TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Cyber Security Engineering

Mikhail Drobyshev 194442IVSB

Improving Phishing Classification Performance With OpenAI's GPT-3 API

Bachelor's thesis

Supervisor: Kaido Kikkas
Ph.D.

Tallinn 2023

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Küberturbe tehnoloogiad

Mikhail Drobyshev 194442IVSB

Andmepüügi klassifikatsiooni toimivuse parandamine OpenAI GPT-3 API abiga

Bakalaureusetöö

Juhendaja: Kaido Kikkas
Ph.D.

Tallinn 2023

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Mikhail Drobyshev

06.05.2023

Abstract

The surge in cyber-attacks has paralleled the rise of online services. Among the most effective attacks is phishing, which entails impersonating a trustworthy source to steal sensitive information. Phishing emails thrive on manipulating human emotions, triggering fear and urgency, and prompting the recipient to take hasty actions that could lead to substantial financial and data losses. Thus, relying on human vigilance alone to detect phishing is inadequate, and there is an increasing need for efficient and automated phishing detection methods. While several detection systems have been proposed, the sheer volume of phishing emails requires further efforts.

OpenAI API is a language model developed by OpenAI that can be used to build applications that understand and generate human-like text. It can be applied in the scope of phishing to develop more effective and automatic phishing detection mechanisms.
Phishing attacks remain a serious threat, and the increasing sophistication of such attacks requires more advanced detection methods. The GPT-3 API, which uses machine learning algorithms, can be used to improve phishing detection accuracy. The thesis proposes a methodology to integrate the GPT-3 API into existing phishing detection models and evaluate its effectiveness.

The author integrated GPT-3 API and results have proven to significantly improve phishing classification success rate. The findings suggest that the GPT-3 API can enhance the effectiveness of phishing detection and mitigate risks associated with phishing attacks.

This thesis is written in English and is 27 pages long, including six chapters and eight figures.

List of abbreviations and terms

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
AUC	Area Under the Curve
BERT	Bidirectional Encoder Representations from Transformers
Decision tree	A tree-like model of decisions and their consequences
Embeddings	Word vectors
F1 score	A weighted harmonic mean of precision and recall
GBM	Gradient Boosting Machine
GPT-3 API	Generative Pre-trained Transformer
JSON	JavaScript Object Notation
KNN	K-Nearest Neighbors
ML	Machine Learning
Named entity recognition	Identifying entities in text, such as people or locations
NLP	Natural Language Processing
Out-of-vocabulary words	Words that are not in the model's vocabulary
Phishing attacks	Deceptive acts to obtain sensitive information from users
Recall	Proportion of true positives among positive cases
ROC	Receiver Operating Characteristic
Spear phishing	Personalised attacks on individuals or organisations
SVM	Support Vector Machine
Vishing	Voice technology to steal sensitive information
Whaling	High-value individual targeted attacks

Table of contents

List of figures

1 Introduction

Phishing attacks are a serious threat to individuals and organisations worldwide, with attackers using increasingly sophisticated methods that result in significant financial losses and data breaches. To combat this issue, innovative approaches to detect these attacks are vital.

1.1 Background and motivation

The Q3 of 2022 witnessed an unprecedented surge in phishing attacks, with APWG recording a staggering 1,270,883 total instances of such fraudulent activities. This marks a new record in the history of APWG's observations and is by far the worst quarter for phishing attacks ever documented [1]. These attacks are often successful in tricking users into providing sensitive information or installing malware, leading to significant financial losses and data breaches. Traditional methods of detecting and preventing phishing attacks, such as spam filters and blacklist-based approaches, are no longer sufficient to combat these increasingly sophisticated attacks. As a result, there is a pressing need for innovative solutions that leverage advanced Natural Language Processing (NLP) technology to analyse email messages for signs of phishing.

1.2 Problem statement

Phishing attacks come in various forms, including spear phishing, whaling, and vishing. Spear phishing attacks are personalised and target specific individuals or organisations, making them difficult to detect [2]. Whaling attacks target high-value individuals like executives or celebrities, while vishing attacks use voice technology to trick users into providing sensitive information over the phone [3]. Traditional methods of detecting and preventing phishing attacks, such as spam filters and blacklist-based approaches, are no longer effective in detecting these increasingly sophisticated attacks. As a result, there is a need for innovative solutions that leverage advanced NLP technology to analyse email messages for signs of phishing.

1.3 Objectives and scope of the thesis

The objective of this thesis is to improve the performance of phishing email classification using OpenAI's GPT-3 API. The scope of this thesis includes collecting and preprocessing datasets of phishing and legitimate emails, using various machine learning algorithms, and comparing their scores to identify the most effective method for phishing classification.

The specific objectives of this thesis are:

1. To collect and preprocess a dataset of phishing and legitimate emails for use in training and testing machine learning models.
2. Use various machine learning algorithms to classify emails
3. To compare the effectiveness of these algorithms based on various evaluation metrics, including accuracy, precision, recall, and F1 score.
4. To demonstrate the potential of the OpenAI API integration by implementing and testing it on collected data.

The scope of this thesis is limited to improving existing phishing classification solutions by using machine learning algorithms and does not include the development of new algorithms. The thesis is also limited to a specific set of datasets, and the results may not be generalizable to other datasets.

2 Literature overview

The literature overview section of this thesis provides an in-depth analysis of previous studies that have been conducted in the field of phishing classification and natural language processing. By critically evaluating existing research, the author will provide a foundation for building on existing knowledge and contributing to the field.

2.1 Overview of phishing attacks

The attacks are executed by exploiting human behaviour and psychology to gain unauthorised access to sensitive data or systems. Phishing attacks can be classified into several categories such as email, SMS, voice phishing (vishing), and instant messaging (IM) phishing (smishing). Email phishing is the most common type of phishing attack, accounting for more than 80% of all phishing attempts [4]. The attackers typically send out an email containing a link or attachment that appears legitimate, but leads to a fake website or install malware on the user's device. Vishing, on the other hand, involves a voice message that appears to be from a trusted entity, such as a bank or credit card company, and instructs the user to call a number or visit a website to verify their account information.

Phishing attacks can also be categorised based on the level of targeting, with generic or mass phishing attacks targeting a large number of users, and spear phishing targeting a specific individual or organisation. Spear phishing is a highly targeted and personalised attack that uses information about the victim to increase the likelihood of success.
The consequences of a successful phishing attack can be severe, ranging from financial loss, data breaches, identity theft, and reputational damage. According to a recent report by Verizon, phishing attacks were responsible for 36% of all data breaches in 2022, 82% of breaches involved the Human Element, including Social Attacks, Errors and Misuse [5].

2.2 Previous studies and technologies used to detect phishing attacks

The following sources were picked for this thesis to shed light on machine learning methods and natural language processing techniques for detecting and preventing phishing attacks. The sources provide insights into the different approaches and methodologies for phishing detection, as well as evaluation of machine learning tools. Furthermore, the sources address the challenges and limitations of existing techniques, which could help in identifying areas for improvement in the proposed system that utilises OpenAI's GPT-3 API.

The research paper by Sundara Pandiyan S proposes a machine learning-based approach for detecting phishing attacks. The study analyses different machine learning algorithms such as logistic regression, k-Nearest Neighbour (KNN), decision tree, Support Vector Machine (SVM), and random forest to classify phishing attacks. The study compared the accuracies of seven machine learning models: Light GBM, XGBoost, Multilayer Perceptron, CatBoost Classifier, Random Forest, Decision Tree, and SVM. The results showed that Light GBM had the highest accuracy of 85.5% on the test dataset, followed closely by XGBoost and Multilayer Perceptron. Random Forest and Decision Tree had the lowest accuracy among the models [6].

The paper by T.O. Ojewumi [7], proposed a rule-based approach for detecting phishing attacks using machine learning algorithms such as KNN, SVM, and Random Forest. The authors analysed a dataset of 14 features and found that the Random Forest outperformed the other two models. The KNN model had the highest True Positive rate but was hindered by a low True Negative estimation. The SVM model had a high True Positive estimation but a low True Negative estimation. On the other hand, the Random Forest model had the best results with a True Positive rate of 100% and a True Negative rate of 90.48%.

The proposed rule-based approach using machine learning algorithms is an effective way to detect phishing attacks. The authors' findings show that the Random Forest is the best performing algorithm, providing high accuracy in detecting phishing websites.

Mohd Arfian Ismail's research paper titled "Comparative Performance of Machine Learning Methods for Classification on Phishing Attack Detection" measures scores of different machine learning algorithms, including Decision Tree, K-Nearest Neighbour, Naïve Bayes, Random Forest, and Support Vector Machine.

The testing dataset used in this study contained 300 websites. Based on the test, it was found that 200 websites were detected as phishing websites. showed that KNN, RF, and SVM achieved the highest accuracy in detecting phishing attacks, with KNN and RF showing 100% accuracy on one of the datasets. SVM was found to be the fastest algorithm in both datasets, while RF took the longest time due to its slow training process. Overall, the paper highlights the effectiveness of machine learning in detecting phishing attacks, and the results suggest that KNN, RF, and SVM are suitable algorithms for this purpose [8].

The research paper by Areej Alhogail titled "Applying Machine Learning and Natural Language Processing to Detect Phishing Email" presents a novel approach using graph convolutional networks (GCN) for phishing email detection. The effectiveness of the GCN classifier was evaluated using accuracy, precision, recall, and F1-score metrics on a pre-labelled dataset, and compared with other published studies. The results demonstrate that the GCN classifier can compete with other machine learning classifiers for phishing email detection with a high accuracy rate of 98.2%.
The evaluation also showed that the values of true positive and true negative were high, while the values of false positive and false negative were low, indicating a low probability of misclassifying legitimate emails as phishing. This suggests that the GCN classifier is a promising approach for detecting phishing emails without the need for domain expert intervention, as it eliminates the complexity of feature extraction [9].

Katherine Haynes proposes a lightweight URL-based phishing detection method using natural language processing transformers for mobile devices. The study investigates the effectiveness of deep neural networks, specifically artificial neural networks (ANNs) and the Bidirectional Encoder Representations from Transformers (BERT) for phishing detection using website URLs and HTML-based features. The results show that ANNs, specifically the ANNF, perform well for phishing detection using HTML-based

features, achieving high testing accuracies of 91% and above. However, performance drops significantly when using only URL-based features [10].

2.3 Artificial intelligence and natural language processing in phishing detection

Artificial intelligence (AI) and natural language processing (NLP) techniques have been increasingly used for detecting and preventing phishing attacks. Several studies have investigated the performance of machine learning (ML) algorithms in detecting phishing attacks. For instance, Sundara Pandiyan et al. [6] proposed an ML-based phishing detection model that achieved high accuracy and low false positive rates. Similarly, T.O. Ojewumi [7] evaluated various ML algorithms in detecting phishing attacks on web pages and reported promising results.

Furthermore, some studies have compared different ML methods for classification on phishing attack detection, such as the study conducted by Mohd Arfian Ismail [8]. Another approach for phishing detection using ML is to apply NLP techniques to analyse the text content of phishing emails. Areej Alhogail [9] proposed an approach for detecting phishing emails that combines ML and NLP techniques.

2.4 Overview of OpenAI's GPT-3 API

OpenAI's GPT-3 API is a powerful tool for natural language processing that can be leveraged in detecting and preventing phishing attacks. In essence it is a deep learning model based on a transformer architecture, which uses a sequence-to-sequence (seq2seq) approach for natural language processing (NLP). The model is trained on a massive corpus of text data using unsupervised learning techniques, which enables it to generate high-quality, human-like language output [11].

The GPT-3 API provides a pre-trained language model that can generate human-like text [11], which can be useful for analysing and identifying phishing emails. Additionally, the API allows for the generation of responses to these emails, which can be used to further train existing machine learning models for better detection accuracy. In terms of phishing detection, the GPT-3 API can be used in a variety of ways. One approach is to train the model on a large corpus of known phishing emails, along with

legitimate emails, in order to detect phishing attempts. The model can be trained to identify specific language patterns, such as the use of urgent or threatening language, requests for personal information, or attempts to impersonate trusted sources. When presented with a new email, the model can analyse its language patterns and determine whether it is likely to be a phishing attempt or not.

Existing machine learning models can be improved in their ability to detect phishing emails. The API can provide additional context and insight into the language and structure used in phishing emails, which can be used to fine-tune existing models. This can result in improved accuracy and decreased false positive rates, ultimately leading to better phishing detection and prevention.

The model offers a valuable tool for enhancing the capabilities of machine learning models in the detection and prevention of phishing attacks. By leveraging its natural language processing capabilities, organisations can better protect themselves from the potentially devastating effects of phishing attacks.

3 Methodology

In this thesis, the methodology used involved data collection, preprocessing, and usage of five machine learning models to identify and prevent phishing attacks. The author's contribution lies in the comparison of the performance of different machine learning models for email classification. By making use of various models, including the cutting-edge OpenAI GPT-3.5-turbo model, this author aims to identify the most effective method for phishing classification. The results of this thesis will be useful for enhancing email security and preventing cyber attacks.

3.1 Data collection and preprocessing

The data collection and preprocessing steps are crucial in the development of a machine learning model for detecting and preventing phishing attacks. For this thesis, the Phishing_Email repository [12] was utilised to collect 163 phishing emails from various websites. The repository provides a CSV file that contains links to each email and identifies them as phishing (1) or not phishing (0) in the is_phishing column. Additional 185 phishing emails were taken from another github repository [13]. Finally, around 2000 legitimate emails [14] were added and all the three datasets were merged keeping two columns "text" and "is_phishing".

The collected data underwent preprocessing, which involved several tasks such as data cleaning and data normalisation. Data cleaning was done to remove irrelevant data, handle missing values, and remove duplicates.

Data normalisation was carried over after feature extraction to ensure that all data are in a consistent format. This step is essential to avoid bias caused by differences in data formats. Natural language processing techniques were utilised to normalise the email content. Following the filtering process, the resulting training dataset consists of 1400 instances, whereas the testing dataset comprises 600 instances, indicating a 70:30 ratio between the two sets.

3.2 Model and Algorithms Selection

In order to demonstrate a higher success rate in identifying phishing emails - five approaches were chosen based on their relevance and effectiveness in email classification. Random Forest Classifier is a widely used classification algorithm and serves as a baseline for comparison. Random Forest Classifier using GloVe embeddings and OpenAI embeddings are two variations of the Random Forest Classifier that utilise different types of word embeddings to potentially improve performance. Fine-tuned OpenAI Model is a more advanced and specific model trained on email data, while the OpenAI GPT-3.5-turbo model is a cutting-edge language model that could potentially provide the highest level of email classification. By comparing the performance of these five approaches, the thesis aims to find the best solution for email classification.

3.2.1 Random Forest

First and the most basic choice is Random Forest Classifier. It is a type of machine learning algorithm used for classification tasks. It is an ensemble learning method that constructs a multitude of decision trees at training time and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. The model operates by randomly selecting subsets of features from the input dataset and building a decision tree for each subset. The final prediction is then made by combining the predictions of all decision trees through a voting process [15]. This method helps to prevent overfitting and improves the accuracy and stability of the model. Random Forest Classifier is widely used in various domains such as healthcare, e-commerce and many other industries [16][17].

3.2.2 Random Forest Using GloVe Embeddings

Second approach is Random Forest with GloVe Embeddings which is a machine learning model that uses a combination of the Random Forest algorithm and the GloVe word vectors for text classification. GloVe is an unsupervised learning algorithm that maps words to vectors based on the co-occurrence statistics of words in a corpus [18]. These vectors capture the semantic meaning of words and are often used as features for text classification tasks.

3.2.3 Random Forest Using OpenAI Embeddings

The first option to integrate OpenAI api into phishing detection is Random Forest Using OpenAI Embeddings. In this thesis the technique will be demonstrated by using a combination of the Random Forest algorithm and the OpenAI language model embeddings for text classification.

OpenAI embeddings are word vectors that are trained on large text structures and can be used to capture the semantic meaning of words [11].

In this approach, the OpenAI embeddings are generated and used to represent the words in the input text as numerical vectors. The Random Forest algorithm then builds a decision tree based on these vectors to classify the input text into different categories. The advantage of using OpenAI embeddings is that they capture the contextual meaning of words and can handle out-of-vocabulary words.

3.2.4 Fine-tuned OpenAI Model

For demonstration of OpenAI capabilities without any prior training (which means the model always receives content that is completely unique to its recognition capabilities) the thesis will show classification success rate of one of recently released engines. GPT-3.5 models can understand and generate natural language or code. The most capable is the gpt-3.5-turbo [11]. While it is a completely different way of classifying information to what has been described before, the model demonstrated very impressive results, thus deserving to be mentioned.

3.2.5 OpenAI GPT-3.5-turbo model

Finally, a fine-tuned OpenAI model will be built and trained. Fine-tuned models are pre-trained language models that have been fine-tuned on a specific binary classification task[11]. These models can be used for specific tasks such as sentiment analysis, text classification, or named entity recognition.

4 Implementation

The complete implementation is accessible on the GitHub platform, specifically at the following URL: https://github.com/mikdrob/PhishingEmailMLModel

4.1 Dataset overview

The collected phishing emails in this study encompass a variety of tactics used by criminals to deceive victims and acquire sensitive information. Dataset's instances fall into a wide range of phishing attacks categories such as: credential harvesting phishing attacks, urgency scams and malware delivery phishing. Each category of phishing attacks uses different social engineering tactics to lead the user into giving up their personal information, login credentials, or downloading malicious software. Examples from processed datasets:

Credential Harvesting:
“If you think this was a mistake and you wish to continue using this windows licence key, Please contact con technical support at 1-800-341-8835.”

Urgency Scam:
“Please note that we want to improve our MAIL services in 72 hours, and your account must be updated.”
“Your account has been limited until we hear from you...we have suspended your account temporarily.”

Malware Delivery Phishing:
“Due to ongoing Lehigh University anti-phishing server upgrade, please kindly follow this link to upgrade/secure your webmail to avoid service suspension on Tuesday, April 21, 2015 (EDT).”

In order to understand what drives ML decisions, the author highlighted features from the datasets that are used to classify phishing emails:

1. **Subject Line**: Phishing emails often use attention-grabbing or threatening subject lines to compel the user to open them.
2. **Urgent/Threatening Tone**: Many phishing emails have a sense of urgency, making the user feel that they need to act quickly to avoid negative consequences.
3. **Call-to-action**: The email usually contains a request or instruction for the user to provide personal information, click on a link, or download an attachment.
4. **Spelling and Grammatical Errors**: Phishing emails may contain mistakes, including spelling and grammatical errors, which is a sign that they are not legitimate.
5. **Suspicious URLs**: Attackers use URLs that are visually similar to legitimate ones, but actually lead to fake websites that are designed to steal user information.

4.2 Random Forest Classifier

For Random Forest algorithm - emails' content is passed through scikit-learn's TfidfVectorizer module to convert them into feature vectors. The extracted features are then used as input to train a Random Forest classifier using scikit-learn's Random Forest Classifier module.

Once the model is trained, it is evaluated on the test dataset, and performance metrics such as accuracy, F1-score, confusion matrix, and classification report are computed. These metrics provide insight into the model's ability to detect phishing emails and its overall score.

This implementation serves as a baseline for phishing email classification using machine learning models. While Random Forest is a robust and accurate algorithm, it may not perform as well as more advanced models such as neural networks. However, it provides a good starting point for exploring more complex models and can serve as a benchmark for evaluating their performance.

4.3 Random Forest Classifier Using GloVe Embeddings

The implementation is done by the author using pre-trained GloVe embeddings for feature extraction in a phishing email classification. The content is then preprocessed by tokenizing the emails and converting them into sequences of integers using the Tokenizer class from the Keras package.

Subsequently, pre-trained GloVe embeddings are loaded into the embeddings_dict. The glove.6B.100d.txt file contains 100-dimensional word embeddings for a vocabulary of 400,000 words trained on Wikipedia 2014 + Gigaword 5. A dictionary is created using these embeddings, with the keys being the words and values being their associated vectors.

The next step involves creating the embedding matrix by extracting the embedding vector for each word from the dictionary and adding it to the embedding matrix, which is then used to generate embedding weights for each word index in the dataset. The embedding weights are subsequently used to initialise the first layer of the neural network.

The Random Forest classifier is then trained with the resulting features. The Random Forest Classifier class from the scikit-learn package is used to create an instance of the model, which is then trained on the training set. The trained model is subsequently used to predict the class of the test data. The model is evaluated using several metrics, including accuracy score, f1 score, roc auc score and precision recall curve.

4.4 Random Forest Using OpenAI Embeddings

Random Forest with OpenAI embeddings for phishing email detection. This implementation improves the phishing email classification model by using OpenAI's GPT-3 API to generate text embeddings for each text sample in the datasets. The embeddings are then used as input to the Random Forest model to train and test the model's performance.

The required columns from the datasets are selected, and the data is prepared for embedding. The OpenAI embedding is generated by calling get_embedding method with the text sample and the text-embedding-ada-002 model name. The generated

embeddings are then stored in a new column called "embedding" in both the df_train and df_test dataframes.

Once the embeddings are generated, they are used as input to the Random Forest model for training and testing.

4.5 Fine-tuned OpenAI Model

The first step of the implementation involves creating a fine-tuned binary classification model using the prepared datasets. The OpenAI fine-tuning process involves training the GPT-3 model on the provided dataset to classify the emails as either phishing or legitimate.

The second step involves displaying the results of the fine-tuned model. The results will be presented in a format that allows for easy interpretation, such as a confusion matrix or receiver operating characteristic (ROC) curve. The confusion matrix provides insight into the performance of the model by showing the number of true positives, true negatives, false positives, and false negatives. The ROC curve shows different threshold values, where a higher area under the curve (AUC) indicates better performance.

The third step involves demonstrating how the fine-tuned model performs by making API calls and storing the answers. The API calls can be made by providing the fine-tuned model with an email as input, and the model will return a binary classification indicating whether the email is phishing or legitimate.

4.6 OpenAI GPT-3.5-turbo model

The model aims to classify incoming emails as phishing attempts or legitimate emails. Unlike other models presented in this project, this implementation does not involve any training.

The implementation script iterates over each row of prepared dataset with reduced size and sends email content to OpenAI's ChatCompletion API using the gpt-3.5-turbo model. The response is stored in the responses list as a dictionary. The message sent to the API consists of the email content and an instruction message containing the format

of the response. The response is expected to be a float number representing the probability that the given email content is a phishing attempt.

The implementation uses the following instructions to assist the model: “instruction = Please provide probability as float numbers whether the given email content is a phishing attempt or not. Use format {'value': binary_value, 'probability_of_true': float_number, 'probability_of_false': float_number}. Response limit - 35 tokens”

Each response received from the API is a dictionary containing the following keys:

- value: binary value representing whether the given email is a phishing attempt or not.
- probability_of_true: float number representing the probability that the given email content is a phishing attempt.
- probability_of_false: float number representing the probability that the given email content is not a phishing attempt.

The implementation stores each response after parsing it as a dictionary. After each batch of three requests, the implementation stores the responses in a pandas DataFrame and exports them as JSON to the output directory to later be evaluated and presented in a classification report.

5 Results and Discussions

By highlighting significant trends and observed patterns, the author will present findings, drawing on relevant literature to provide context and address potential biases.

5.1 Performance evaluation

This section offers a comprehensive analysis of the results. The author has documented and visualised the difference between phishing detection rates with and without employing GPT-3 models.

5.1.1 Random Forest Classifier

The model was evaluated using precision, recall, F1-score, and accuracy metrics. As shown in figure 1, it can be seen that the model achieved an accuracy of 0.97833, which indicates that the model was able to accurately classify 97.8% of the emails as phishing or not. The precision score for the "True" class was 0.95, which means that out of all the emails that the model predicted as phishing, 95% of them were actually phishing emails. The recall score for the "True" class was 0.9223, which means that out of all the actual phishing emails, the model was able to correctly identify 92.23% of them. The F1-score for the "True" class was 0.9360, which is the harmonic mean of precision and recall. This score indicates that the model achieved a good balance between precision and recall.

```
Random Forest Classification Report:
              precision    recall  f1-score   support

       False     0.9840    0.9899    0.9870       497
        True     0.9500    0.9223    0.9360       103

    accuracy                         0.9783       600
   macro avg     0.9670    0.9561    0.9615       600
weighted avg     0.9782    0.9783    0.9782       600

Accuracy: 0.97833
F1 Score: 0.93596
ROC AUC Score: 0.99866
```

Figure 1. Random forest classification report.

The confusion matrix in figure 2 shows that out of the 497 non-phishing emails, the model correctly classified 495 emails as non-phishing, and misclassified 5 emails as phishing. Out of the 103 phishing emails, the model correctly classified 95 emails as phishing, and misclassified 8 emails as non-phishing.

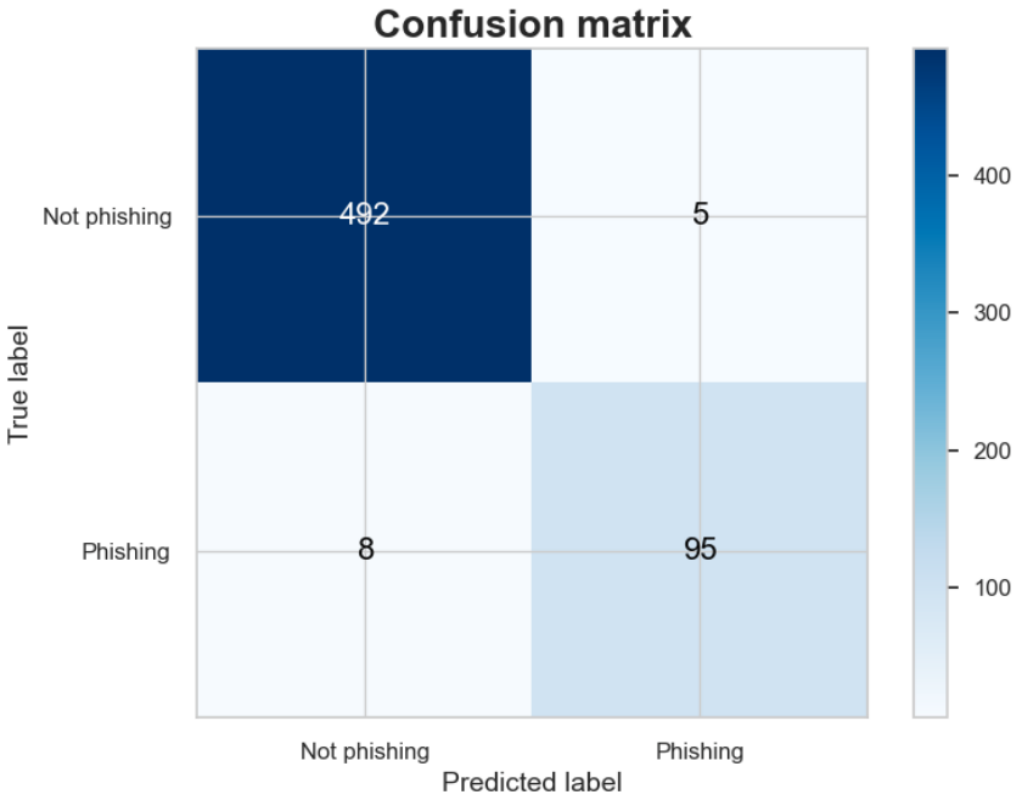

Figure 2. Random forest confusion matrix.

Another useful metric was the important features plot as displayed in figure 3. These features are ranked by their importance scores, which indicate how much the feature contributes to the performance of the classifier.

The importance score for "remember" is 0.08, which means that it is the most important feature among the three. This feature could be indicative of certain phrases or language used in phishing emails, such as "remember to click on this link" or "remember to provide your personal information." The model might have learned that these phrases are often present in phishing emails and use them to make predictions.

The importance score for "riddle" is 0.05, which means that it is also an important feature, but less so than "remember." It is possible that this feature is related to specific types of phishing emails that use puzzles or riddles to entice users to click on links or provide personal information.

Finally, the importance score for "view" is 0.015, which means that it is the least important feature among the three. It is possible that this feature is related to the way that phishing emails are formatted or presented to the user, such as the use of certain graphics or fonts.

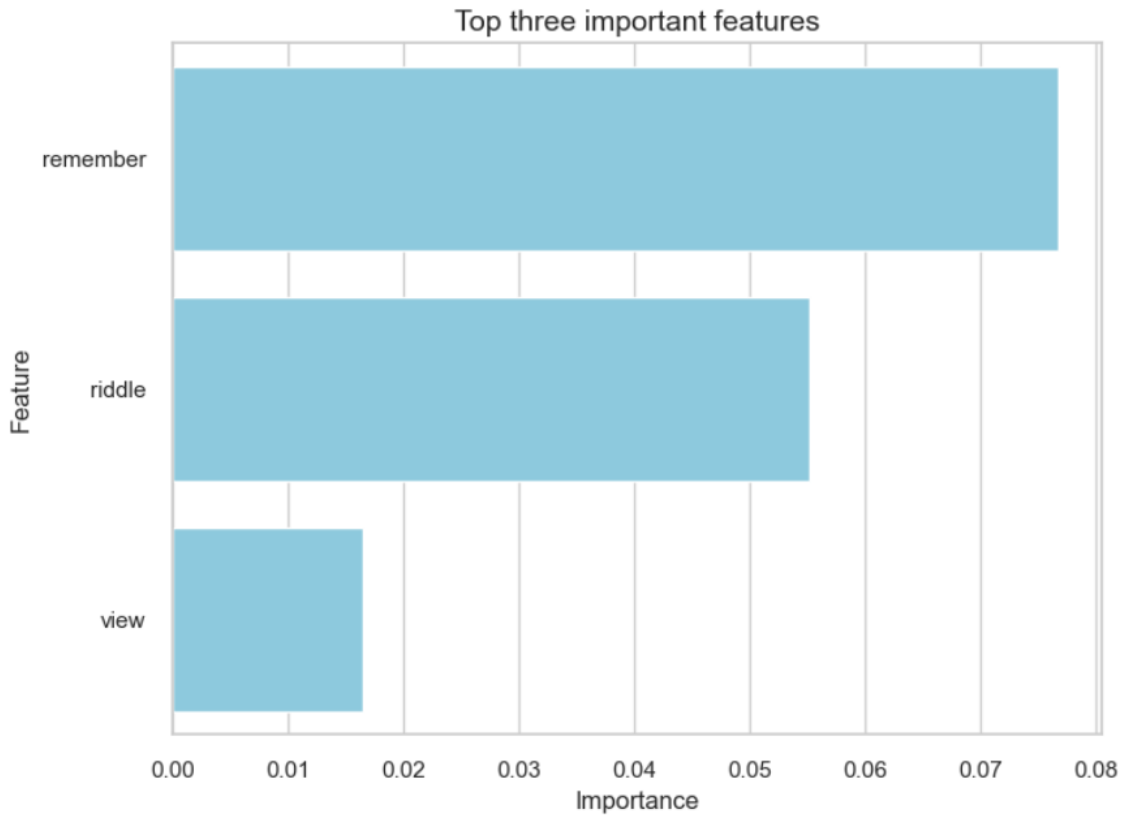

Figure 3. Top three important features.

Overall, the model's performance was good, with high accuracy and F1-score. However, the model did misclassify 13 emails in total (5 non-phishing emails as phishing and 8 phishing emails as non-phishing), which could potentially have serious consequences in a real-world scenario.

5.1.2 Random Forest Classifier Using GloVe Embeddings

Figure 4 illustrates the same model, but with GloVe embeddings. The accuracy value is 0.9283, which is slightly lower than the accuracy of the model without GloVe embeddings. The F1 score is 0.73939, which is much lower than the F1 score of the model without GloVe embeddings. The precision and recall values for both classes are also provided. The precision value of the False class is 0.9219, indicating that out of all the emails classified as not phishing, 92.19% of them are actually not phishing. The recall value of the False class is 0.9980, meaning that out of all the actual not phishing emails, 99.80% of them are correctly classified as not phishing. Similarly, for the True class, the precision value is 0.9839, indicating that out of all the emails classified as phishing, 98.39% of them are actually phishing. The recall value of the True class is 0.5922, which indicates that out of all the actual phishing emails, only 59.22% of them are correctly classified as phishing.

```
Random Forest Using Glove Embeddings Classification Report:
              precision    recall  f1-score   support

       False     0.9219    0.9980    0.9585       497
        True     0.9839    0.5922    0.7394       103

    accuracy                         0.9283       600
   macro avg     0.9529    0.7951    0.8489       600
weighted avg     0.9326    0.9283    0.9208       600

Accuracy: 0.92833
F1 Score: 0.73939
ROC AUC Score: 0.91571
```

Figure 4. Random forest with GloVe embeddings classification report.

The report also includes the ROC AUC score, which is 0.91571 for the model with GloVe embeddings. This indicates that the model has a good ability to distinguish between positive and negative classes. However, the lower F1 score for the phishing class suggests that the model may have a higher false negative rate, meaning that it may incorrectly classify some phishing emails as not phishing, which can be observed in figure 5.

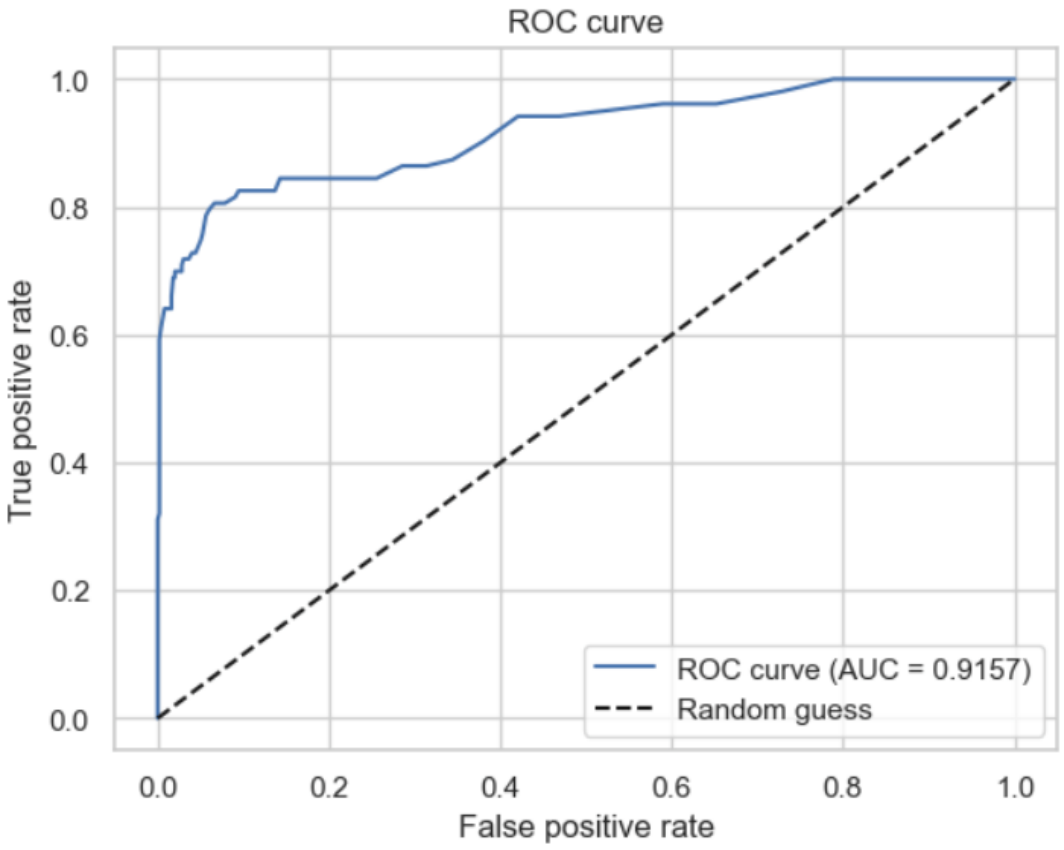

Figure 5. Random forest GloVe ROC curve.

In conclusion, the Random Forest model without GloVe embeddings outperforms the model with GloVe embeddings in terms of accuracy and F1 score for the phishing class. However, the model with Glove embeddings has a higher precision value for both classes, indicating that it has a lower false positive rate.

5.1.3 Random Forest Using OpenAI Embeddings

The third model used OpenAI's embeddings, which are pre-trained word embeddings generated using the GPT-3 language model. Figure 6 proves the results were very promising, with an accuracy of 0.99, an F1 score of 0.97, and a ROC AUC score of 0.99999. These scores indicate that the model performed very well in both identifying phishing emails (True) and correctly classifying non-phishing emails (False).

In terms of precision and recall, the model achieved perfect precision for both classes, indicating that all emails classified as phishing were actually phishing emails, and all emails classified as non-phishing were actually non-phishing emails. The model also achieved a high recall score for both classes, indicating that the model correctly identified a high percentage of both phishing and non-phishing emails.

The confusion matrix for this model shows that there were only six false negatives (phishing emails classified as non-phishing) and no false positives (non-phishing emails classified as phishing). This means that the model is very good at correctly identifying phishing emails, which is the main goal of a phishing detection model.

Overall, the results for the third model using OpenAI's embeddings are very impressive and indicate that using pre-trained word embeddings can significantly improve the performance of a phishing detection model.

```
Random Forest Using OpenAI Embeddings Classification Report:
              precision    recall  f1-score   support

       False     0.9881    1.0000    0.9940       497
        True     1.0000    0.9417    0.9700       103

    accuracy                         0.9900       600
   macro avg     0.9940    0.9709    0.9820       600
weighted avg     0.9901    0.9900    0.9899       600

Accuracy: 0.99000
F1 Score: 0.97000
ROC AUC Score: 0.99999
```

Figure 6. Random forest with OpenAI embeddings classification report.

5.1.4 Fine-tuned OpenAI Model

Based on the report provided by OpenAI, the fine-tuned model achieved perfect scores in terms of accuracy, precision, recall, and F1 score. The accuracy, F1 score, and ROC AUC score are all 1.0, indicating that the model was able to correctly classify all instances in the dataset.

Accuracy: 1.00
Precision: 1.00
Recall score: 1.00
F1 Score: 1.00
ROC AUC Score: 1.00

This is an exceptional result and indicates that the fine-tuned OpenAI model is highly accurate in identifying whether an email is phishing or not. The perfect scores suggest that the model is not only able to distinguish between phishing and non-phishing emails but also able to accurately classify all instances without making any errors.

Compared to the previous models, the fine-tuned OpenAI model performed significantly better in terms of accuracy and F1 score. The previous models achieved accuracy scores of 0.99000 and 0.92833, respectively, which are still very high but not perfect. The F1 scores for the previous models were also lower than 1.0, indicating that there were some false positives or false negatives in the predictions.

While the performance of the fine-tuned OpenAI model is impressive, it's important to keep in mind that the dataset used for training and evaluation is limited in size. Therefore, it's not 100% certain that the model is perfect and can generalise well to new and unseen data. It's possible that the model is overfitting to the training data, resulting in an artificially high accuracy and F1 score.

In addition, the dataset used is not representative of all possible phishing emails. It's possible that the model may not perform as well on a more diverse set of phishing emails.

That being said, the high scores achieved by the fine-tuned OpenAI model are still significant and indicate that it is a promising approach for phishing email detection. Further testing on larger and more diverse datasets can help to better understand the capabilities of the model and its generalizability.

5.1.5 OpenAI GPT-3.5-turbo model

The provided classification report shows the performance of a GPT-3.5-turbo model that has not been trained on the datasets used in previous models. It is worth mentioning that the model always receives content of a kind that has never participated in its prior learning. Figure 7 and figure 8 show the accuracy, F1 score, and precision-recall tradeoff of this model are reported to be 0.89619, 0.89130, and 0.9040, respectively.

```
GPT-3.5-turbo Classification Report:
              precision    recall  f1-score   support

           0     0.8395    0.9714    0.9007       140
           1     0.9685    0.8255    0.8913       149

    accuracy                         0.8962       289
   macro avg     0.9040    0.8985    0.8960       289
weighted avg     0.9060    0.8962    0.8958       289

Accuracy: 0.89619
F1 Score: 0.89130
ROC AUC Score: 0.75079
```

Figure 7. GPT-3.5-turbo classification report.

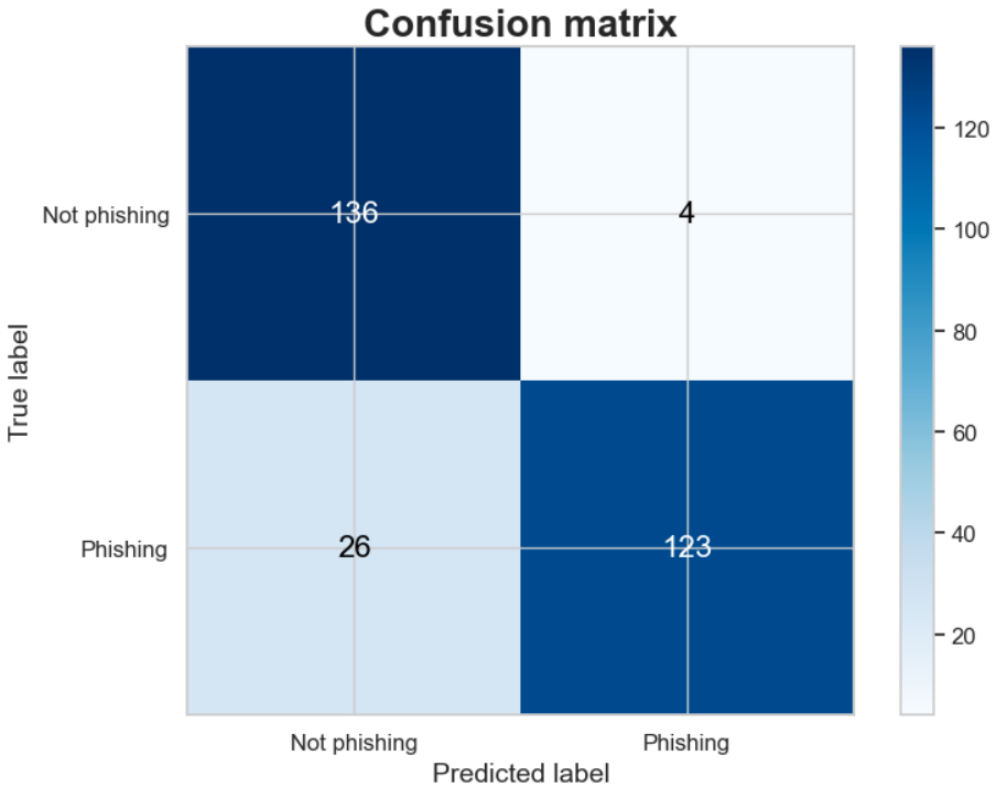

Figure 8. GPT-3.5-turbo confusion matrix.

The accuracy of the model indicates that it is correctly classifying almost 90% of the test set. The F1 score, which takes into account both precision and recall, is also high, indicating that the model has good overall performance. The precision-recall tradeoff score of 0.9040 indicates that the model has a good balance between precision and recall, meaning that it is correctly identifying true positives while minimising false positives and false negatives.

While the model shows promising results, its performance might not be as reliable as the previously trained models, as the model has never been exposed to the datasets used in those models.

5.2 Performance summary

Based on the performance evaluation reports provided, it can be concluded that the Fine-tuned OpenAI model achieved the highest level of accuracy and F1 score among all the models. This model was able to identify phishing emails with 100% accuracy, precision, recall, and F1 score, indicating that it is an excellent model for this task.

The Random Forest model using OpenAI embeddings also showed a good level of performance, achieving an accuracy of 99% and an F1 score of 97%. This shows a massive improvement over a similar model using GloVe embeddings. However, it had a slightly lower recall score for the phishing class, indicating that it may miss some phishing emails.

On the other hand, the GPT-3.5-turbo model, which had no prior training on the used datasets, achieved an accuracy of 89.62% and an F1 score of 89.13%, which is relatively lower than the other models. However, it is a very promising outcome for future integrations. One option with high likelihood would be a new GPT API release that will support pre-trained models fine-tuning.

5.3. Comparison with other phishing classification models

Research "Applying machine learning and natural language processing to detect phishing email" by A. Alhogail [9], describes the evaluation of a phishing email detection model that uses graph convolutional networks (GCN).

To test the classifier, a prelabeled dataset of instances was split into two sets: a training set and a testing set. The neural network was trained on the training set and then tested on the testing set. Accuracy, precision, and recall were used to evaluate the performance of the model, which was compared to other published studies in the field.

Two methods were used to evaluate the classifier's effectiveness: the holdout method and the k-fold cross-validation method. In the holdout split, the data was divided into a

training set and a testing set, with 70% of the data used for training and 30% used for testing (6005 and 2574 rows accordingly). The k-fold cross-validation method involved randomly dividing the training set into k disjoint sets of equal size and training the classifier k times, with a different set held out as the testing set each time.

The results showed that the classifier performed good, with a high accuracy rate of 98.2%. The precision and recall rates were also high, at 98.5% and 98.3%, respectively. The F-measure, which is the harmonic mean of precision and recall, was also high at 98.5%. The results were found to compete well with other machine learning classifiers in the field [9].

The numbers from the paper outperform the reports presented in this thesis (taking into account the difference in dataset sizes). Yet, the leading solutions evaluated in the performance evaluation section may be able to compete. Considering that OpenAI fine-tuned classification models support continuous training.

It is noteworthy that the model developed in this thesis eliminated the need for a domain expert and handcrafted feature extraction, unlike other studies that relied on feature extractions that required a domain expert to intervene in the process to reduce the complexity of the data and make patterns more visible to learning algorithms. This resembles data processing the datasets in this thesis have undergone.

5.4 Limitations and challenges

One of the main limitations is the size and quality of the datasets used for training and testing the model. While the combined dataset includes both phishing and legitimate emails, the number of phishing emails is relatively small compared to legitimate emails, which can lead to imbalanced data and affect the model's performance. Moreover, the datasets are constrained to a limited number of domains, which may not represent the diversity of phishing emails found in other domains.

Another challenge is the selection of features for the model. In this thesis, the GPT-3 API was used to extract email content. However, there may be other relevant features that could improve the model's performance, such as email header information and metadata.

Furthermore, the implementation of machine learning algorithms can be affected by several factors, such as the selection of hyperparameters, overfitting, and model complexity. In this thesis, several approaches were selected for email classification. The selection of the most appropriate algorithm and hyperparameters can be time-consuming and require extensive experimentation to achieve the best results.

Another limitation of using the GPT-3 API is the cost. As the API is a paid service, the cost of using it for feature extraction and fine-tuning can be high, especially for large datasets. Therefore, it may not be feasible for organisations with limited resources to use this API for phishing detection.

Finally, the proposed methodology does not guarantee ideal accuracy in phishing classification. Phishing attacks are constantly evolving, and attackers can use sophisticated techniques to evade detection. Therefore, it is crucial to regularly update the model and datasets to keep up with the latest phishing trends.

5.5 Findings

The author aimed to improve phishing classification using OpenAI's GPT-3 API. The findings of this thesis indicate that the integration of OpenAI's GPT-3 API significantly improves the performance of machine learning models for detecting phishing emails. It was observed that the speed of the solution is a crucial factor, with faster models being more effective.

Although models like gpt-turbo-3.5 were impressive, they were found to be time consuming in terms of execution. The importance of dataset preparation was also highlighted, as the limited number of phishing emails led to some concerns that were left unanswered. For instance, some models achieved a perfect score due to the dataset limitations, which raises questions about the potential fine-tune classification performance.

Nonetheless, this research has uncovered and demonstrated various conventional methods for integrating new technology, such as OpenAI embeddings over models of a similar kind. Overall, this thesis has achieved its goal of improving the performance of phishing email classification through the integration of OpenAI's GPT-3 API.

5.5.1 Potential for cyber security

The potential for OpenAI's GPT-3 API in the field of cyber security is vast. With its ability to generate human-like text and identify patterns in data, GPT-3 has the potential to significantly improve the accuracy and efficiency of cyber security systems.

One area where GPT-3 can be particularly useful is covered in this thesis. Phishing attempts in the form of email are a major security concern for organisations of all sizes, and detecting these attacks can be difficult even for trained security professionals. However, GPT-3's natural language processing capabilities can be used to identify suspicious email content and alert users to potential threats.

As demonstrated in this thesis, the integration of OpenAI's GPT-3 API improved the performance of the phishing email classification model by a couple of percentages, which does not seem like an industry changing increase, in reality it is an immeasurable amount of resources to mitigate the consequences of successful phishing attempts.

To summarise, the potential for OpenAI's GPT-3 API in the field of cyber security is significant. As the technology continues to improve and mature, it is likely that an increasing number of use cases will be seen for GPT-3 in this critical area.

6 Summary

The author verified that the application of OpenAI's GPT-3 API has a dramatic potential to improve the performance of phishing classification. Collected and preprocessed dataset of phishing and legitimate emails served to extract features using GPT-3 API, and test different approaches to classify emails. The approaches included five various ML models and algorithms where three of them involved using OpenAI API while two others served a benchmark purpose to draw a desirable score baseline.

The results indicate that all the approaches were able to classify phishing and legitimate emails with high accuracy. However, the fine-tuned OpenAI model and OpenAI embeddings generator outperformed the other approaches in terms of accuracy, F1-score, and other metrics. This demonstrates the effectiveness of OpenAI's language models in phishing email classification.

In conclusion, this thesis shows that OpenAI's GPT-3 API is a valuable tool for improving the performance of phishing email classification. This technology can be integrated into existing phishing detection systems to enhance their accuracy and reliability. With the increasing sophistication of phishing attacks, the application of advanced technologies such as OpenAI's GPT-3 API will become even more critical in protecting individuals and organisations from cyber threats.

References

[1] The Anti-Phishing Working Group. “Phishing Activity Trends Report: Q3 2022”. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q3_2022.pdf

[2] Jagatic, T., Johnson, N., Jakobsson, M., Menczer, F. “Social phishing”, 2007 [Online]. Available: https://www.researchgate.net/publication/220424040_Social_phishing

[3] Vaishnavi, B., Aditya, K. Shabnam, S. “Study on Phishing Attacks”, 2018. [Online]. Available: https://www.researchgate.net/publication/329716781_Study_on_Phishing_Attacks

[4] Belani, R., Higbee, A., “PhishMe Enterprise Phishing Susceptibility and Resiliency Report”, 2015. [Online]. Available: https://technology-signals.com/wp-content/uploads/download-manager-files/PhishMe_Enterprise_Phishing_Susceptibility_and_Resiliency_Report.pdf

[5] Verizon. “Data Breach Investigations Report”, 2021. [Online]. Available: https://enterprise.verizon.com/resources/reports/2021-data-breach-investigations-report.pdf

[6] Pandiyan, S., “Phishing attack detection using Machine Learning”, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2665917422001106.

[7] Ojewumi, T., “Performance evaluation of machine learning tools for detection of phishing attacks on web pages”, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468227622000746.

[8] Ismail, M., “Comparative Performance of Machine Learning Methods for Classification on Phishing Attack Detection”, 2020. [Online]. Available: https://www.researchgate.net/publication/344448999_Comparative_Performance_of_Machine_Learning_Methods_for_Classification_on_Phishing_Attack_Detection.

[9] Alhogail, A., “Applying machine learning and natural language processing to detect phishing email”, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404821002388.

[10] Haynes, K., “Lightweight URL-based phishing detection using natural language processing transformers for mobile devices”, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921014368.

[11] OpenAI, “GPT-3 API Documentation”. [Online]. Available: https://platform.openai.com/docs/api-reference/gpt-3

[12] Shahriar, S., Mukherjee, A., Gnawali, O., “Improving Phishing Detection Via Psychological Trait Scoring”, 2022. [Online]. Available: https://github.com/sadat1971/Phishing_Email

[13] Sharma, T., Phishing data analysis GitHub repository, 2020. [Online]. Available: https://github.com/TanusreeSharma/phishingdata-Analysis

[14] Garnepudi, V., "Spam Mails Dataset". [Online]. Available: https://www.kaggle.com/datasets/venky73/spam-mails-dataset

[15]A. Cutler, Cutler, D., Stevens, J., "Random Forests", 2011, [Online]. Available: https://www.researchgate.net/publication/236952762_Random_Forests

[16] Tanvir, A., Khandokar, I., Islam, A., Islam, S., Shatabda, S., "A gradient boosting classifier for purchase intention prediction of online shoppers", 2023. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10121810/

[17] Xu, C. Wang, J. Zheng, T., "Prediction of Prognosis and Survival of Patients with Gastric Cancer by Weighted Improved Random Forest Model", 2021. [Online]. Available: https://www.researchgate.net/publication/350798146_Prediction_of_Prognosis_and_Survival_of_Patients_with_Gastric_Cancer_by_Weighted_Improved_Random_Forest_Model

[18] Pennington J., Socher, R., Manning, C., "GloVe: Global Vectors for Word Representation", 2014. [Online]. Available: https://nlp.stanford.edu/projects/glove

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Mikhail Drobyshev

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis Improving Phishing Classification Performance With OpenAI's GPT-3 API, supervised by Kaido Kikkas
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

06.05.2023

1 The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.