wie z. B. Internetseiten übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

Datum	Unterschrift