JOURNAL OF INFORMATION SYSTEMS APPLIED RESEARCH

Volume 17, No. 1
April 2024
ISSN: 1946-1836

In this issue:

iscap
Information Systems & Computing Academic Professionals

The **Journal of Information Systems Applied Research** (JISAR) is a double-blind peer reviewed academic journal published by ISCAP, Information Systems and Computing Academic Professionals. Publishing frequency is three issues a year. The first date of publication was December 1, 2008.

JISAR is published online (https://jisar.org) in connection with the ISCAP (Information Systems and Computing Academic Professionals) Conference, where submissions are also double-blind peer reviewed. Our sister publication, the Proceedings of the ISCAP Conference, features all papers, teaching cases and abstracts from the conference. (https://iscap.us/proceedings)

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the conference. At that point papers are divided into award papers (top 15%) and other submitted works. The non-award winning papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in JISAR. Currently the acceptance rate for the journal is approximately 35%.

Questions should be addressed to the editor at editor@jisar.org or the publisher at publisher@jisar.org. Special thanks to members of ISCAP who perform the editorial and review processes for JISAR.

2024 ISCAP Board of Directors

JOURNAL OF
INFORMATION SYSTEMS APPLIED RESEARCH

Using Textual Analytics to Process Information Overload of Cyber Security Subreddits

Stephanie Omakwu
so05640@georgiasouthern.edu

Hayden Wimmer
hwimmer@georgiasouthern.edu

Department of Information Technology
Georgia Southern University
Statesboro, GA 30460, USA

Carl M Rebman, Jr.
carlr@sandiego.edu
Knauss School of Business
Department of Supply Chain, Operations, and Information Systems
University of San Diego
San Diego, CA 92110, USA

Abstract

Increases in digitalization have made it possible to track and measure every click, every payment, every message, and almost everyone's daily thoughts. Companies are extremely interested in the robustness of this data, specifically regarding understanding the sentiment of consumers. Yet the amount of information being produced and processed is quite staggering causing information overload. As such, companies tend to fall into analysis paralysis which can result in missing important insights that could help their business. The goal of this study is to analyze and categorize the top posts on multiple hacking subreddits to determine the most discussed topics and to examine the sentiment of these posts expressed by the users. We began by scraping data, specifically the title, ID, score, comments, and URL for each top post from multiple hacking subreddit communities. We then used the Natural Language Toolkit (NLTK) to perform the data preprocessing techniques for an effective analytic process and bias-free results. The results of the testing allowed us to filter through the posts and determine whether sentiment was positive, negative, or neutral. In the case of the hacking subreddits, many of the posts were of a neutral opinion. This study aims to provide a contribution by utilizing Natural Language Processing methods Topic Modeling such as Term Frequency Inverse Document Frequency, Latent Semantic Analysis (LSA) algorithm, and Sentiment Analysis to gather and synthesize cybersecurity data.

Keywords: information overload, text analytics, sentiment analysis, LSA, term frequency, NLP

Recommended Citation: Omakwu, S., Wimmer, H., Rebman Jr., C., (2024). Using Textual Analytics to Process Information Overload of Cyber Security Subreddits. *Journal of Information Systems Applied Research.* 17(1), pp 64-74, https://doi.org/10.62273/AJJR5232

Using Textual Analytics to Process Information Overload of Cyber Security Subreddits

Stephanie Omakwu, Hayden Wimmer and Carl M. Rebman, Jr.

1. INTRODUCTION

More and more data is generated as a result of the rising digitalization of our daily lives. This data comes from a highly diverse range of sources. Approximately 90% of the data generated by organizations is categorized as unstructured data. The size of the data produced daily is astonishing and more than any human and many computers could handle. Most of the data contains a significant amount of text that is written in natural language that includes slang and sarcasm. This poses a number of difficult issues for extraction and synthesis during analysis (Carnot et al., 2020).

Unstructured data comes in various forms which does not usually fit neatly into traditional data models. It is also a challenge to store and manage unstructured data because it also does not easily fit into relational databases or spreadsheets like Excel. These challenges have historically hindered analysis and search efforts, thus rendering it less useful for organizations. However, with the rise of modern business intelligence tools and platforms the landscape has changed. These innovations now enable companies and organizations the ability to manage and conduct analysis of unstructured data (Jain et al., 2020).

This research uses computer-based techniques like Natural Language Processing (NLP) to study cybersecurity data. Specifically, this study uses Topic Modeling, which assists in finding out what subjects are talked about the most, and Sentiment Analysis, which helps in understanding how people feel about certain discussions. It is important for businesses to keep up with the latest cybersecurity practices, especially in our interconnected world. This way, companies can build a strong defense against changing cyber risks and ensure the safety of their data. The goal of the study is to analyze and categorize the top posts on multiple hacking subreddits to determine the most discussed topics and to examine the sentiment of the top posts expressed by users in hacking subreddits.

2. BACKGROUND

Over the past twenty years computational linguistics, also known as the rule-based modeling of human language, has developed into an intriguing field of study as well as a useful technology that is increasingly being implemented into consumer products such as Apple's Siri and Skype Translator. These advancements were made possible by four major factors: (i) a significant increase in computing power; (ii) the availability of extremely large amounts of linguistic data; (iii) the development of highly effective machine learning (ML) methods; and (iv) a much deeper understanding of the structure of human language and how it is used in social contexts (Hirschberg & Manning, 2015).

When computational linguistics is combined with statistical, machine learning, and deep learning models it forms the basis of ‘natural language processing’ or NLP. NLP is considered to be part of artificial intelligence (AI) technologies that is concerned with providing computers the capacity to comprehend written and spoken words. The goal of the technology is to provide computers the ability to comprehend human language in the form of text or speech data and to "understand" its full meaning, including the speaker's or writer's intention and sentiment (IBM, 2023). The very first step in the NLP project model creation is to execute text preprocessing which is transforming text into a clean and consistent format. Preprocessing procedures include the following:

Removal of Punctuation
This involves eliminating all of the text's punctuation using the Python's string library which comes pre-loaded with a number of punctuation marks, including '!"#$%&'()*+,-./:;?@[]_'|'. Removal of punctuation will aid in treating each text equally (Rahimi & Homayounpour, 2023).

Removal of Stopwords
Stop words are the words that are often excluded before processing natural language. These are the most prevalent words in any language (such as articles, prepositions,

pronouns, conjunctions, etc.), yet they do not significantly add to the text's content. The words "the," "a," "an," "so," and "what" are a few stop words in English. Any human language has an abundance of stop words(Giri & Banerjee, 2023). By removing these words, we can make our text more focused on the key information by eliminating the low-level information. A list of terms that are regarded as stop words in the English language is included in the NLTK library. Removing Stop words decreases the dataset size and, as a result, the training time because there are fewer tokens to be trained.

Tokenization

Tokenization is an easy procedure that turns raw data into a meaningful data string, it involves splitting sentences into smaller units referred to as tokens. For example, "Computers understand language with NLP." when broken into tokens will be "Computers", "understand", "language", "with", "NLP", ".". Tokenization is most known for its applications in cybersecurity and the development of non-fungible token (NFT) a specific kind of digital asset that uses blockchain technology to signify ownership and authenticity of a particular object or piece of content, because NFTs are non-fungible, each token is unique and cannot be traded for another token of the same kind. Tokenization also plays a significant role in the NLP process as it is a technique used to break down phrases and paragraphs into simpler language-assignable elements (Choo & Kim, 2023).

Stemming

By simply combining the suffix with the basic root of the word, stemming reduces a word to its stem while maintaining the word's semantic meaning. Stemming disregards grammatical conventions of the language (Pramana et al., 2022). For example, "producing, production" is stemmed into its root word "produc."

Lemmatization

Lemmatization, in contrast to stemming, must always result in a real word form. Lemmatization reduces word variants by removing inflectional endings and returning the term to its base or dictionary form (Pramana et al., 2022). For example, the word "paraphrasing" is lemmatized to its based form "paraphrase".

3. LITERATURE REVIEW

This section discusses relevant literature that was used as the basis of our methodology. Specifically, these studies address our use and selection of classifier models, sentiment analysis techniques such as Latent Semantic Analysis, Industry 4.0, fake news, social media, algorithms, and natural language processing methods.

Kumar and Subba (2020) emphasized how sentiment analysis frameworks have high energy and processing requirements, which restricts their ability to be deployed in real time. They found the frameworks performed poorly when applied to corpora of textual data that contain emoticons and other unusual texts. Kumar and Subba (2020) proposed a sentimental analysis framework based on Support Vector Machine (SVM). Real-time sentiment analysis of the text documents was performed using the trained SVM classifier model and the deployment phase where Real-time sentiment analysis of the text documents is performed using the trained SVM classifier model. The proposed sentimental analysis framework outperformed previous similar frameworks proposed in the literature, according to experimental results on the Amazon electronics item review dataset and the IMDB movie review data corpus.

Subba and Gupta (2021) presented a brand-new host intrusion detection system (HIDS) framework based on truncated singular value decomposition (SVD) and tfidfvectorizer for the real-time detection of unusual system operations. The system calls trace files that are entered into the proposed HIDS framework, which converts them into n-gram feature vector representational models and calculates the tfidf values via tfidfvectorizer. Then, depending on their tfidf values, the modified n-gram feature vectors are dimension reduced using truncated SVD. According to experimental findings using the benchmark datasets ADFA-LD and ADFA-WD, the proposed HIDS framework efficiently and effectively detects abnormal system operations with little processing cost.

Wagire et al. (2020) identified that Industry 4.0 experienced a rapid increase in research articles while being a relatively new and developing subject of study. The goal of their research was to identify "patterns of research" in the area of Industry 4.0. The enormous corpus of 503 academic paper abstracts published in various journals and conference proceedings is reviewed and knowledge is extracted using Latent Semantic Analysis (LSA). The technique used retrieves several latent elements that define the newly developing study pattern. High-loaded papers are subjected to cross-loading analysis to determine the semantic relationship between research fields and themes. The findings of the

LSA reveal 100 study topics and 13 major research fields. The survey identifies "new business model" and "smart factory" as the two most important research areas. A taxonomy is created that includes five industrial theme categories. 4.0 field.

The most significant and well-known unsupervised techniques in automatic voice recognition are Maximal Marginal Relevance (MMR) and Latent Semantic Analysis (LSA). The goal of Ramezani et al. (2023) was to determine how well the two unsupervised algorithms described above performed when transcribing summaries of Persian broadcast news. The 58 news documents in the corpus used in this study were derived from the manual transcription of Persian broadcast news over a period of more than 15 hours. Over the course of 45 days, it was gathered by four native transcribers from the news on three radio channels and four TV channels. Over 115,000 words and 7,000 sentences make up the corpus. The findings indicate that MMR performs better than LSA in query-based broadcast news summarization while LSA outperforms MMR in generic summarization.

Mayopu et al. (2023) showed that fake news is constantly created to mislead readers, spreads quickly, and has a significant negative impact on human civilization through social media. This project intended to create an efficient method that combines latent semantic analysis (LSA) and natural language processing (NLP) utilizing singular value decomposition (SVD) techniques to assist social scientists in analyzing fake news and identifying its precise components. In order to construct a summary by recognizing a latent or hidden semantic structure, latent semantic analysis (LSA) was used to extract relevant sentences from an input material. The LSA approach models gathered documents as a term-document matrix (TDM) and employs singular value decomposition (SVD) to derive concepts from collected documents. The efficacy of Mayopu et al. (2023) techniques was illustrated using a genuine scenario from the 2016 United States presidential election campaign which displayed five concepts and were taken from the LSA and are indicative of political fake news during an election.

Numerous laws are made by regulatory authorities and must be adhered to, as a result, finding regulatory non-compliance involves complex compliance requirements and time-consuming procedures. Information Technology (IT) offers a wide range of governance, management, and security frameworks to let firms conduct their processes at a much more advanced level. Huyut et al. (2022) presented a method based on Latent Semantic Analysis (LSA) to produce a particular relatedness correlation map to have an objective and statistical relationship map. A relatedness map between a banking regulation and a best practice was made by (Huyut et al., 2022), under regard to the 1202 actions under Control Objectives for Information Technologies (Cobit 2019).

Huyut et al. (2022) examined 224 statements of this regulation and they used multi-criteria decision-making (MCDM) analysis techniques to support their LSA results. Fuzzy Analytics Hierarchy Process (FAHP) was used to prioritize their criteria, and Weighted Aggregated Sum Product Assessment Method (WASPAS) was used to compare the results of similarity tests between pairs of regulations and Cobit activities.

Chiny et al. (2022) performed an exploratory study on information collected from Flixable, a search engine that displays Netflix material. The TF-IDF and Cosine similarity algorithms, which are popular models in Natural Language Processing (NLP), were also employed to construct a recommendation system used to highlight crucial information about the material offered on this site through evaluation of a dataset of 7,787 unique records. A crucial step of the analysis was the Word Cloud library, a component of an NLP software stack, used to calculate the word clouds' densities. In order to examine the similarities between the titles and descriptions of the Netflix programs, Chiny et al. (2022) applied the TF-IDF and Cosine Similarity algorithms to them. Countries including the United States, India, and the United Kingdom stood out to be the nations with the greatest availability of Netflix material based on investigation emphasized data on the distribution of programs broadcast by genre (69% movies and 31% TV shows). Their study also highlighted the word-by-word analysis of the phrases used in the televised programs.

Prasanth et al. (2022) outlined the methodology used by team CENTamil to identify offensive comments in Tamil. The goal of this investigation was to determine whether a particular comment contains offensive language. To build feature vectors, the author utilized TF-IDF with char-wb analyzers and the Random Kitchen Sink (RKS) algorithm. The YouTube comments in the datasets were first preprocessed, and the preprocessed texts were then transformed into vectors. To make the data clean, noise was removed at the preprocessing stage. For classification of the YouTube comments, a Support Vector Machine (SVM) classifier with a polynomial kernel was employed. Using this technique, Prasanth et al. (2022) was able to rank first with a f1-score of 0.32 for the Tamil dataset and seventh with a f1-score of 0.25 for the Tamil-English dataset, respectively.

Liang and Niu (2022) hypothesized that text classification, a method used in sentiment analysis, intelligent recommendation systems, and intelligent question-and-answer systems, could automatically categorize and label text in accordance with predefined rules. The bidirectional LSTM input structure and the TF-IDF method were changed in this paper. The author specifically updated the TF-IDF calculation and used a sliding window to parse the words. This study trained the neural network, validated it, and tested it using the Sohu news dataset. In an 8:1:1 ratio, the dataset was split into training sets, validation sets, and testing sets. The text features were extracted using a combination of bidirectional LSTM and Text-CNN for classification prediction, with the word2vec vector serving as the word embedding layer. While Text-CNN can extract local crucial features at the sentence level, Bidirectional LSTM treats texts as sequences to understand information. Good precision was attained as a result.

4. METHOD

It is important for businesses to keep up with the latest cybersecurity practices, especially in an interconnected world. In doing so, companies can build a strong defense against changing cyber risks and ensure the safety of their data.

Using Python's PRAW (python Reddit API wrapper) module, which enables Reddit API using Python scripts, we first began by scraping data, specifically the top posts title, ID, score, comments, and URL for each post, from multiple hacking subreddit communities into a.csv file. We used the Natural Language Toolkit (NLTK) to perform the following data preprocessing techniques for an effective analytic process and bias-free results: punctuation removal, tokenization, stop word removal, stemming, and lemmatization.

On completing the data preprocessing, we further analyzed the data by performing the Term Frequency Inverse Document Frequency (TFIDF) Vectorization Using Tfidfvectorizer, Latent Semantic Analysis (LSA) and Sentiment Analysis. The hardware machine that is enabling the implementation of this experiment is an 11th Gen Intel(R) Core (TM) i7-1165G7 @ 2.80GHz 2.80 GHz processor, 16.0 GB (15.7 GB usable) RAM, Windows 10 Home 64-bit operating system, x64-based processor and Python as the language used for analysis.

The TFIDF results for all subreddits indicate that "Hack", "Hacking", "Ethical", and "Comptia" based on their TFIDF scores were the most significant words. With the LSA results identifying the 3 most discussed topics in the subreddits as certifications, techniques on how to do ethical hacking, and training tools. Lastly the base Polarity indicated that there were more Neutral than positive and a little fewer negative sentiment from Top posts in the subreddits.

System Architecture

Figure 1 displays the system architecture on which the project was built on. Visual studio code was utilized as an enabling environment for all phases of this project.

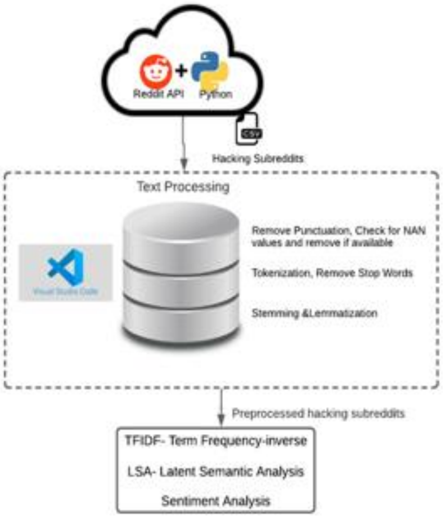

Reddit API
+
Python
CSV
Hacking Subreddits
Text Processing
Visual Studio Code
Remove Punctuation, Check for NAN values and remove if available
Tokenization, Remove Stop Words
Stemming &Lemmatization
Preprocessed hacking subreddits
TFIDF- Term Frequency-inverse
LSA- Latent Semantic Analysis
Sentiment Analysis

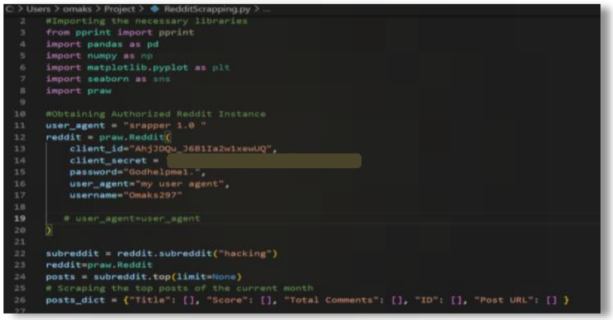
Figure 1: System Architecture

Data Selection and Preprocessing

Data for this study was pulled from Reddit which is a popular platform where experts and students can share knowledge and experiences, there are numerous groups, thousands of communities within it called subreddits where users can converse about articles on predetermined subjects whether you enjoy technology, sports, breaking news, TV fan theories, or a never-ending stream of the prettiest animals online.

Millions of individuals post, vote, and comment every day in communities/subreddits catered to their interests across the globe, with reddit having a daily active users of over 57 Million users, over a 100k active subreddits/communities and over 13 billion posts and comments (Dive Into Anything, 2023). All Top posts from their creation date were extracted from hacking subreddits dedicated to hacking content into .csv files, the subreddits were selected based on popularity.

Attributes such as title, score, id, total comments, and post url were extracted from r/ethicalhacking (389k) with 1001 top posts, r/hackintutorials (228k) with 997 top posts and r/hacking (2.6miilon) with 979 top posts. We started off by scraping data specifically the top posts title, ID, score, comments, and URL for each post from multiple hacking subreddit communities into a .csv file using python's PRAW (python Reddit API wrapper) module which allows Reddit API through python scripts(Boe., 2023).

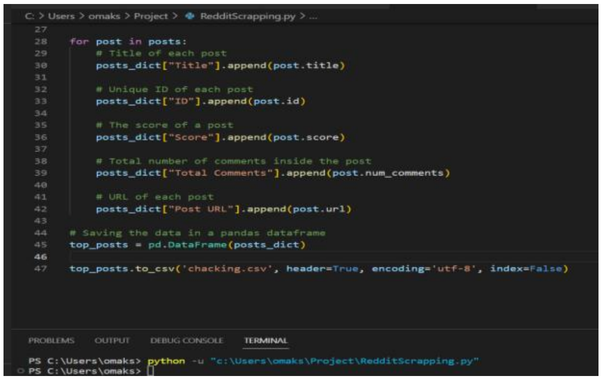


```
#Importing the necessary libraries
from pprint import pprint
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import praw

#Obtaining Authorized Reddit Instance
user_agent = "srapper 1.0 "
reddit = praw.Reddit(
    client_id="AhjJDQu_J681Ia2w1xewUQ",
    client_secret = [redacted]
    password="Godhelpme1.",
    user_agent="my user agent",
    username="Omaks297"

   # user_agent=user_agent
)

subreddit = reddit.subreddit("hacking")
reddit=praw.Reddit
posts = subreddit.top(limit=None)
# Scraping the top posts of the current month
posts_dict = {"Title": [], "Score": [], "Total Comments": [], "ID": [], "Post URL": [] }
```

Figure 2*:* Creating PRAW Instance

To enable data extraction from reddit the PRAW library was installed, then we created a reddit developer account under which we created a Reddit app which generated the client_Id, secrete and user agent values which we used to connect to Reddit using Python. To give us access to perform all possible actions on reddit a PRAW authorized instance was created.

To improve the consistency, accuracy, and reliability of our dataset, for an efficient analysis process, and non-bias result Natural Language toolkit (NLTK) a leading platform for building Python programs to work with human language data was utilized to perform data preprocessing techniques such as removal of punctuations, removal of stop words, tokenization, stemming, and lemmatization. NLTK provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum("Natural Language Toolkit," 2023).



```

for post in posts:
    # Title of each post
    posts_dict["Title"].append(post.title)

    # Unique ID of each post
    posts_dict["ID"].append(post.id)

    # The score of a post
    posts_dict["Score"].append(post.score)

    # Total number of comments inside the post
    posts_dict["Total Comments"].append(post.num_comments)

    # URL of each post
    posts_dict["Post URL"].append(post.url)

# Saving the data in a pandas dataframe
top_posts = pd.DataFrame(posts_dict)

top_posts.to_csv('chacking.csv', header=True, encoding='utf-8', index=False)
```

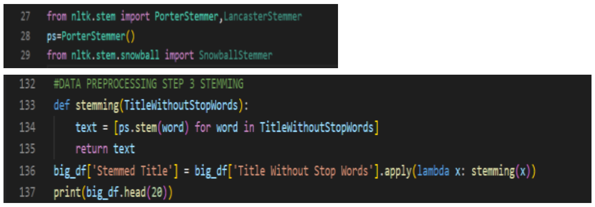
```
PS C:\Users\omaks> python -u "c:\Users\omaks\Project\RedditScrapping.py"
PS C:\Users\omaks>
```

Figure 3: Scraping the Top posts from a subreddit

For data preprocessing and cleaning we began by Removing Punctuation using a customized created method in collaboration with a for loop.

Data Preprocessing–Removing Punctuation

For easier interpretation the dataset was split from sentences into smaller units called token which can be more easily assigned meaning. The the word_tokenize function under the NLTK.tokenize package was utilized for the tokenization procedure.



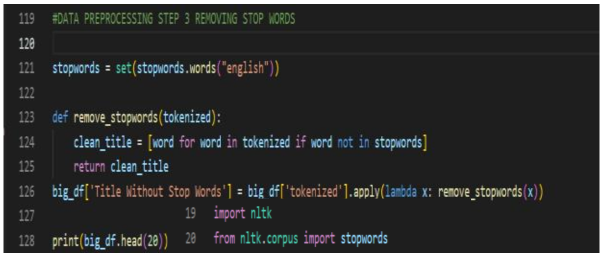
```
from nltk.stem import PorterStemmer,LancasterStemmer
ps=PorterStemmer()
from nltk.stem.snowball import SnowballStemmer
```

```
#DATA PREPROCESSING STEP 3 STEMMING
def stemming(TitleWithoutStopWords):
    text = [ps.stem(word) for word in TitleWithoutStopWords]
    return text
big_df['Stemmed Title'] = big_df['Title Without Stop Words'].apply(lambda x: stemming(x))
print(big_df.head(20))
```

Figure 4: Removal of Punctuation

Data Preprocessing - Removing Stop words

To create room for the ability to focus more on the most important information in the dataset we removed Stopwords using a customized created method with a for loop.



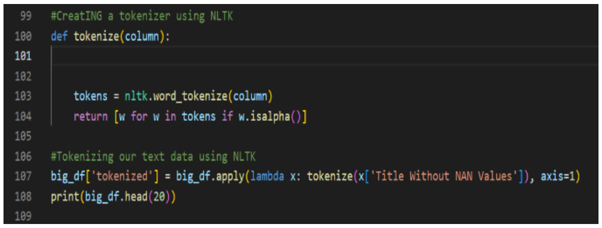
```
#DATA PREPROCESSING STEP 3 REMOVING STOP WORDS


stopwords = set(stopwords.words("english"))

def remove_stopwords(tokenized):
    clean_title = [word for word in tokenized if word not in stopwords]
    return clean_title
big_df['Title Without Stop Words'] = big_df['tokenized'].apply(lambda x: remove_stopwords(x))

print(big_df.head(20))
```

```
import nltk
from nltk.corpus import stopwords
```

Figure 5: Removal of Stop Words

Data Preprocessing - Tokenization

For easier interpretation I split the sentences into tokens using the nltk.word_tokeinze function.

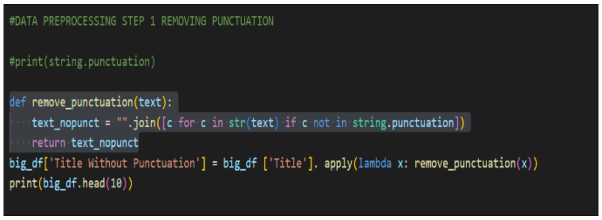


```
#CreatING a tokenizer using NLTK
def tokenize(column):


    tokens = nltk.word_tokenize(column)
    return [w for w in tokens if w.isalpha()]

#Tokenizing our text data using NLTK
big_df['tokenized'] = big_df.apply(lambda x: tokenize(x['Title Without NAN Values']), axis=1)
print(big_df.head(20))
```

Figure 6: Tokenization

Data Preprocessing - Stemming

Another data preprocessing technique that was utilized is Stemming which is a natural language processing technique used to extract the base word form of the words by removing affixes from them, it is just like cutting down branches of a tree to its stems. The PorterStemmer and SnowballStemm algorithm were compared within the NLTK package, but they approximately generated the same output.

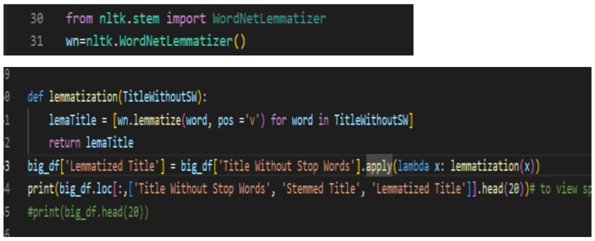


```
#DATA PREPROCESSING STEP 1 REMOVING PUNCTUATION


#print(string.punctuation)


def remove_punctuation(text):
    text_nopunct = "".join([c for c in str(text) if c not in string.punctuation])
    return text_nopunct
big_df['Title Without Punctuation'] = big_df ['Title']. apply(lambda x: remove_punctuation(x))
print(big_df.head(10))
```

Figure 7: Stemming

Data Preprocessing – Lemmatization

The last data preprocessing technique used is Lemmatization. This method was used to normalize the data, it is a similar technique to stemming but with the focus on finding a valid word. The output after lemmatization is called a lemma which is the root word. To implement the lemmatization technique, the WordNetLemmatizer class provided by the NLTK package was utilized.



```
from nltk.stem import WordNetLemmatizer
wn=nltk.WordNetLemmatizer()
```

```

def lemmatization(TitleWithoutSW):
    lemaTitle = [wn.lemmatize(word, pos ='v') for word in TitleWithoutSW]
    return lemaTitle
big_df['Lemmatized Title'] = big_df['Title Without Stop Words'].apply(lambda x: lemmatization(x))
print(big_df.loc[:,['Title Without Stop Words', 'Stemmed Title', 'Lemmatized Title']].head(20))# to view sp
#print(big_df.head(20))

```

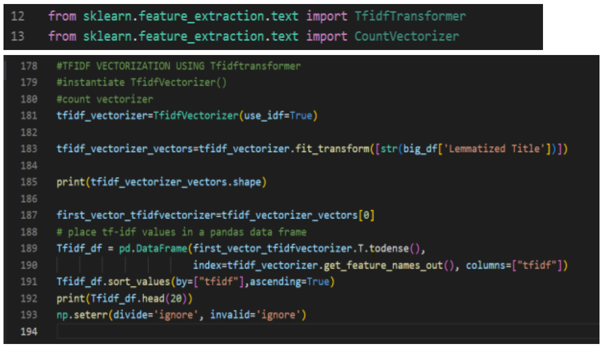
Figure 8: Lemmatization

Term Frequency Inverse Document Frequency (TFIDF)

After cleaning the dataset, the first text analysis we proceeded to perform the Term Frequency Inverse Document Frequency (TFIDF) Vectorization by importing scikit-learn python library and using sklearn.feature_extraction.text.TfidfVectorizer to convert the collection of raw documents to a matrix of TF-IDF features. Machine learning algorithms often use numerical data, so when dealing with textual data or any (NLP) task, the data first needs to be converted to a vector of numerical data by a process known as vectorization(Kumar & Subba, 2020).

TF-IDF vectorization involves calculating the TF-IDF score for every word in your corpus relative to that document and then putting that information into a vector. TF-IDF can be broken down into two parts TF (term frequency) and IDF (inverse document frequency). Term frequency works by looking at the frequency of a particular term you are concerned with relative to the document. Inverse document frequency looks at how common (or uncommon) a word is amongst the corpus. IDF also helps with filtering stopwords like "of", "as", "the", etc. since they appear frequently in an English corpus. Thus, by taking inverse document frequency, the weighting of frequent terms while making infrequent terms have a higher impact can be minimized(Irawaty et al., 2020).

To summarize the key intuition motivating TF-IDF is the importance of a term is inversely related to its frequency across documents.TF gives us information on how often a term appears in a document and IDF gives us information about the relative rarity of a term in the collection of documents. TF-IDF can be a very handy metric for determining how important a term is in a document or dataset.



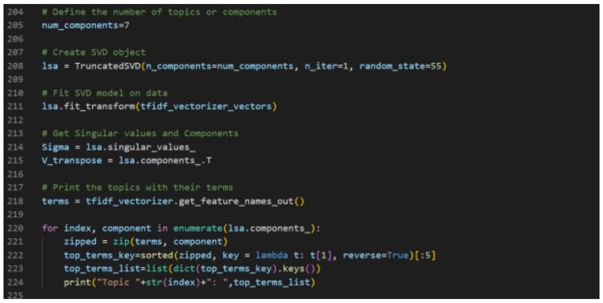
```
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer
```

```
#TFIDF VECTORIZATION USING Tfidftransformer
#instantiate TfidfVectorizer()
#count vectorizer
tfidf_vectorizer=TfidfVectorizer(use_idf=True)

tfidf_vectorizer_vectors=tfidf_vectorizer.fit_transform([str(big_df['Lemmatized Title'])])

print(tfidf_vectorizer_vectors.shape)

first_vector_tfidfvectorizer=tfidf_vectorizer_vectors[0]
# place tf-idf values in a pandas data frame
Tfidf_df = pd.DataFrame(first_vector_tfidfvectorizer.T.todense(),
                        index=tfidf_vectorizer.get_feature_names_out(), columns=["tfidf"])
Tfidf_df.sort_values(by=["tfidf"],ascending=True)
print(Tfidf_df.head(20))
np.seterr(divide='ignore', invalid='ignore')

```

Figure 9: TFIDF

Latent Semantic Analysis

Latent Semantic Analysis (LSA), also known as Latent Semantic Indexing (LSI), is a fully automated unsupervised statistical-algebraic summarization strategy that employs an extractive method to analyze texts and uncover hidden semantic relationships between the text's words and sentences. It generates a collection of concepts by conducting an analysis of the relationship between a group of terms and the documents that contain them. A machine learning approach called Singular Value Decomposition, or SVD, is used by LSA to attempt to extract the dimensions.



```python
# Define the number of topics or components
num_components=7

# Create SVD object
lsa = TruncatedSVD(n_components=num_components, n_iter=1, random_state=55)

# Fit SVD model on data
lsa.fit_transform(tfidf_vectorizer_vectors)

# Get Singular values and Components
Sigma = lsa.singular_values_
V_transpose = lsa.components_.T

# Print the topics with their terms
terms = tfidf_vectorizer.get_feature_names_out()

for index, component in enumerate(lsa.components_):
    zipped = zip(terms, component)
    top_terms_key=sorted(zipped, key = lambda t: t[1], reverse=True)[:5]
    top_terms_list=list(dict(top_terms_key).keys())
    print("Topic "+str(index)+": ",top_terms_list)
```

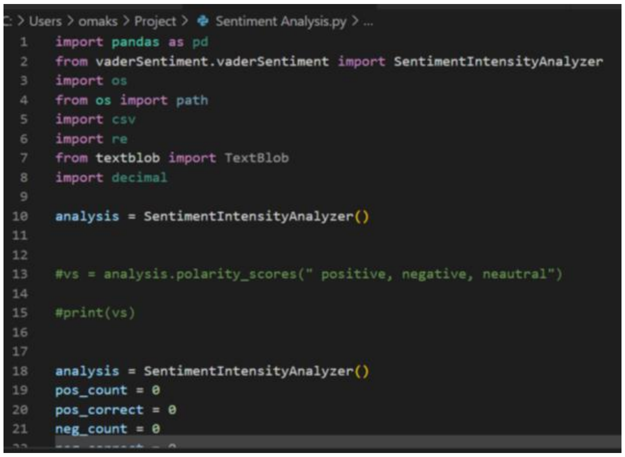
Figure 10: LSA

Sentiment Analysis

To understand the tone and how members of the individual subreddits feel about the discussions on the subreddits we performed a Sentiment analysis, also known as opinion mining, is a technique used in natural language processing (NLP) to determine the emotional undertone of a document. This is a common method used by companies to identify and group opinions about a given good, service, or concept. Data mining, machine learning, artificial intelligence, and computational linguistics are all used in sentiment analysis to sort through text for sentiment and subjective information, such as whether it's expressing positive, negative, or neutral sentiments.

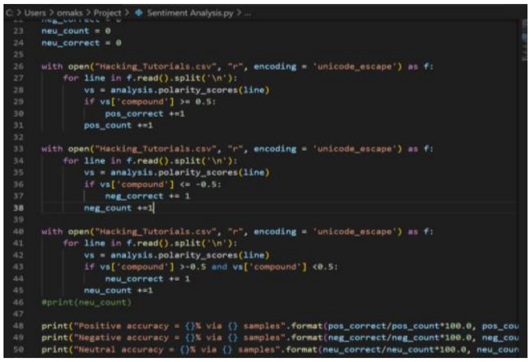
VADER (Valence Aware Dictionary for Sentiment Reasoning) an NLTK module that provides sentiment scores based on the words used was utilized for the implementation of sentiment analysis. It is a rule-based sentiment analyzer in which the terms are generally labeled as per their semantic orientation as either positive, neutral, or negative (Elbagir & Yang, 2019).

The sentiment was ascertained using the polarity scores approach, the preprocessed top posts were categorized into positive, neutral, and negative using the VADER Sentiment Analyzer. A good metric for assessing the sentiment in a particular top post is the compound value. The threshold values in the suggested approach are used to classify the top post as either good, negative, or neutral.



```python
import pandas as pd
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import os
from os import path
import csv
import re
from textblob import TextBlob
import decimal

analysis = SentimentIntensityAnalyzer()


#vs = analysis.polarity_scores(" positive, negative, neautral")

#print(vs)


analysis = SentimentIntensityAnalyzer()
pos_count = 0
pos_correct = 0
neg_count = 0
```

Figure 11: Importing the necessary Sentiment Analysis Library



```python
neu_count = 0
neu_correct = 0

with open("Hacking_Tutorials.csv", "r", encoding = 'unicode_escape') as f:
    for line in f.read().split('\n'):
        vs = analysis.polarity_scores(line)
        if vs['compound'] >= 0.5:
            pos_correct +=1
        pos_count +=1

with open("Hacking_Tutorials.csv", "r", encoding = 'unicode_escape') as f:
    for line in f.read().split('\n'):
        vs = analysis.polarity_scores(line)
        if vs['compound'] <= -0.5:
            neg_correct += 1
        neg_count +=1

with open("Hacking_Tutorials.csv", "r", encoding = 'unicode_escape') as f:
    for line in f.read().split('\n'):
        vs = analysis.polarity_scores(line)
        if vs['compound'] >-0.5 and vs['compound'] <0.5:
            neu_correct += 1
        neu_count +=1
#print(neu_count)

print("Positive accuracy = {}% via {} samples".format(pos_correct/pos_count*100.0, pos_cou
print("Negative accuracy = {}% via {} samples".format(neg_correct/neg_count*100.0, neg_cou
print("Neutral accuracy = {}% via {} samples".format(neu_correct/neu_count*100.0, neu_coun
```

Figure 12: Sentiment Analysis Using VADER

5. EXPERIMENTAL RESULTS

The more important or relevant a word is, the higher its TF-IDF score. Table 1 shows the results of the most significant words for the Hacking Tutorial Subreddit. "Comptia" which is a cybersecurity certification had the highest score of 0.251976. Table 2 presents the TFIDF results for Hacking subreddit which indicate that "Hack" was the most significant with the TFIDF value of 0.353553. Likewise, Table 3 shows the results for the Ethical Hacking Subreddit which indicates that "hacking", and "ethical" were the most significant words.

Word	TFIDF Score
Banned	0.125988
book	0.125988
Bounty	0.125988
Box	0.125988
Brilliant	0.125988
Bug	0.125988
Community	0.125988
Comptia	0.251976
D-type	0.125988
Escape	0.125988

Table 1: TFIDF Results for Hacking Tutorial Subreddit – Top 10 Words

Word	TFIDF Score
Dutch	0.117851
Ethical	0.117851
E-ticket	0.117851
Every	0.117851
Exploit	0.117851
Folina Flaw	0.117851
Free	0.117851
Government	0.117851
Hack	0.353553
Hacker	0.117851

Table 2: TFIDF Results for Hacking Subreddits – Top 10 Words

Word	TFIDF Score
Ethical	0.235702
Hacking	0.471405
Hangout	0.117851
Humble	0.117851
Interested	0.117851
Joining	0.117851
Length	0.117851
Life	0.117851
Message	0.117851
Mine	0.117851

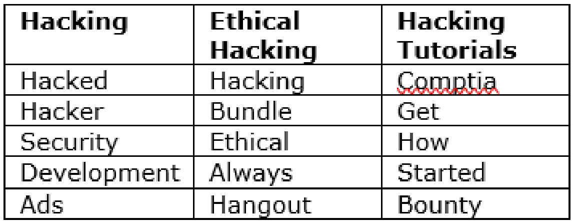
Table 3*:* TFIDF Results for Ethical Hacking Subreddit – Top 10 Words

Search engines are a frequent example of how TF-IDF is used in the field of information retrieval. A search engine can utilize TF-IDF to help rank search results based on relevance, with results that are more relevant to the user having higher TF-IDF scores. This is because TF-IDF can inform you about the relevant importance of a term based upon a document, this can be used to choose keywords (or even tags) for a document or to summarize articles more effectively.

Latent Semantic Analysis

To decompose every term and document as a vector the document-term matrix was used specifically the sklearn's Truncated SVD, because our data was from 3 different subreddits and to avoid overlapping topics we decided to have 3 topics for our text data though the number of topics can be specified using the n_components parameter. Our LSA result indicates the 3 most frequently discussed topics in the subreddits are certifications, techniques on how to do ethical hacking, and training tools. LSA in comparison to the vector space model provides better outcomes, being limited to document term matrix decomposition makes LSA faster than other available techniques.

Table 4: Comparison of LSA Topics Discussed in the Subreddits



Hacking	Ethical Hacking	Hacking Tutorials
Hacked	Hacking	Comptia
Hacker	Bundle	Get
Security	Ethical	How
Development	Always	Started
Ads	Hangout	Bounty

Semantic Analysis

This section discusses the findings of a sentiment analysis on the different hacking subreddits conducted using the NLTK and VADER sentiment analysis tools. As determined by the VADER Sentiment Analyzer, Figures 13, 14 & 15 display the sentiment percentage of the top posts in the individual subreddits as positive, negative, or neutral.

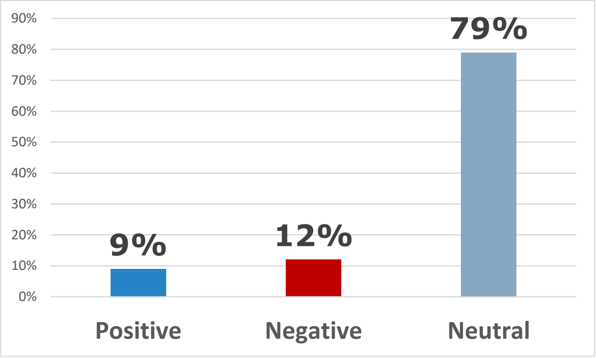


Figure 13 Hacking Subreddit Top Posts

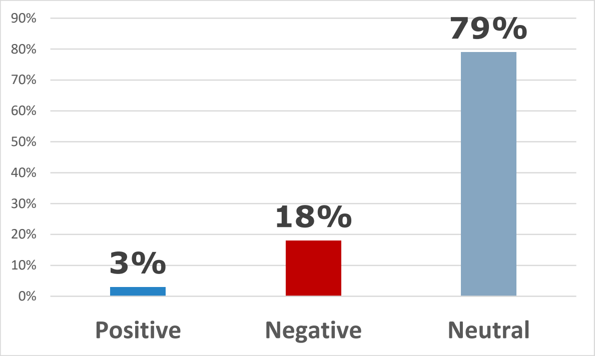


Figure 14 Hacking Tutorial Subreddit Top Posts

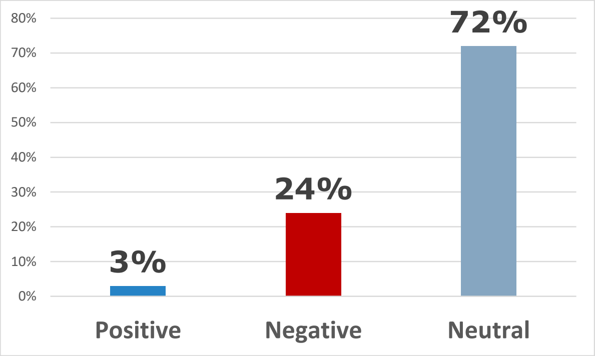


Figure 15 Ethical Hacking Subreddit Top Posts

According to the Base Polarity which is the standard scale, there were more Neutral than positive and a little fewer negative sentiment from Top posts in the subreddits data was collected from. 79% of Hacking subreddit top posts expressed neutral, 12% had negative and 9% expressed positive opinion, while in the ethical hacking subreddit 73% of the top post were neutral, 24% were positive and 3% had negative sentiments. Finally, the hacking tutorial results also indicate 79% of the top posts displayed neutral opinion, 18% were positive and 3% were negative. Based off these results we were able to confidently deduce and conclude that the hacking subreddit top posts analyzed are fact or knowledge based and less subjective as not a significant percentage of emotions and polarity is indicated in the results.

6. CONCLUSION AND FUTURE RESEARCH

To fully understand the most discussed topics in subreddits that pertain to hacking, we investigated top posts on hacking subreddits based on data collected from multiple hacking subreddits. Then we preprocessed the data by eliminating punctuation, stop words, tokenization additionally, using the stemming and lemmatization technique, all derived words in the documents are returned to their stem or base word and lemma form.

We conducted a Term Frequency and Inverse Document Frequency of the corpus which indicated "hack", "hacking" and "Comptia" were more relevant terms in our data according to their TFIDF scores. We conducted a latent sematic analysis which was utilized to analyze the association of words in the same context. We also performed sentiment analysis on the data from all 3 subreddits using the VADER algorithm, with the results indicating that there were more neutral sentiments than positive and a little fewer negative sentiment on the Top posts.

To sum up, the results of the frequency analysis highlight important topics of conversation in the context of cybersecurity. Specifically, the subjects that received the most extensive discussion involve Certifications, Techniques related to ethical hacking, and the usage of training tools. These findings illuminate significant areas of focus and exploration among individuals, showcasing a notable emphasis on improving expertise, understanding, and qualifications in the field of cybersecurity. Notwithstanding these findings, it would be helpful if future research were to examine other cybersecurity subbreddits such as r/malware, r/netsec, and r/cybersecurity to see if the same topics and more importantly same sentiment analysis appear.

7. REFERENCES

Boe., B. (2023). https://praw.readthedocs.io/en/stable/getting_started/installation.html

Carnot, M. L., Bernardino, J., Laranjeiro, N., & Gonçalo Oliveira, H. (2020). Applying text analytics for studying research trends in dependability. *Entropy, 22*(11), 1303. https://doi.org/10.3390/e22111303

Chiny, M., Chihab, M., Bencharef, O., & Chihab, Y. (2022). Netflix Recommendation System based on TF-IDF and Cosine Similarity Algorithms. *no. Bml*, 15-20. https://doi.org/10.1080/08839514.2023.2175112

Choo, S., & Kim, W. (2023). A study on the evaluation of tokenizer performance in natural language processing. *Applied*

Artificial Intelligence, 37(1), 2175112. https://doi.org/10.1080/08839514.2023.2175112

Dive Into Anything. (2023). https://www.redditinc.com/

Elbagir, S., & Yang, J. (2019). Twitter sentiment analysis using natural language toolkit and VADER sentiment. Proceedings of the international multiconference of engineers and computer scientists.

Giri, S., & Banerjee, S. (2023). Performance analysis of annotation detection techniques for cyber-bullying messages using word-embedded deep neural networks. *Social Network Analysis and Mining, 13*(1), 1-12. https://doi.org/10.1007/s13278-022-01023-2

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science, 349*(6245), 261-266. https://doi.org/10.32604/cmc.2022.024190

Huyut, M. M., Kocaoğlu, B., & Meram, U. (2022). Regulation Relatedness Map Creation Method with Latent Semantic Analysis. *Computers, Materials and Continua*.

IBM. (2023). *What is natural language processing (NLP)?* Retrieved 04/24/2023 from https://www.ibm.com/topics/natural-language-processing#:~:text=Natural%20language%20processing%20(NLP)%20refers,same%20way%20human%20beings%20can.

Irawaty, I., Andreswari, R., & Pramesti, D. (2020). Vectorizer comparison for sentiment analysis on social media youtube: A case study. 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE), DOI: 10.1109/IC2IE50715.2020.9274650

Kumar, V., & Subba, B. (2020). A TfidfVectorizer and SVM based sentiment analysis framework for text data corpus. 2020 National Conference on Communications (NCC), DOI: 10.1109/NCC48643.2020.9056085

Liang, M., & Niu, T. (2022). Research on Text Classification Techniques Based on Improved TF-IDF Algorithm and LSTM Inputs. *Procedia Computer Science, 208*, 460-470. https://doi.org/10.1016/j.procs.2022.10.064

Mayopu, R. G., Wang, Y.-Y., & Chen, L.-S. (2023). Analyzing Online Fake News Using Latent Semantic Analysis: Case of USA Election Campaign. *Big Data and Cognitive Computing, 7*(2), 81. https://doi.org/10.3390/bdcc7020081

Natural Language Toolkit. (2023). https://www.nltk.org/

Pramana, R., Subroto, J. J., & Gunawan, A. A. S. (2022). Systematic Literature Review of Stemming and Lemmatization Performance for Sentence Similarity. 2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA), DOI: 10.1109/ICITDA55840.2022.9971451

Prasanth, S., Raj, R. A., Adhithan, P., Premjith, B., & Kp, S. (2022). CEN-Tamil@ DravidianLangTech-ACL2022: Abusive Comment detection in Tamil using TF-IDF and Random Kitchen Sink Algorithm. Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages, https://doi.org/10.18653/v1/2022.dravidianlangtech-1.11

Rahimi, Z., & Homayounpour, M. M. (2023). The impact of preprocessing on word embedding quality: A comparative study. *Language Resources and Evaluation, 57*(1), 257-291. https://doi.org/10.1007/s10579-022-09620-5

Ramezani, M., Shahryari, M.-S., Feizi-Derakhshi, A.-R., & Feizi-Derakhshi, M.-R. (2023). Unsupervised Broadcast News Summarization; a comparative study on Maximal Marginal Relevance (MMR) and Latent Semantic Analysis (LSA). *arXiv preprint arXiv:2301.02284*. https://doi.org/10.1109/CSICC58665.2023.10105403

Subba, B., & Gupta, P. (2021). A tfidfvectorizer and singular value decomposition based host intrusion detection system framework for detecting anomalous system processes. *Computers & Security, 100*, 102084. https://doi.org/10.1016/j.cose.2020.102084

Wagire, A. A., Rathore, A., & Jain, R. (2020). Analysis and synthesis of Industry 4.0 research landscape: Using latent semantic analysis approach. *Journal of Manufacturing Technology Management, 31*(1), 31-51. https://doi.org/10.1108/JMTM-10-2018-0349