ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Project Report

CENG 408
P202106

Content-based Analysis in order to Detect Sensitive Data

Batuhan TATLISERT
201811057

Nazım DÖLEKÇEKİÇ
201811026

Alper PUNAR
201711055

Beril YOKARIBAŞ
201811069

Advisor: Instructor Dr.Ayşe Nurdan SARAN

Table of Contents

Öz

Şirketler ve organizasyonlar kendi müşterilerinin ve çalışanlarının kişisel bilgilerini ve yazılı iletişimlerini dışardan veya içeriden gelebilecek her türlü güvenlik tehditine karşı saklamak ve korumak zorundadır. Güvenliği birçok şey tehdit edebilir. Örnek olarak e-mail ile gönderilmiş kişisel bilgi, belirli bir dokümanın içeriği veya fotoğrafı, banka hesabının ekran görüntüsü ve benzeri verilebilir. DLP sistemleri ile verileri analiz edebilir ve herhangi bir veri sızıntısına karşı güvenliği arttırabiliriz. DLP sistemleri hassas veya kişisel verileri ve yetkisiz ifşaya karşı korumak için kullanılır. Bu projede, DLP sistemlerinde hareket halindeki verileri istatistiksel olarak analiz etmek için kullanılabilecek teknikleri ve hassas verilerin sızıntısının nasıl engellenebileceğini araştırıyoruz.

Anahtar Kelimeler: veri sızıntısı engelleme sistemleri, DLP, n-gram, fingerprinting, makine öğrenmesi

Abstract

Companies or organizations must store and protect all personal information about their customers, employees, and all other written communications against any inner or outer security threats. There are numerous things that can threaten security. For instance, personal information sent by e-mail, information or image in a certain document, screenshots of bank account so on and so forth. We can analyze data with Data Leakage Prevention (DLP) systems and increase security against any data leakage. DLP systems are used to preserve sensitive or personal data, protect against unauthorized disclosure. In this project, we survey the techniques that can be used to statistically analyze data in motion within DLP systems and how to intercept leakage of sensitive data.

Keywords: data leakage prevention systems, DLP, n-gram, fingerprinting, machine learning

Literature Review

Introduction

This project aims to use fingerprinting with statistical analysis in Data Leakage Prevention (DLP) Systems so that the protection can be done in a more effective manner. The reason we use both of them instead of just using data fingerprinting is traditional fingerprinting can be deceived easily. Because with traditional hashes just an extra space in the text can cause the change of the whole fingerprint. We aim to use statistical analysis like N-gram-based methods to bypass this shortcoming. This document surveys the DLPSs, how can be improved and what sort of development plan should be implemented for the project.

DLP is the abbreviation of Data Leakage Prevention. before looking at what DLP is we should look into the Data Leakage (DL). DL means unauthorized or unwanted disclosure of the data. DLP systems are used to protect sensitive data leakage without consent. Protecting sensitive data from being leaked to the public has been a big problem for organizations and companies. Traditional protection systems such as firewalls, intrusion detection systems were not enough to protect it. Because generally, they have predefined rules, they are not flexible in this regard. Data can be leaked in different forms. That's where DLPS comes in.

DLPSs aim to prevent this leakage different from traditional protection measures. One of the differences is DLPSs are content-aware. Because they analyze the text not just look at its context information. "…we define DLPSs as designated analytical systems used to protect data from unauthorized disclosure at all states using remedial actions triggered by a set of rules." [1].

1. Content-Context

The difference between content-based and context-based can be understood with a popular example [2]. We think of content like a letter and context as the envelope of it. Context includes size, recipients, sender, metadata, time, or anything other than the message itself. To protect the sensitive data itself we can look into the envelope itself. We open the envelope and read its contents after we decide how to handle it. With this approach, we can protect content itself other than just protecting the envelope, context.

Looking into content and analyzing it is obviously more time consuming than contextual analysis and this is one of the bigger differences between DLPSs and traditional protection systems. To look into the content we need file cracking. In basic text emails, opening it is easy but we need to look into excel sheets, PDFs, Docx files, etc. In short, to be able to look at content we need to dig into it.

In our project, we use a content-based approach. In short, we read the text inside the mail, Word, PDF, etc. It is important to know the difference between content and context.

2. N-gram

N-gram algorithm is used to find the repetition rate in a sequential sequence. N represents the controller value of repetition we can briefly explain as number of words we want to look at. N represents the controller value of repetition and gram is the frequency of the certain repetition on the sequence. The N-gram method can be used to obtain the 1-gram, Bı-gram, Tri-gram features. N-gram-split is a new comparison method based on the n-gram in order to reduce the dimensions. Bı-gram in other words 2-gram method is that 2 pairs of words that occur together looking at before word and afterward. And just like 2-gram, the 3-gram method looks for three words at a time.

The k-skip-2-gram is the same as the 2-gram when k equals 0. For example, for a protein sequence ACDEF, the results are as follows: 2-gram = {AC, CD, DE, EF}, 0-skip-2-gram = {AC, CD, DE, EF}, 1-skip-2-gram = {AD, CE, DF}, and 2-skip-2-gram = {AE, CF}. If we consider only the 2-skip-2-gram, then the final feature is a sparse vector and is difficult to address directly.[3]

Generally, Zipfian fashion is used to distribute N-grams. Zipfian's law is a relation between rank order and frequency of occurrence: it states that when observations (e.g., words) are ranked by their frequency, the frequency of a particular observation is inversely proportional to its rank.[4] Gombel and Bosch categorized gram methods in their research as follows:

1. N-grams – A sequence of n-word tokens that are all consecutive. For example: "to be or not to be"
2. Skipgrams – A fixed-length sequence of p-word tokens and q token placeholders/wildcards with total length n (n = p + q), the placeholders constitute gaps or skips and a skip-gram can contain multiple of these. In turn, a gap can span one or more tokens. For example: "to _ or _ _ be"
3. Flexgrams – A sequence with one or more gaps of variable length, which implies the pattern by itself is of undefined length. For example: "to * or * be" Definitions may be flexible and depend on the source, the information is written here is from the author's of the research perspective. Some of them may use the term skip-gram to include what we explained as flexgram, or use another term such as "elastigram" to refer to flexgrams.[5]

2.1 K-SKIP-N-GRAMS

This approach is used to produce modified fingerprints. It is a durable method to identify the very first version of the data even if we made modifications to it. With the help of k-skip n-grams, we can eliminate unnecessary or less significant data from confidential documents. We need to apply intensive indexing in order to work with this approach. The major drawback is that it will be required extra storage due to indexing.[1]

2.2 N-gram Applications

N-gram algorithm has an import role in statistical natural language processing, spelling correction, handwriting recognition, and communication theory [11]. There are numerous samples of projects related to n-gram technology such as text compression.[6-7], text compression [8], spelling and error correction [9-10].

3. Fingerprinting

Data fingerprinting [13][1][14]The most common method used for content-based sensitive data detection is fingerprinting. Fingerprinting is basically hashing sensitive data and compares it with hashed inspected data. Although fingerprinting is a very common and effective approach, it has limitations. If sensitive data changes in the slightest way, comparing the hash values will be ineffective for detecting sensitive content. In the classic fingerprinting approach, the sensitive document is split tokens firstly. After that, preprocessing techniques, removing suffix and prefix form a word or removing stopwords(I,are,but, etc.) or more, are applied to tokens and apply n-gram model. N-grams that are obtained are hashed using Rabin, MD5, or another hash function.

There are other methods for fingerprinting that is mostly enhanced versions of the classic method.

-Nevin Heintze approaches statistically the tokens extracted from the document. This method checks the frequency of tokens in the document set. If a common token appears in several documents, the method ignores this token. This can decrease the rate of false-negative [12][13].

-Shapira at el. uses 'sorted skip-n-gram' to detect modified sensitive context. This method uses both sensitive and non-sensitive tokens to achieve higher detection accuracy. However, this method uses too much memory to improve detection accuracy[1][14].

There are more methods, one that uses similarity of hash values but it has relatively low detection accuracy, one that uses bloom filter to fast similarity check but it has high false-positive rate.

Although of all these methods, detecting modified sensitive content is quite difficult. The strength of fingerprinting is it has a very low false-positive rate (close to 0). It provides protection on the customer, sensitive data while omitting other similar data used by employees. Personal credit cards for online orders can be given as an example. One of the weaknesses of fingerprinting is live connections can affect database performance. In large databases product, performance is affected negatively.

4. NLP

Natural Language Processing (NLP) makes computers understand human language. It analyzes the meaning of the words, structure of the sentences, grammatical structures, etc. After that, it uses algorithms and calculations to get meaning from them. Some examples can be given to NLP such as Google Assist, Alexa, and Siri. In short, it makes computers understand or make some sense of human languages. It uses machine learning to better understand the human language and improve itself. This process is automated without requiring writing new instructions or algorithms for it thanks to machine learning. It takes a significant role to know language morphology levels in order to work with the certain language. We can categorize the levels as morphological lexical, syntactic, semantic, and pragmatic discourse [16].

BERT (Bidirectional Encoder Representations from Transformers) is an open source machine learning framework for NLP. It helps computers to understand the meaning of language in the text using texts around it to establish context. It is based on transformers. BERT can read from both left to right and right to left at the same time. Old language models couldn't do it at the same time. This difference comes from the bidirectionality of Transformers. It has been made thanks to Google's research on Transformers.

4.1 Natural Language Processing Applications

There are so many applications of Natural Language processing it can be given several examples such as Information Retrieval, Character Recognition, Spell Checking, Machine Translation, Dialogue Systems [15]. In our project, we are going to build a text classification application by natural language processing and classify our documents with confidence scores in order to defect sensitive data.

4.2 Text Classification In Natural Language

Text classification is to determine whether each document in our possession belongs to predetermined classes. We know that every application may need different requirements for the categorization process so classification can be in several ways [18].

Text Categorization Types Single Or Multi Label Classification Category-Pivoted Or Document-Pivoted Classification Hard Categorization

1. Single Or Multi Label Classification Within in single classification method, each text in a class belongs to the same class. And that specific text can not belong to any other class. For a constant value of J the expression of (d_j, d_i) can only be true for only one of them. Single classification can also be called binary classification. The binary name refers there are only 2 classes with a single label. But if each text can belong to more than one class we can choose more than one label for the classification process. For d_j,∈D documents we can choose k number of labels ($0 \leq k \leq m$) [18].

2. Category-Pivoted and Document-Pivoted Classification In the Document-Pivoted classification method, the goal is to identify all possible labels of s_i classes associated with a particular b_i document. This method is usable if the system will gain more documents over time. And for the Category-Pivoted it is aimed to determine all b_i documents that must enter that class for each s_i class. This method is used if class sets are going to be changed or merged with each other after some period of time [18].

3. Hard Categorization Assigning a single category to each document. It is a definite indication of whether the document belongs to the class or not.

4.3 Text Classification Process

After the Data Gathering and OCR, we will get our text data and begin the machine learning process with text classification. We will analyze and clean the data from unnecessary characters or blanks using REGEX. This term will be explained later. Once we tokenized data, we will represent it as vectors in order to classify our document class and return confidence levels. Then finally we will have a learning algorithm and it will be saved[17].

General Project Flow

-In the data gathering phase first, we will find datasets by crawling. These datasets will include two types: sensitive and non-sensitive documents.

-After finding documents we will extract text from them (OCR). These documents can be any type of document such as Excel, Word, pdf, jpg, png, etc. We are extracting text from documents using Tesseract. Tesseract is integrated into Apache-Tika.

-We will preprocess these texts using the NLTK library. Preprocess includes typo correction, stem analysis, etc. Stem analysis means reducing inflected or derived words to their word stem, base, root form.

-We will first tokenize then vectorize words using n-gram or k-skip gram.

-Then we will use fingerprinting. If the results are below the threshold we will send the text vectors, labels, and fingerprinting results to the classifier, machine learning. Fingerprinting results will be combined with machine learning.

-In the last step we will save and check the results.

4.4 Vectors In Machine Learning Algorithms

Machine learning (ML) and deep learning (DL) algorithms do not accept text data without vectorized representation. In our project, we will represent text data with word2vec. Generally, the simple machine learning applications use TF-IDF and count vectorizer. But they can't associate between words. So word embedding will be needed in order to understand the meaning and relation between words. The most popular method of Word embedding is Word2vec. It transforms words into vectors and supports basic operations such as adding, subtracting, merging, and so on. Word2vec can be applied in two ways. Which are Skip Grams and Continous Bag Of Words (CBOW). Basically, CBOW starts from the context and tries to find the target word and encode it as a single vector. And Skip Grams will work as vice versa it will first take the target words as input. CBOW has a small corpus and it is faster as compared to skip gram. But skip gram has higher dimensions with a large corpus.

4.5 Regular Expressions (REGEX) And Natural Language Processing

In order to maximize the use of the NLP system before the n-gram approach, we will minimize our text data with regular expression processing [19].[20] It is a sequence of characters mainly used to find or replace patterns embedded in the text. For instance, if we have a pattern that a string should start with a B and ends with L we can eliminate other strings that it is not match with the given pattern. In our Project, we will use REGEX to eliminate spaces, special and repetitive characters.[20] There is a python built-in module called “re” for Regular Expressions and we will use its functions to have correct strings of the pattern. [6]For instance e-mail addresses can be valid only strings which are acceptable with this pattern “/[\w._%+-]+@[\w.-]+.[a-zA-Z]{2,4}/”. It matches anything within brackets. Special characters in REGEX are called “metacharacters”. The most common ones are “,^,$,.,*,+,?,[,{,(,)|” so we will omit them to get effective and minimal cost of vectors.

5. Related Works

-Email relay

It seeks TC number patterns in e-mail data attachments[21]. It opens and reads attachments such as excel, pdf, jpg, png (Picture), Word, etc. It uses Regular expression or regular expression (Regex). This project uses Docker so that it can work on every operating system (OS). In our project, we are going to use Apache-Tika, Tesseract, for extracting text from any type of data. Apache-Tika is an open source data extraction, file cracking program. In Koray's project, these types are extracted manually. Because when this project made Apache-Tika were outdated, now instead of approaching types manually we can simply extract them. In our project, we are going to use machine learning besides pattern matching (fingerprinting).

-DLP System in Images Project

This project tries to find encrypted data in pictures using pixels [22]. It uses stenography to find corruption between pixels. In this aspect it uses DLP. In our project, we look into text while this project looks at images to find sensitive data.

-Text Classification Project

Hart et al. (2011) tokenized documents and use unigrams. They used binary weighting to train the machine and use support vector machine with the linear kernel (which is a supervised machine learning algorithm that works by drawing a straight line between two classes)[23] with the linear kernel to classify data which are public enterprise data, private enterprise data, and non-enterprise data. The classifier achieves acceptable results on enterprise data but has a high false positive rate on non-enterprise data and overfitting on feature selection. To solve these problems, they used a method that they called supplement and adjust. They added a data set that is non-enterprise from Wikipedia or Reuters to training data with the purpose of preventing the classifier from overfitting the enterprise data and adjusting decision boundary(that separates the data points into specific classes)[25] for decreasing high false negative generating from supplement data. These processes reduce the false negative rate but increase false positive rate.To reduce false positive rate, they create combined classifiers with public, private, and non-enterprise documents and trained a new classifier with that combined classifier[24].

-MyDLP

It is an open source program [26]. It uses Apache-Tika, Tesseract as well for Optical Character Recognition (OCR), text extraction. Written in Java language. It parses XML Paper Specification (XPS) for metadata extraction. It uses hash, fingerprinting. It uses K-gram for token extraction and for bit hashes and data, it hashes vector. In short, it extracts text from documents (OCR), then it preprocesses data In our project, we are going to use both fingerprinting and machine learning.

-Google APIs/Python-DLP

This is not an open source project but we can look into some details that are shared like info types[27]. In InfoType detector reference table [28], it includes info type and description of it for every country. For example, for Turkey, it includes TC id number. These documentations may help with the project.

Software Requirements Specification

1.Introduction

1.1 Purpose of this Document

This system aims to develop software in order to detect sensitive data and prevent data leakage.

The purpose of this document is to present a detailed description of the *"Content-based Analysis in order to Detect-Sensitive Data Project".* This document will explain features, constraints, functional and non-functional requirements, use cases, and how the system works.

The main purpose of this project is to detect documents, and texts, with sensitive information using content-based analysis. This document is intended for both developers and stakeholders.

1.2 Scope of our Project

The scope of the project is to use this system to detect sensitive information e-mails. It will check the mail's text and attachments (Word, Excel, Pdf, Image…) and decide whether it contains sensitive information or not.

We planned the project as a desktop application, for now, later e-mail relay will be added to the system. We concentrated on the text classification part of the project.

Data that contains sensitive information will be given to the system so that it can understand what makes that data sensitive. Data can be from any topic and area but for better results, it would be better to constrain it so the success of determining whether it is sensitive or not would be higher. To achieve better results system will need large amounts of data so constraining the topics might backfire if the gathered amount of data is too low to operate the learning process. The project will use machine learning to achieve results.

2. General Description

2.1 Glossary

Term	Definition
Admin	Checks the feedback from the system via the information box to the user.
BERT	Bidirectional Encoder Representations from Transformers
CBOW	Continuous Bag Of Words
CSE	Computer Science and Engineering
DB	A database is an organized collection of structured information, or data, typically stored in a system.
DL	Deep Learning
DLP	Data Leakage Prevention
DLPS	Data Leakage Prevention System
GCHQ	Government Communications Headquarters
GCSB	Government Communications Security Bureau
ML	Machine Learning
NATO	North Atlantic Treaty Organization
NLP	Natural Language Processing
NSA	National Security Agency
OCR	Optical Character Recognition
ODNI	Office of the Director of National Intelligence
Project member	Everyone who contributed to the project.
REGEX	Regular Expression
StakeHolder	A stakeholder is any individual or group that has an interest in this system and the outcomes of actions.

TF-IDF	Term Frequency -Inverse Document Frequency
UC	A particular tag with a number refers to the use case name.
User	It specifies all users who send emails.

2.2 User Characteristics

The intended user can be part of an organization and needs to send or receive emails with attachments such as text files, word documents, pdf, and images. The system should check if there are any security threads before any submission or right after receiving the data.

Although the system should not require any educational degree to use all users who use the system should know a few basic terms like, what classified means. They should be able to understand the message shown to them.
"It contains classified information."
"It contains sensitive information." etc.

Normal users can be an employee of a company, students, teachers, etc.

Admin is the user that is going to take action according to the feedback that came from the system. Admin can be from the company that uses this system.

The normal user of the system uses it indirectly in some way. He or she sends an e-mail and this e-mail goes to the system, E-mail Relay. If the system finds an e-mail and its attachments contain sensitive information it will stop the e-mail from sending and sending a message to the admin according to the level of sensitivity in it. So the admin should have a basic understanding of the system's feedback. Admin will decide what he should do with this feedback, decisions might be related to his company regulations.

2.3 Product Perspective

A data leakage system is a standalone system once the libraries are included. It does not require any sub-systems. . In order to send an email and use the system, wifi connection and device are needed.

2.4 Overview of Functional Requirements

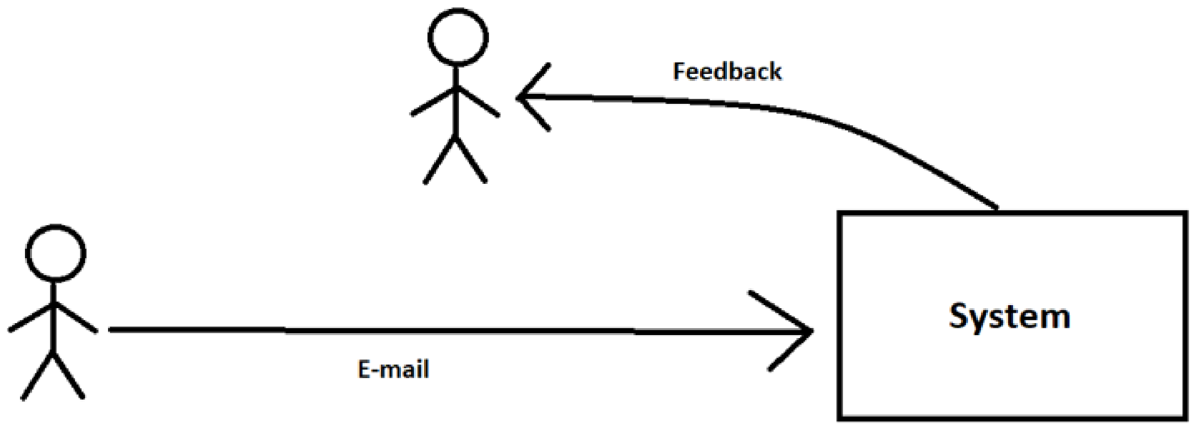


Figure 1: Basic Structure

The system should take data in many forms (Word, Excel, Pdf, Image, etc.) and use OCR to transform the information into text. After preprocessing and tokenization processes on data, data should be sent to the system. The system will use machine learning and output a result. The result will contain are these documents contain sensitive data or not and the sensitivity level of it as Highly Sensitive, Sensitive, and Safe.

In short, the user should give the system a document and the system will find the document's sensitivity level. Before that system will be trained with a large data set.

The system will give information about risk management. The confidence score will be calculated via the model.

The higher the confidence level, the higher number of matched items will be sensitive information. The system looks for the primary element that the information is suitable with a specified pattern and supporting elements such as keywords like a 'credit card'[1].

[1]If the system finds both primary and supporting elements, the confidence level is high. Just occurrence of the primary element will make the confidence level low. By utilizing confidence levels we can determine how many false positives and false negatives we receive when evaluating items for sensitive information. A high confidence level will return a few false positives but may result in a higher level with more false negatives. A low confidence level will return more false positives but few false negatives. Lastly, the medium confidence level will fall between the other two. Low counts should be used for High confidence level patterns and low confidence patterns should be used with higher counts.

The general structure can be seen in the figure below. User will send an e-mail. The E-mail will enter into the E-mail relay. Our model will be placed in the filter section. E-mail and its attachments, whole content, will be filtered via Model. According to the result, it may be directly sent to the receiver, target user, or if it is deemed as a secret e-mail it will be stopped and be sent to the Admin. Admin will check the contents and can decide to send it to the receiver or permanently stop it.

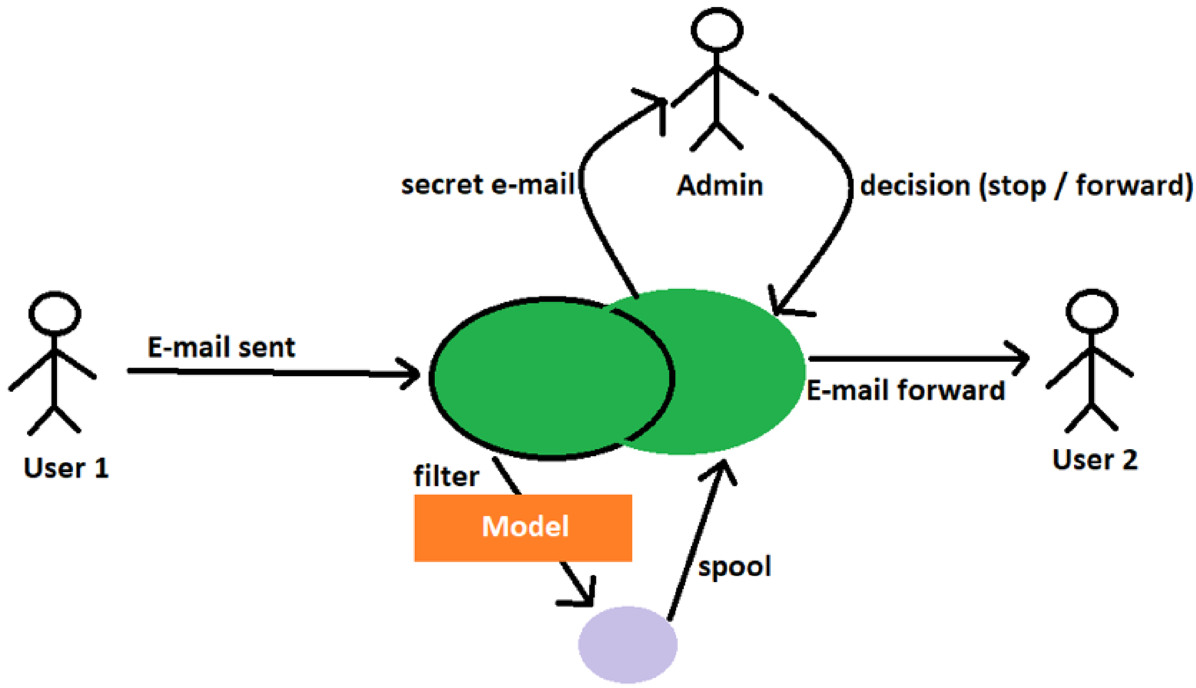


Figure 2: General Structure

2.4.1 Send E-mail

Common users should be able to send an e-mail with attachments. Feedback will be given regarding whether did mail send successfully or not.

2.4.2 Receive Feedback

The system will check this e-mail and its attachments and decide its sensitivity level. If it is too high it will directly halt the e-mail sending process and will send feedback to the admin. If it is high, again, feedback will be sent to the admin.

Admin should be able to receive feedback from the system and the problematic e-mail itself.

2.4.3 Calculate Score / Determine Sensitivity

The system will determine whether e-mail and its attachments contain sensitive information or not. It will stop the mail and return feedback to the admin according to its parameters or let go of the e-mail.

2.4.4 Train Machine

The machine will be trained with some portion of the data, %80 training, and another portion of data will be for validation, %20 validation. The exact partitioning of data can be changed through the project.

2.5 General Constraints and Assumptions

While creating our software system, we determined the features, capabilities, and interfaces to meet the needs and demands of our stakeholders. This brought us some restrictions in the system software design part. In our project, we have a few constraints and assumptions.

Users are required to have an active email account in order to use the system.

Another restrictive factor in the use of the system is the internet speed. Mail should be sent and received successfully. The amount of delay will not be covered during the performance. We will assume that the e-mail traffic is not heavy for handling and management purposes. Linux operating system and python libraries will be used via the system software development part. Next, we will assume that users have basic knowledge about e-mail operators(send, receive, attachments…)

Gathered data should be enough to perform machine learning satisfyingly. We assume gathered data is enough for getting acceptable results via training. How much data is required for this "acceptable result" and what exactly the "acceptable result" is will be added later.
It must be tested to determine how large the dataset needs to get this acceptable result. The acceptable result should be close to 0.7 at least. If it is too far from it no matter what we improve on the system side we will not be able to achieve it. Because machine learning is highly dependent on the data gathered, both the quality and quantity of the gathered data.

If the gathered data is few in numbers machine cannot learn and overfitting may happen because we will also need to test the training with a partition of the gathered data.

If the quality of the data is too bad or the topics and the information they contain too broad even with the large amounts machine might not be able to learn well enough, therefore results might be bad. In short, results can only be good as the data itself.

The data we found consists of 2210 PDF files. They are a culmination of documents from NSA, GCHQ, ODNI, CSE, GCSB, NATO, and various other agencies. Extra data might be added should it be required.

We assume the machine is already trained.

EXPENDITURES	
Computer	We use our own computers
Other Devices	We can use our mobile phones and tablets
Internet Connection	We use our own internet connection
Software	Python,TenserFlow,Apache Tika,OCR,MyDLP,ElastikSearch,Kibana,Teserract
Human Resources	Our Project team consists of 4 people
Textbook/Magazine/Support Material	N/A
Memory	Moderates on the computer
ISO	N/A

3. Specific Requirements

3.1 Interface Requirements

3.1.1 User Interface

Linux command window.

The layout will be simple and easy to read.

For the admin interface, he/she will be able to check the feedback from the system.

3.1.2 Hardware Interface

No hardware interface is needed.

3.1.3 Software Interface

The system does not need any other software interface other than the operating system.

3.1.4 Communication Interfaces

There is no communication interface.

3.2 Detailed Description of Functional Requirements

3.2.1 Send E-mail

Function Name	**Send E-mail**
Purpose/Description	Let users send an e-mail.
Inputs	E-mail itself. Text and attachments (Word, Excel, Image…)
Processing	User will write text and add attachments he/she wants to send.
Output	-Feedback message regarding whether does e-mail was sent successfully or not.

3.2.2 Receive Feedback

Function Name	**Receive Feedback**
Purpose/Description	The system will send feedback to the admin.
Inputs	None
Processing	Admin will check on the feedback.
Output	Feedback to admin. -Deny: If "confidence score" is too high -Alert: If "confidence score" is high.

"Confidence Score": Probability of containing sensitive information.

3.2.3 Calculate Score

Function Name	**Calculate Score**
Purpose/Description	The final score will be given to each document regarding its sensitivity and confidence level.
Inputs	Various data, text, Word, Excel, Image file, etc.
Processing	It will find a score as a confidence score.
Output	**-** Confidence Score

3.2.4 Train Machine

Function Name	**Train Machine**
Purpose/Description	The machine will be trained with a portion of the already gathered data.
Inputs	Various data, text, Word, Excel, Image file, etc.
Processing	The machine will be trained to build a model via these preprocessed data.
Output	None

3.3 Non-Functional Requirements

3.3.1 Performance

The precision must not be under 0.7.

Deciding whether an e-mail can be sent or not shouldn't take too long. The exact parameter for this will be determined later in the project.

We tested that the time required to get a confidence score is 0.74 seconds with our data. Even though the exact time may vary regarding e-mail's contents it will not exceed one second.

3.3.2 Availability

After the training system should be able to take the document at any moment and should be able to give the result back.

3.3.3 Security

Gathered data contains sensitive information so it should be stored in a secure place without an internet connection. After the training, it may be deleted. The process repeats with new data sets.

The main goal of this project is to prevent data leakage in e-mails.

3.3.4 Portability

The system will run on the server environment.

3.3.5 Ease of Use

Sending an e-mail will be a simple process. Users can send it in a few steps. (Exact number of steps can be changed, determined, later on).

4. Analysis – UML

UML (Unified Modeling Language)

4.1 Use Cases

4.1.1 Draw use case diagram

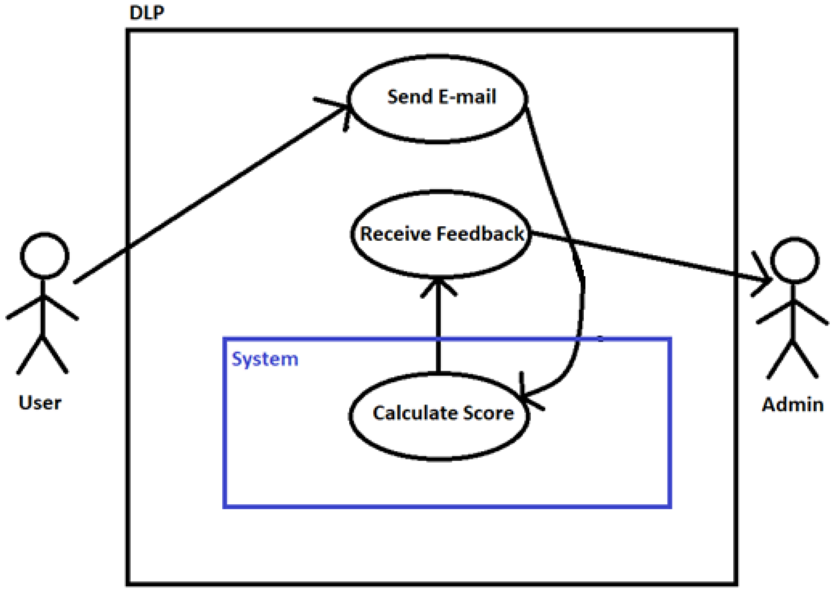


Figure 3

4.1.2.1 Send E-mail

Use Case Name	Send E-mail
Actors	User, a common user of the system.
Trigger	User clicks send button
Overview	User sends e-mail
Precondition	The user should at least write or add something. It should not be blank.
Inputs	Text and various attachments(Word, Excel, Image file, etc.)
Scenario	After the user writes text and adds attachments to the mail he/she will send it to the system.
Exceptions	

4.1.2.2 Receive Feedback

Use Case Name	Receive Feedback
Actors	Admin
Trigger	None
Overview	Admin takes feedback from the system regarding e-mail.
Precondition	
Inputs	None
Scenario	After the user writes text and adds attachments to the mail he/she will send it to the system.
Exceptions	

4.1.2.3 Calculate Score

Use Case Name	Calculate Score
Actors	None
Trigger	New E-mail sending.
Overview	The system calculates the score and sends this score to receive feedback function if the document contains sensitive information.
Precondition	None
Inputs	E-mail and its attachments
Scenario	The system will give a score to each document to get a confidence score. Documents that its confidence score reaches a certain number will be sent to receive the feedback function
Exceptions	None

4.1.2.4 Train Machine

Use Case Name	Train Machine
Actors	None
Trigger	None
Overview	The machine will be trained with the gathered data.
Precondition	Data must be preprocessed in order to be used in the training machine.
Inputs	None
Scenario	Large amounts of data will be sent to the machine in order to build a model to detect sensitive information in the e-mails.
Exceptions	Data might be corrupt because if results are far from ideal.

Software Design Description

1.Introduction

1.1. Purpose

This software design document describes the architecture and system design of the DLP System. The main purpose of this project is to detect documents, texts, and attachments with sensitive information using content-based analysis.

It addresses the SRS document of it and describes the project for the developers.

1.2 Overview

Section 2 is System Overview. It provides an overview of the system, software, libraries, etc.

Section 3, System Design, contains diagrams for the project. In Section 3.2 system is divided into levels. Section 3.3 contains the class diagram of the system. Section 4 contains user interface design, and what users will see when they use the system. It contains some examples.

1.3 Definitions and Acronyms

Term	Definition
Admin	Admin will receive feedback from the system.
BERT	Bidirectional Encoder Representations from Transformers
DLP	Data Leakage Prevention
DLPS	Data Leakage Prevention System
Doc2vec	It is an NLP tool for representing documents as a vector
Mailparser	It will be used for extracting the body and the attachment from the e-mail itself.
NLP	Natural Language Processing
OCR	Optical Character Recognition
REGEX	Regular Expression
Scrapy	It will be used for data crawling
SDD	Software Design Description
SRS	Software Requirements Specification
TensorFlow	It is an end-to-end open-source platform for machine learning.
User	User will send e-mail.

2. System Overview

This document is based on an SRS document of the same system. Some other details can be seen on the SRS document

2.1 Software and Tools Used

We are going to use tika/tesseract OCR for OCR operation. It will take e-mail and its attachments(PDF, JPG, PNG, Doc, etc) and extract text from them. We are planning to clean this data with Mailparser but it can be changed later on.

For tokenizing the data we are going to use BERT. After that for vectorizing we are considering using TensorFlow or doc2vec. It can be changed later on. For constructing the model we are going to use TensorFlow machine learning.

We are also going to set up an e-mail server. We will not implement the e-mail sending interface.

For determining these and understanding whether the gathered data is good enough we conducted a few tests. We tested BERT with cross-validation in TensorFlow. We didn’t manage to get a higher accuracy score than 68.75. Because of that, we chanced the datasets.

Secret data, a total of 2200 entries, comes from various agencies. We used 1600 of it to train the model. Public data comes from NSA news and NATO news so that subjects of secret and public datasets can be more correlated. It also has 1600 entries.

After constructing the model we tested it with BBC and a few other datasets with various subjects. Detailed information about these datasets can be found in the Test Results Document.

2.2 Assumptions

We assume that proper data is already obtained by crawling or any other means and it is labeled to use in our project before the User sends an e-mail and Admin receives feedback from it. Proper data should be suitable for our project. By saying suitable we mean its quantity should not be too low, or the topics vary too much that the quantity needed for it becomes too enormous in order for it to be called “proper data”. The second aspect we called, topic, is the quality aspect of the data and as it can be seen it can be important as the quantity. In short gathering, data process is an offline process prior to the User and Admin’s interactions.

So we assume that the machine is already trained with proper data, both enough quality and quantity. It should be enough to get an acceptable result. Exact metrics for “acceptable results” will be determined later on through the project. It requires training and testing.

But even now we can say that result should not be too below the 0.7 precision. Because it is the minimum required performance.

3. System Design

3.1 Architectural Design

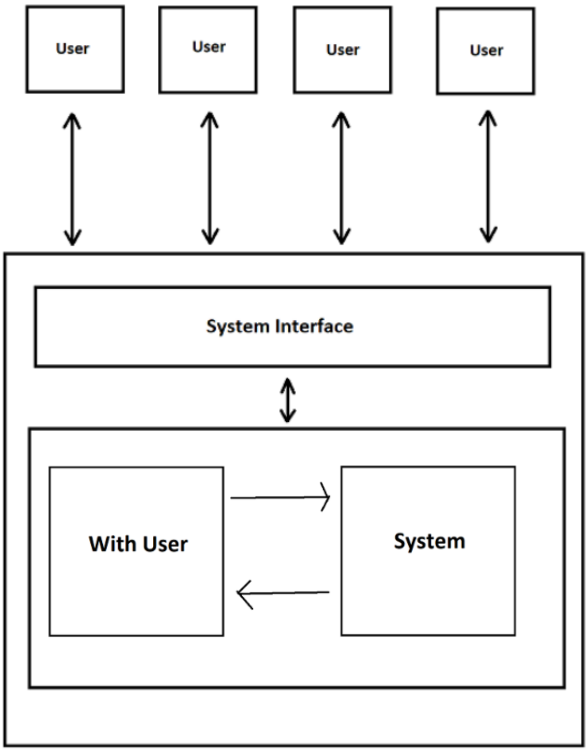


Figure 1: Block Diagram for "DLP system"

3.2 Decomposition Description

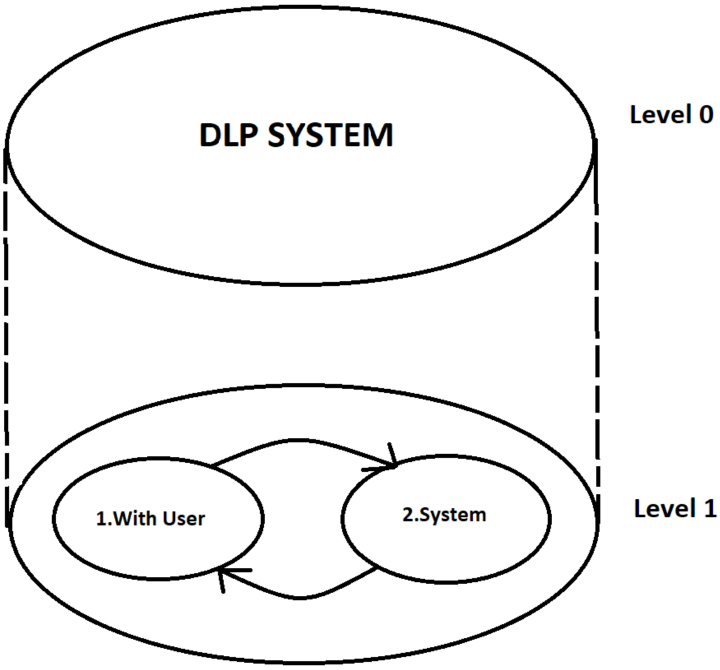


Figure 2

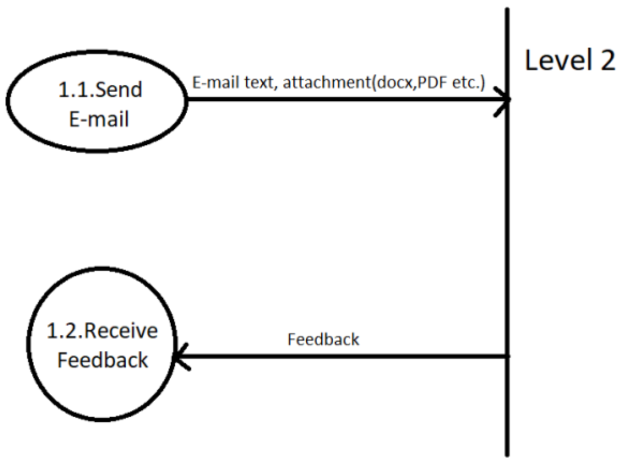


Figure 3

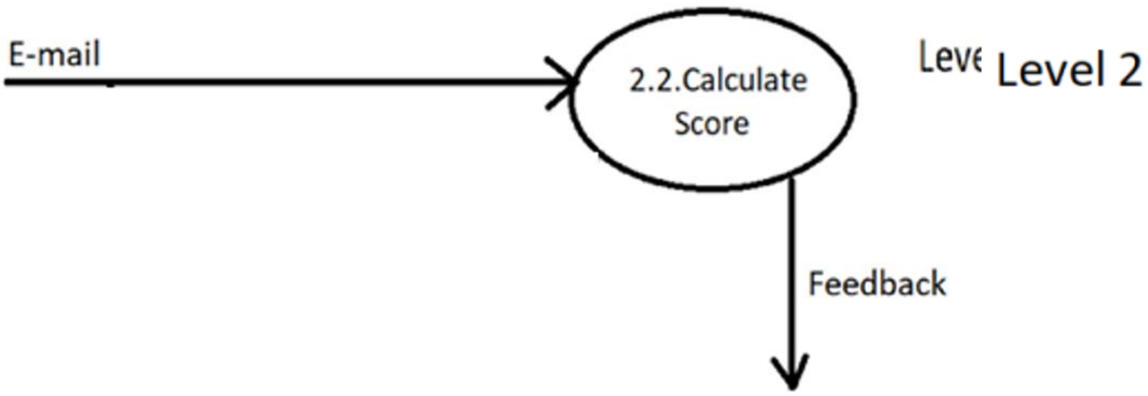


Figure 4

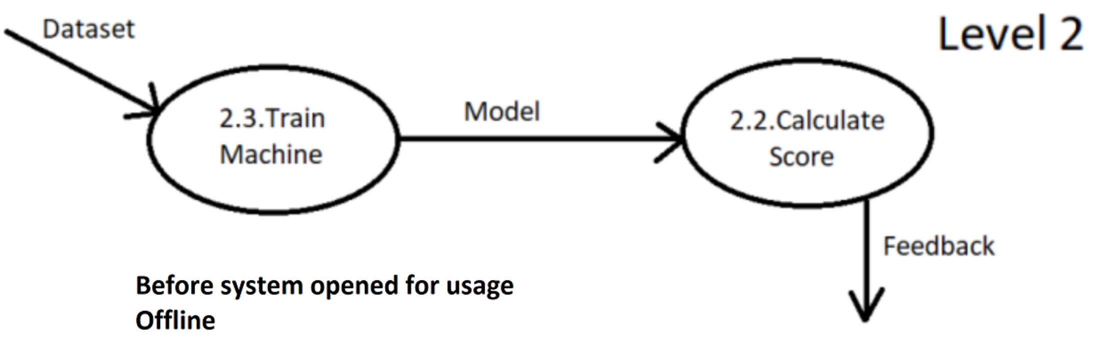


Figure 5

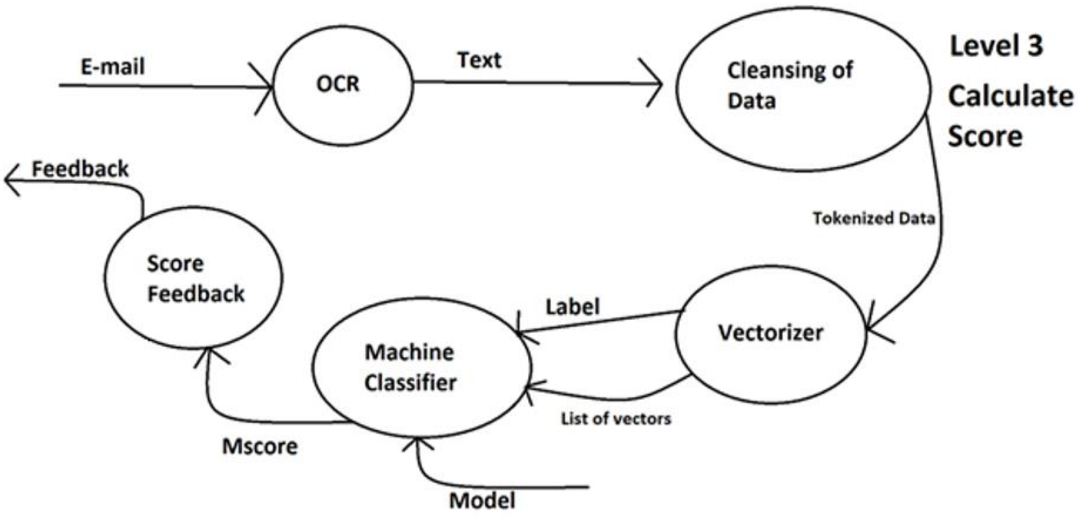


Figure 6

3.3 Activity Diagram

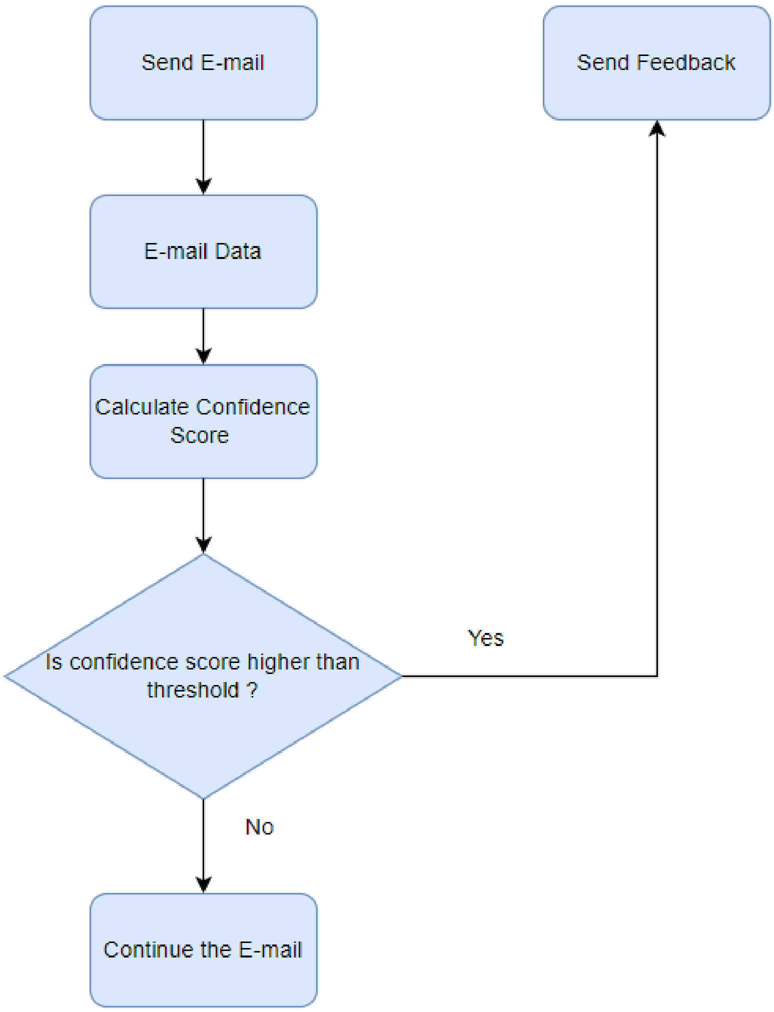


Figure 7

4.Admin Interface Design

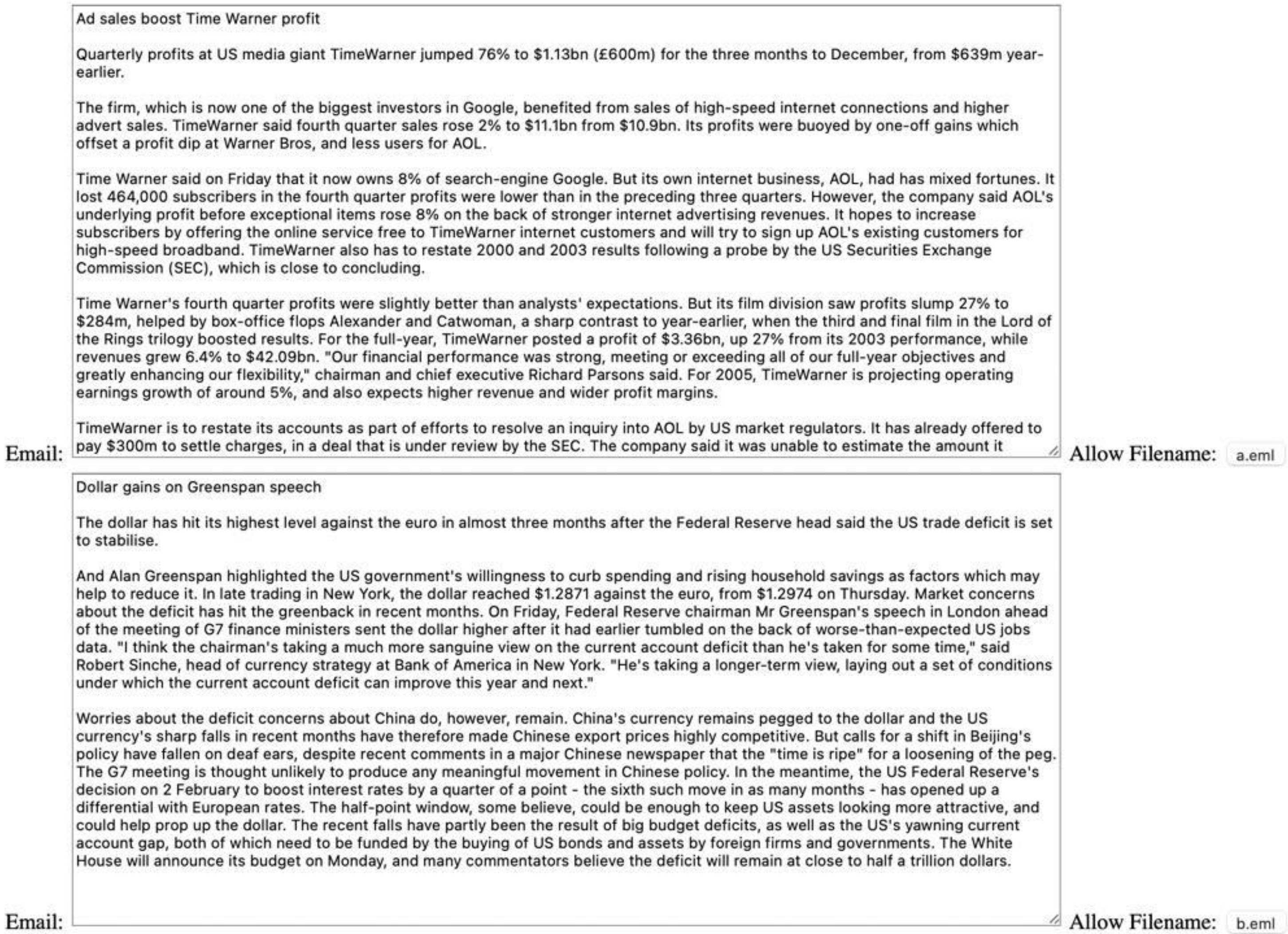


Figure 8: Feedback message, pop up interface

Project Work Plan

Ceng 408 WorkPlan

Project : Content-based Analysis in order to Detect Sensitive Data
Project Start Date : 21.02.2022

TASK	ASSIGNED TO / DUE DATE	Week 1	Week 2	Week 3	Week4	Week5	Week6	Week7	Week8	Week9	Week10	Week11	Week12	Week 13	Week 14	Week 15+
						3/25/2022	4/1/2022						5/13/2022	5/20/2022	5/27/2022	
SRS - SDD	Week(1-5)															
Work Plan	Team															
SRS - SDD	Batuhan - Nazım															
Data	Alper - Batuhan															
Algorithm Tests	Beril - Nazım															
Github /Wiki Implementation	Batuhan															
Test Plan Document	Week(5-6)															
Abstract - Introduction	Team															
Github /Wiki Implementation	Beril															
Software Product	Week(1-12)															
Software	Team															
First Presentation	Team															
Github /Wiki Implementation	Nazım															
User Manual	Week(12-13)															
Implementation	Team															
Github /Wiki Implementation	Alper															
Various Reports and Materials	Week(6-15+)															
Project Test Plan	Nazım - Beril - Batuhan															
Project Report	Batuhan															
Test Results	Nazım - Beril															
Project Poster	Team															
Update Project Webpage	Alper															
Demo Video	Nazım - Beril															
Github /Wiki Implementation	Beril															
Presentation	Week(15-16)															
	Team															

Test Plan

1-Introduction

1.1-Version Control

Version No	Description of Changes	Date
1.0	First Version	April 01, 2022

1.2-Overview

Our project's aim is to detect sensitive/secret data. It is based on machine learning and OCR. The general test plan of this document is mainly prepared to test these two concepts.

1.3-Scope

This document contains information about the test plan of the DLP system.

1.4-Terminology

Acronym	Definition
DLPS	Data Leakage Prevention System
OCR	Optical Character Recognition
SDD	Software Design Document
SMTP	Simple Mail Transfer Protocol
SRS	Software Requirement Specification

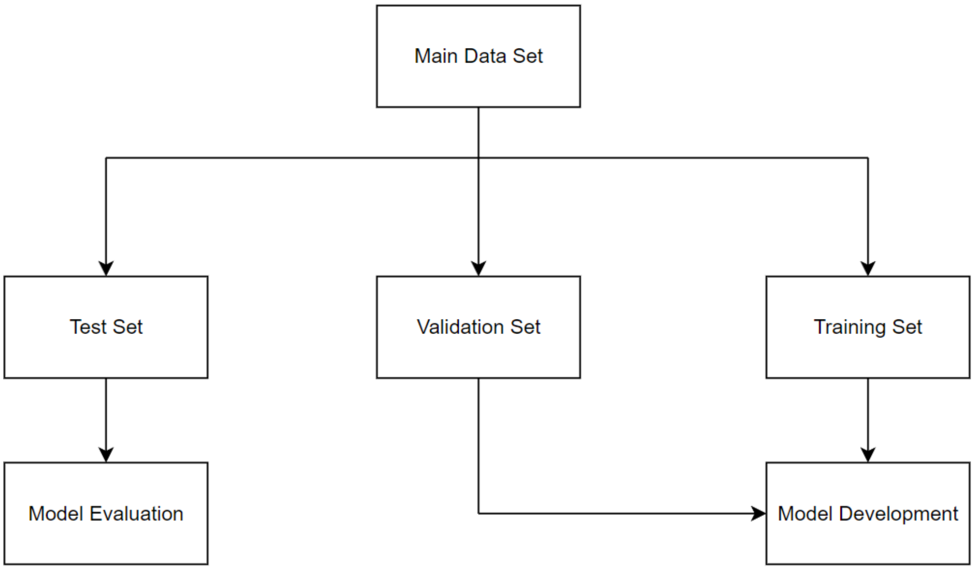
2-Features To Be Tested

2.1-Train(TR)

Train includes the dataset loading, model creation, and model evaluation.

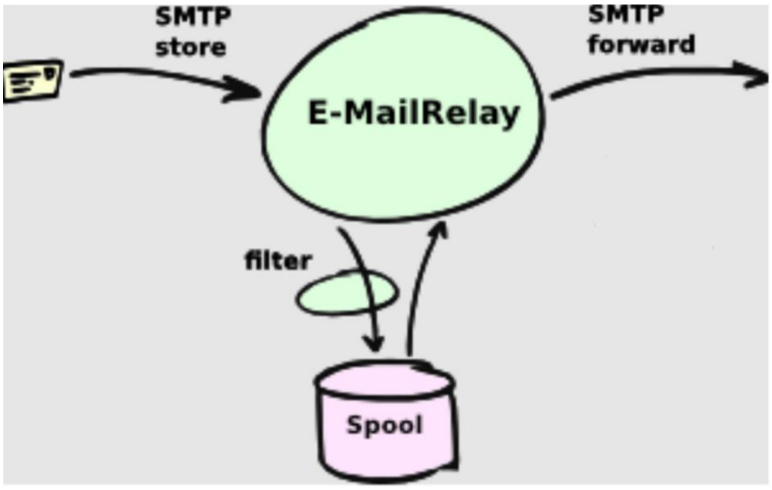
Our total data set consists of 4435 entries, 2225 of them are public entries with various contents such as politics, sport, business, etc. and 2210 of them are secret entries from various agencies such as NSA, ODNI, etc. We will divide 1614 from secret data set and we will use 1614 entry from NSA and NATO news as public data sets for model training. In total training data set consists of 3228 entries. . %80 percent of the training data set will be a training set and %20 will validation set.

The test set for public entries consists of BBC dataset and two other minor datasets. Total of 2539 entries. Their individual entry numbers can be seen at the Test Result part below. 611 entry from the secret data set which we separated earlier will be used as test set as well.



2.2-E-Mail(EA)

We will look into the e-mail if needed OCR will be used to extract its contents. Later mail will be filtered with our model. After filtering if its conditions are met, e-mail has not seemed secret, then it will be forwarded.



3-Features Not To Be Tested

Our project is a machine learning system that decides whether the data contains secret information or not. Therefore feedback interface will not be tested.

4-Item Pass/Fail Criteria

In general, if the data set is properly loaded, and the model successfully created and validated this would be sufficient for the Train feature. And an e-mail successfully received by the E-MailRelay, filtered, and forwarded to the target user properly would be sufficient for the E-Mail feature.

4.1-Exit Criteria

- 100% of the test cases are executed
- 95% of the test cases passed
- All High and Medium Priority test cases passed

5.References

[1] SDD · CankayaUniversity/ceng-407-408-2021-2022-Content-based-Analysis-in-order-to-Detect-Sensitive-Data Wiki. (n.d.). Retrieved March 29, 2022, from https://github.com/CankayaUniversity/ceng-407-408-2021-2022-Content-based-Analysis-in-order-to-Detect-Sensitive-Data/wiki/SDD [2] Software Requirements Specification SRS · CankayaUniversity/ceng-407-408-2021-2022-Content-based-Analysis-in-order-to-Detect-Sensitive-Data Wiki. (n.d.). Retrieved March 29, 2022, from https://github.com/CankayaUniversity/ceng-407-408-2021-2022-Content-based-Analysis-in-order-to-Detect-Sensitive-Data/wiki/Software-Requirements-Specification-SRS [3] E-MailRelay. (n.d.). Retrieved March 31, 2022, from http://emailrelay.sourceforge.net/

6.Test Design Specifications

6.1-Train(TR)

6.1.1-Subfeatures to be tested

6.1.1.1-Data(TR.DT)

We will check whether the data set is properly loaded into the model so that it can be used for training.

6.1.1.2-Model(TR.MD)

Model creation will be checked. Did the model successfully created , or there were some complications that prevented its creation?

6.1.1.3-OCR(TR.OCR)

OCR will be tested with various data types such as PDF, Jpeg, PNG, Tiff, Word, etc. Excluding HTML. The library we use, Tika, is not able to OCR HTML files.

6.1.1.4-Evaluation(TR.EVA)

The created model will be evaluated. Is its accuracy, validation metrics high enough, higher than the required metric values or there are some problems regarding that. If it is deemed necessary hyper tunning of parameters will be done.
The accuracy of the model should be higher than 0.7. Time to respond, model's response to entry should not take more than 5 seconds. The time to respond depends on the configuration of the machine that is running the model. It also depends on the CPU usage at that moment.

6.1.2-Test Cases

Here list all the related test cases for this feature

TC ID	Requirements	Priority	Scenario Description
TR.DT	2.1	H	Did data properly load into our system? Did we successfully open and extract its information from the source file.
TR.MD	2.2	H	The creation of the model will be checked. In case of a failure, problems that caused this failure will be solved. After that model creation will be repeated until successful creation.
TR.OCR	2.3	H	OCR function will be tested. Whether is it able to extract text from different types of data.
TR.EVA	2.4	H	The created model will be evaluated based on its accuracy(higher than 0.7), time to respond (less than 5 seconds), and test accuracy value.

6.2-E-Mail(EA)

6.2.1-Subfeatures to be tested

Does the email sent successfully come into the E-MailRelay?

6.2.1.1-Send E-mail(EA.SE)

The model that we created will be positioned in the filter location. Does the filter work properly?

6.2.1.2-Filter(EA.FL)

The model that we created will be positioned in the filter location. Does the filter work properly?

6.2.1.3-Forward(EA.FW)

The forward function will be checked. Does mail is successfully sent to the target user, does he/she can see the mail.

6.2.2-Test Cases

Here list all the related test cases for this feature

TC ID	Requirements	Priority	Scenario Description
EA.SE	3.1	H	An e-mail will be received into E-MailRelay
EA.FL	3.2	H	Did e-mail filtered successfully ?
EA.FW	3.3	H	Is the target user of the e-mail able to receive it? Can he/she receive and see the e-mail.

7.Detailed Test Case

7.1-TR.DT

TC_ID	TR.DT
Purpose	Load data set from the source file.
Requirements	2.1
Priority	High.
Estimated Time Needed	Less than 1 Minute
Dependency	-
Setup	The data set should be already prepared.
Procedure	[A01] Load data set.
	[A02] Check whether it is loaded properly or not. Compare it with the source.
Cleanup	Exit

7.2-TR.MD

TC_ID	TR.MD
Purpose	Create a model via machine learning.
Requirements	2.2
Priority	High.
Estimated Time Needed	Depends on the epoch value and learning rate. Each epoch takes 10 to 20 minutes. On average 15 epochs will be needed.
Dependency	The data set should be already loaded into the model so that it can train.
Setup	Relevant libraries should already be installed. (scikit-learn, tensorflow, tensorflow_text …)
Procedure	[A01] Start model creation.
	[A02] Wait until the end of creation to check whether it is successful or not.
Cleanup	Exit

7.3-TR.OCR

TC_ID	TR.OCR
Purpose	Test the OCR function
Requirements	2.3
Priority	High.
Estimated Time Needed	It extracts data for around 1 second. The exact time is highly dependent on the content size and data type. We plan to use a total of 50 entries. 10 from each data type.
Dependency	Different types of data should be already prepared
Setup	Tika library should be already installed.
Procedure	[A01] OCR test data will be sent.
	[A02] Results will be compared with the original data.
Cleanup	Exit

7.4-TR.EVA

TC_ID	TR.EVA
Purpose	Evaluate the created model.
Requirements	2.4
Priority	High.
Estimated Time Needed	The model determines one entry around 0.75 to 5 seconds. Total time might change according to test entry numbers. Around 3200 entries. 1600 secret, 1600 public.
Dependency	The model must already be created.
Setup	Test data should be loaded.
Procedure	[A01] At the end of model creation binary accuracy score will be printed to screen. The score will be checked whether is it above the threshold or not.
	[A02] Send entries to the model and check the confidence scores. Did the model successfully classify entries with good accuracy, 0.7, or did it fail?
Cleanup	Exit

7.5-EA.SE

TC_ID	EA.SE
Purpose	Check whether E-MailRelay receives mail successfully.
Requirements	3.1
Priority	High.
Estimated Time Needed	Less than a second.
Dependency	An e-mail should be coming to the e-mail server.
Setup	E-MailRelay should be running.
Procedure	[A01] An e-mail will come to the E-MailRelay
	[A02] Check whether it is successfully came or not.
Cleanup	Exit

7.6-EA.FL

TC_ID	EA.FL
Purpose	Check whether the filter works properly.
Requirements	3.1
Priority	High.
Estimated Time Needed	It should be lesser than 5 seconds for a single mail. Time is subject to change depending on the content size in the e-mail.
Dependency	Mail should be already received by E-MailRelay
Setup	The model should already be created.
Procedure	[A01] E-mail enters the filter.
	[A02] Filter returns a value. (0 for public, -1 for secret data)
Cleanup	Exit

7.7-EA.FW

TC_ID	EA.FW
Purpose	Check whether the target user successfully receives the e-mail.
Requirements	3.1
Priority	High.
Estimated Time Needed	Less than 5 seconds
Dependency	Mail should be received 0 value, public, by the filter. Otherwise, mail will not be attempted to forward.
Setup	E-MailRelay should be running.
Procedure	[A01] E-mail will be forwarded to the target user.
	[A02] Check whether the target user received the e-mail and can see the contents of it.
Cleanup	Exit

Test Results

Our secret data consists of 2225 entries and comes from various agencies, mainly NSA. We used 1600 of it to train the model. Public data comes from NSA news and NATO news so the contents of secret and public datasets can be more closely related. It also has 1600 entries.

625 secret entry used for the test. Model classified them with 99.73 precision.

After constructing the model we tested it with BBC and a few other datasets with various subjects. BBC datasets consist of Business, Entertainment, Politics, Sports, and Tech subjects. Their number can be seen in the table below. Other than BBC we also have two datasets about

BBC	Total Entry Number	Falsely Classified
Business	510	20
Entertainment	386	20
Politics	417	9
Sport	511	1
Tech	401	27

BBC contains a total of 2.225 entries. The falsely classified entry number is 77. Precision is 96.54

Maths and Food. It can also be seen in the below table.

Other Public Datasets	Total Entry Number	Falsely Classified
Food	100	18
Maths	214	10

The model can classify with a high percentage of the subjects it understands.

Test Cases

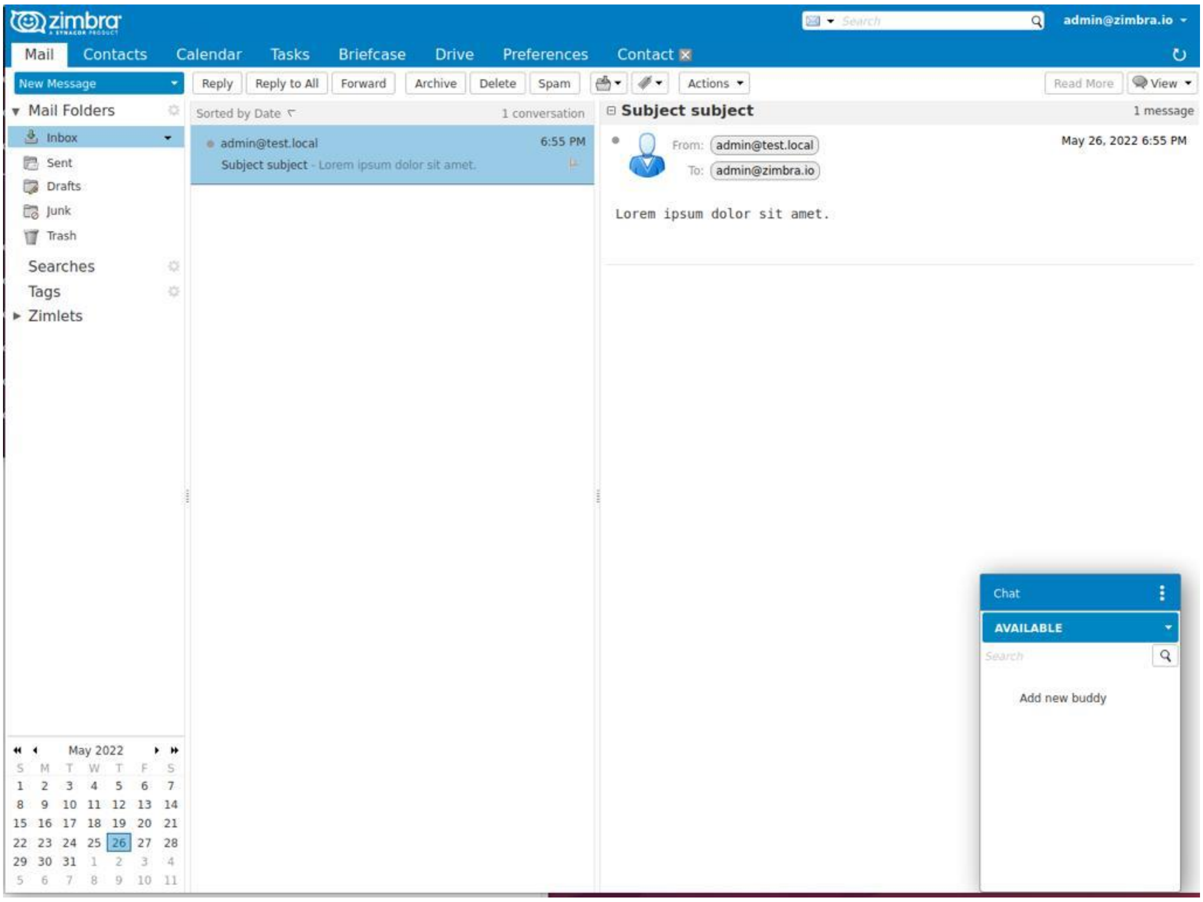
TC ID	Priority	Run By	Result	Explanation
TR.DT	High		Pass	Load train data set to the model.
TR.MD	High		Pass	Construct the model with the given data set.
TR.OCR	High		Pass	Use OCR to get plain text from PDF and its contents. (images)
TR.EVA	High		Pass	Model evaluation.
EA.SE	High		Pass	Sending e-mail.
EA.FL	High		Pass	Filter performance.
EA.FW	High		Pass	Receiving e-mail.

```
emailrelay-parser_1  | 2022-05-26 15:08:44,193 - Server is waiting for port 1235
1/1 [==============================] - 0s 103ms/step
emailrelay-parser_1  | 2022-05-26 15:08:44,354 - Content:
emailrelay-parser_1  |
emailrelay-parser_1  |
emailrelay-parser_1  |  CTIC: It's Not Just Another Pretty Space
emailrelay-parser_1  |
emailrelay-parser_1  | FROM: the Counterterrorism Primary Production Center (S2I)
emailrelay-parser_1  | Unknown
emailrelay-parser_1  |
emailrelay-parser_1  | Run Date: 09/26/2006
emailrelay-parser_1  |
emailrelay-parser_1  |  Okay, the Counterterrorism Innovation Center (CTIC) is different. And a bit
emailrelay-parser_1  | disruptive. CTIC looks and feels like a design studio, recreation room, tradeshow floor, cognitive
emailrelay-parser_1  | systems lab and Internet cafe all rolled into one. Its main function, however, is pretty simple:
emailrelay-parser_1  | CTIC is a place where the right mix of people and technology can help us imagine better ways of
emailrelay-parser_1  | doing our work and help validate various technical approaches to getting it done.
emailrelay-parser_1  |
emailrelay-parser_1  |  The CTIC has three main focus areas. It is:
emailrelay-parser_1  |
emailrelay-parser_1  | 1.       an innovation center where analysts can come together to discover and create new
emailrelay-parser_1  | approaches to a problem;
emailrelay-parser_1  |
emailrelay-parser_1  | 2.       a transformation center where analysts can ""test-drive"" high-powered tools not
emailrelay-parser_1  | previously available to the government, to see how we can use them to improve our
emailrelay-parser_1  | analytic horsepower; and
emailrelay-parser_1  |
emailrelay-parser_1  | 3.       a germination center where analysts can find new ways to use existing tools that have
emailrelay-parser_1  | a proven value in another, specific arena of NSA's business.
emailrelay-parser_1  |
emailrelay-parser_1  |  There are plenty of great ideas out there, and a lot of problem-solving already
emailrelay-parser_1  | underway. Part of CTIC's mission is to amplify and focus such efforts while building a foundation
emailrelay-parser_1  | to tackle bigger, systemic issues.
emailrelay-parser_1  |
emailrelay-parser_1  |  Some of those bigger issues are grouped around target technology themes like
emailrelay-parser_1  | the following:
emailrelay-parser_1  |
emailrelay-parser_1  | •         Contributing to the development of VoiceRT , a voice-services platform. Initially
emailrelay-parser_1  | VoiceRT will focus on voice keyword search technology and then expand to other
emailrelay-parser_1  | applications such as speaker and language identification. (See related article .)
emailrelay-parser_1  |
emailrelay-parser_1  | •         Persona Analysis and Tracking Techniques ( PATT ): PATT intends to translate
emailrelay-parser_1  | data into information sets that analysts can readily absorb, and create data
emailrelay-parser_1  | summarization presentations, such as a summary of a target's activity online.
emailrelay-parser_1  |
emailrelay-parser_1  | •         The CTIC will foster visual analytics innovation. ""Visual analytics"" is a new
emailrelay-parser_1  | term coined for an old concept - that humans are optimized to make judgments on visual
emailrelay-parser_1  | data.
emailrelay-parser_1  |
emailrelay-parser_1  | •         Workflow Management - i.e. efficiently managing those sets of processes
emailrelay-parser_1  | and functions required to fulfill our daily tasks. In the CTIC we will develop technical
emailrelay-parser_1  | solutions so that analysts are relieved from manual processes, such as data entry and
emailrelay-parser_1  | working in multiple GUI's (graphical user interfaces), by giving them a fully automated
emailrelay-parser_1  | workspace.
emailrelay-parser_1  |
emailrelay-parser_1  |  CTIC project teams study specific issues, characterize gaps and shortfalls, imagine
emailrelay-parser_1  | possible solutions, and recommend mature commercial- and government-off-the-shelf (COTS,
emailrelay-parser_1  | GOTS) technologies for evaluation. Other projects, organized around themes like analytic
emailrelay-parser_1  | integration and customer interaction, are intended to lay the foundation for improvements in
emailrelay-parser_1  | collaboration techniques and higher-level analysis.
emailrelay-parser_1  |
emailrelay-parser_1  |  The ""before"" view...
emailrelay-parser_1  |
emailrelay-parser_1  | "" SIDtoday articles may not be republished or reposted outside NSANet
emailrelay-parser_1  | without the consent of S0121 (DL sid comms).""
emailrelay-parser_1  |
emailrelay-parser_1  |
emailrelay-parser_1  |
emailrelay-parser_1  | DERIVED FROM: NSA/CSSM 1-52, DATED 08 JAN 2007 DECLASSIFY ON: 20320108
emailrelay-parser_1  |
emailrelay-parser_1  | 2022-05-26 15:08:44,355 - Score: 0.999787
emailrelay-out_1     | emailrelay: 20220526.150844.355: info: rejected by filter: [Mail Cannot Forward!]
emailrelay-out_1     | emailrelay: 20220526.150844.356: info: smtp connection closed: smtp protocol done: 172.18.0.4:43124
emailrelay-in_1      | emailrelay: 20220526.150844.355: info: failing file: "emailrelay.8.1653577298.3.envelope.busy" -> "emailrelay.8.1653577298.3.envelope.bad"
```

This is the plain text that will go to the Model in the Filter part. This plain text is acquired with the OCR from PDF.

```
emailrelay-parser_1  | 2022-05-26 15:55:46,374 - Server is waiting for port 1235
1/1 [==============================] - 0s 77ms/step
emailrelay-parser_1  | 2022-05-26 15:55:46,484 - Content: Lorem ipsum dolor sit amet.
emailrelay-parser_1  |
emailrelay-parser_1  | 2022-05-26 15:55:46,485 - Score: 0.000581
```

Model classified the entry in 77ms.



E-mail receiving screenshot.

Conclusion

This document contains wide information about our project. It contains Literature Review, SRS, and SDD documents. While preparing this report, dlp systems, mydlp, n-gram technology, fingerprinting, and machine learning were researched and information was obtained about them. We searched for a content-based approach in order to work with the most suitable algorithms that match our purpose. Tools and terms had been covered while learning the concepts and the samples reviewed the related algorithms.

References

Literature Review References

[1] S. Alneyadi, E. Sithirasenan, and V. Muthukkumarasamy, “A survey on data leakage prevention systems,” J. Netw. Comput. Appl., vol. 62, pp. 137–152, 2016, doi: 10.1016/j.jnca.2016.01.008.

[2] R. Mogull, “Understanding and Selecting a Data Loss Prevention Solution,” pp. 1–26, 2010, [Online]. Available: https://securosis.com/assets/library/reports/DLP-Whitepaper.pdf.

[3] L. Wei, P. Xing, J. Zeng, J. X. Chen, R. Su, and F. Guo, “Improved prediction of protein–protein interactions using novel negative samples, features, and an ensemble classifier,” Artificial Intelligence in Medicine, vol. 83, pp. 67–74, Nov. 2017, doi: 10.1016/j.artmed.2017.03.001.

[4] L. Aitchison, N. Corradi, and P. E. Latham, “Zipf’s Law Arises Naturally When There Are Underlying, Unobserved Variables,” PLoS Computational Biology, vol. 12, no. 12, Dec. 2016, doi: 10.1371/journal.pcbi.1005110.

[5] M. van Gompel and A. van den Bosch, “Efficient n-gram, Skipgram and Flexgram Modelling with Colibri Core,” Journal of Open Research Software, vol. 4, Aug. 2016, doi: 10.5334/jors.105.

[6] P. Willett, “Document Retrieval Experiments Using Indexing Vocabularies of Varying Size. II. Hashing, Truncation. Digram and Trigram Encoding of Index Terms.” J. Doc. 35, 296, 1979.

[7] W. B. Cavnar, “N-gram-based Text Filtering for TREC-2,” The Second Text Retrieval Conference (TREC-2), NIST Special Publication 500-215, National Institute of Standards and Technology, Gaitherburg, Maryland, 1994.

[8] E. J. Yannakoudakis, P.Goyal, and J. A. Huggill, “The Generation and Use of TextFragments for Data Compression,” Inf. Proc. Mgt. 18, 15, 1982.

[9] C. Y. Suen, “N-gram Statistics for Natural Language Understanding and Text Processing,” IEEE Trans. on Pattern Analysis & Machine Intelligence. PAMI, 1(2), pp.164-172, April 1979.

[10] R. C. Angell, G. E. Freund, and P. Willette, “Automatic Spelling Correction Using Trigram Similarity Measure,” Inf. Proc. Mgt. 18, 255, 1983.

[11] L. Aitchison, N. Corradi, and P. E. Latham, “Zipf’s Law Arises Naturally When There Are Underlying, Unobserved Variables,” PLoS Computational Biology, vol. 12, no. 12, Dec. 2016, doi: 10.1371/journal.pcbi.1005110.

[12] 1 N. Heintze, “Scalable Document Fingerprinting,” 1996 USENIX Work. Electron. Commer., pp. 191–200, 1996, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.8072.

[13] 2 Y. Shapira, B. Shapira, and A. Shabtai, “Content-based data leakage detection using extended

fingerprinting," 2013, [Online]. Available: http://arxiv.org/abs/1302.2028.

[14] 4 S. Ulahannanl and R. Jose, "Fuzzy Fingerprint Method for Detection of Sensitive Data Exposure," Int. J. Cybern. Informatics, vol. 5, no. 2, pp. 61–69, 2016, doi: 10.5121/ijci.2016.5207.

[15] E. D. Liddy, "SURFACE SURFACE Center for Natural Language Processing School of Information Studies (iSchool) 2001 Natural Language Processing Natural Language Processing Natural Language Processing 1." [Online]. Available: https://surface.syr.edu/cnlp

[16] Kırcicek, O., 2021. Doğal Dil İşleme Nedir ve Uygulama Alanları Nelerdir?. [online] https://www.veribilimiokulu.com/. Available at: https://www.veribilimiokulu.com/dogal-dil-isleme-nedir-ve-uygulama-alanlari-nelerdir/ [Accessed 8 November 2021].

[17] Ikonomakis, Emmanouil & Kotsiantis, Sotiris & Tampakas, V.. (2005). Text classification: a recent overview. 125.

[18] A.TANTUG Cuneyd, "Metin Sınıflandırma Text Classification". .

[19] 19 Matheny, Michael E et al. "Detection of blood culture bacterial contamination using natural language processing." AMIA ... Annual Symposium proceedings. AMIA Symposium vol. 2009 411-5. 14 Nov. 2009

[20] Devopedia. 2019. "Regular Expression." Version 7, August 29. Accessed 2021-09-09. https://devopedia.org/regular-expression

[21] "kyilmaz80/emailrelay-dlp: An e-mail leakage prevention filter for postfix." https://github.com/kyilmaz80/emailrelay-dlp (accessed Nov. 12, 2021).

[22] S. Yousef, "Data Leakage / Loss Prevention (DLP) Systems Analysis and Solutions Submitted by In partial fulfillment of requirements for the degree of Master in," 2015.

[23] 21 "Support Vector Machine Python Example | by Cory Maklin | Towards Data Science." https://towardsdatascience.com/support-vector-machine-python-example-d67d9b63f1c8 (accessed Nov. 12, 2021).

[24] 22 M. Hart, P. Manadhata, and R. Johnson, "Text classification for data loss prevention," HP Lab. Tech. Rep., no. 114, pp. 1–21, 2011.

[25] 23 "DECISION BOUNDARY FOR CLASSIFIERS: AN INTRODUCTION | by Suchismita Sahu | Analytics Vidhya | Medium." https://medium.com/analytics-vidhya/decision-boundary-for-classifiers-an-introduction-cc67c6d3da0e (accessed Nov. 12, 2021).

[26] "Data Loss Prevention Software | Best Data Loss Protection Solutions." https://mydlp.com/ (accessed Nov. 12, 2021).

[27] "googleapis/python-dlp." https://github.com/googleapis/python-dlp (accessed Nov. 12, 2021).

[28] "InfoType detector reference | Data Loss Prevention Documentation." https://cloud.google.com/dlp/docs/infotypes-reference (accessed Nov. 12, 2021).

SRS References

[1] Learn About Sensitive Information Types ,2021 [Online]. Available:

https://docs.microsoft.com/en-us/microsoft-365/compliance/sensitive-information-type-learn-about?view=o365-worldwide

[2] IEEE Computer Society. Software Engineering Standards Committee., & IEEE-SA Standards Board. (1998). *IEEE recommended practice for software requirements specifications*. Institute of Electrical and Electronics Engineers.

SDD References

[1] Engineering Standards Committee of the IEEE Computer Society, S. (2009). *IEEE Std 1016-2009 (Revision of IEEE Std 1016-1998), IEEE Standard for Information Technology—Systems Design—Software Design Descriptions*.

[2] *How to Write a Software Design Document (SDD)*. (n.d.). Retrieved December 31, 2021, from https://www.nuclino.com/articles/software-design-document

[3] S. Alneyadi, E. Sithirasenan, and V. Muthukkumarasamy, "A survey on data leakage prevention systems," J. Netw. Comput. Appl., vol. 62, pp. 137–152, 2016, doi: 10.1016/j.jnca.2016.01.008.

[4] R. Mogull, "Understanding and Selecting a Data Loss Prevention Solution," pp. 1–26, 2010, [Online]. Available: https://securosis.com/assets/library/reports/DLP-Whitepaper.pdf.