UNIVERSITY OF NAIROBI

FACULTY OF SCIENCE AND TECHNOLOGY

COMPUTER SCIENCE DEPARTMENT

AUTOMATED CYBERSECURITY BRIEFING USING DEEP LEARNING

By:

Josech Mayaka

P52/37694/2020

Email: josechmayaka193@gmail.com

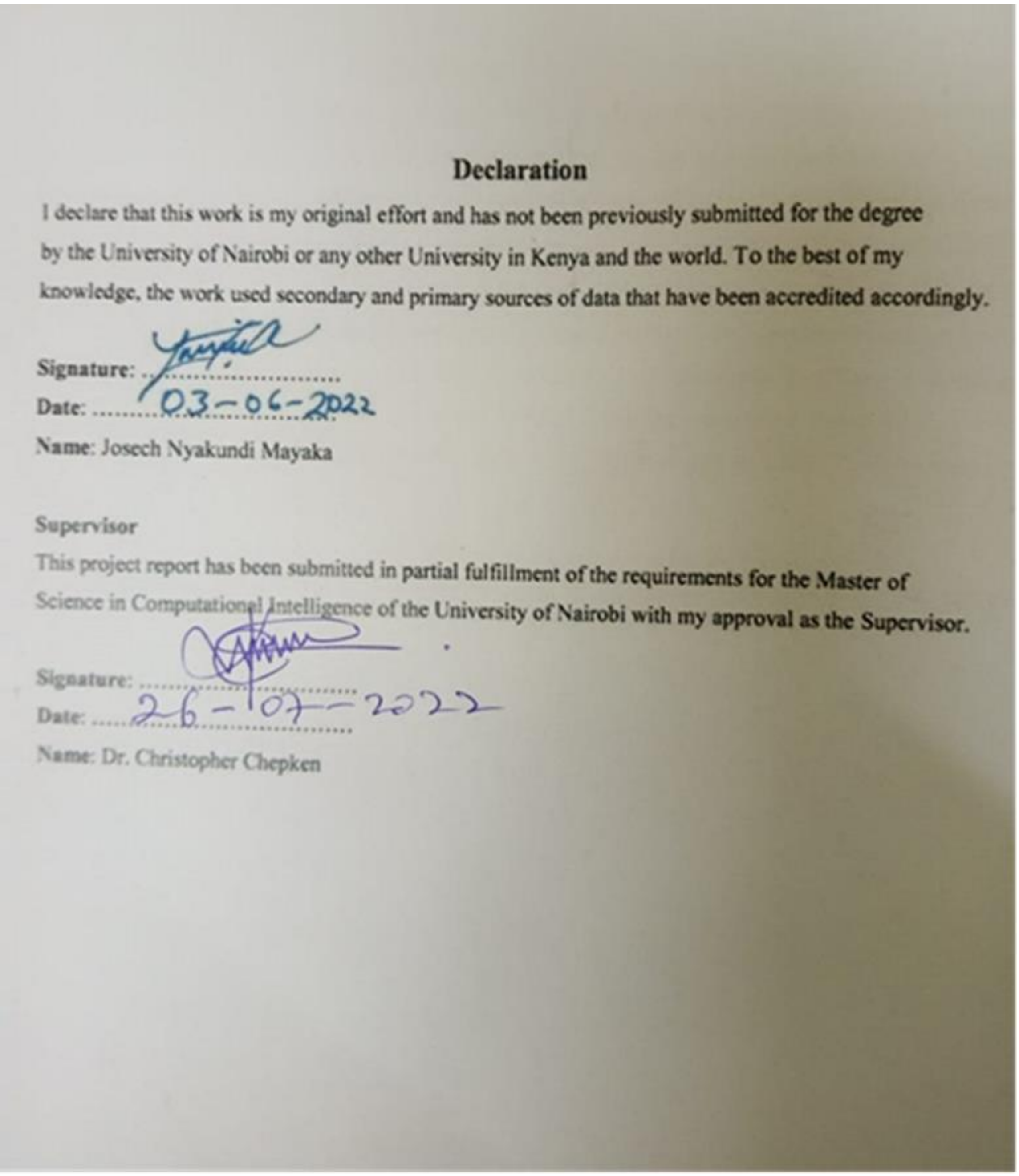
Supervisor:

Dr. Chepken Christopher

Email: chepken@uonbi.ac.ke

JUNE 2022

A research project report submitted in partial fulfillment of the requirements for the Degree of Masters of Science in Computational Intelligence of the University of Nairobi, Kenya



Declaration

I declare that this work is my original effort and has not been previously submitted for the degree by the University of Nairobi or any other University in Kenya and the world. To the best of my knowledge, the work used secondary and primary sources of data that have been accredited accordingly.

Signature: ....

Date: 03-06-2022

Name: Josech Nyakundi Mayaka

Supervisor

This project report has been submitted in partial fulfillment of the requirements for the Master of Science in Computational Intelligence of the University of Nairobi with my approval as the Supervisor.

Signature: ....

Date: 26-07-2022

Name: Dr. Christopher Chepken

ABSTRACT

The urgency and timely requirements of cybersecurity briefings poses a challenge to a few cybersecurity professionals who have to read and summarize vast amount of cybersecurity reports from several sources (personal communication, October 26, 2021). This paper demonstrates a solution based on Long Short Term Memory that automates the process of generating briefs from various cybersecurity report sources and further assesses the standardly used metric(ROUGE) for summary evaluation. This was achieved through the use of CRISP-DM methodology and application of the natural language processing techniques. After training and testing the model, it outperformed other summarizers such as lexRank. Abstractive technique is considered to be relatively strong and dynamic, because sentences that form summaries are generated based on their semantic meaning. On assessing various ROUGE variants, it was clear that evaluating specific summaries require different ROUGE metrics. For instance, ROUGE-1 and ROUGE-2 may be useful if you're working on extractive summarization.

Keywords: Cybersecurity Briefing, Recurrent Neural Network, Long Short Term Memory, Abstractive Summary, Extractive Summary, ROUGE, ROUGE-AR.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ROUGE	-	Recall-Oriented Understudy for Gisting Evaluation
RNN	-	Recurrent Neural Network
LSTM	-	Long Short Term Memory
CRISP-DM	-	Cross-Industry Standard Process for Data Mining
ACB	-	Automated Cybersecurity Briefing
ICT	-	Information Communication Technology
ROUGE-AR	-	Recall Oriented Understudy Gisting Evaluation - Anaphor Resolution
GOOGLE COLAB	-	Google Colaboratory

1. INTRODUCTION

1.1 Background Study

In today's computerized world, there is the emergence of new cyber threats and risks every minute, amounting to several threats and risks within a single day that are propagated within the global cyberspace. The growing number of devices connecting to the internet widens the cyber threat landscape and thereby by increasing the chances of successful attacks. Quick responses that are supported by cybersecurity briefs that aid in making effective and well informed strategic decisions are needed (personal communication, August 10, 2021). Cybercrime has now become a big business risk for both organizations and Nation states globally. The need for automated summary generation through text summarization in industries such as security is becoming inevitable due to voluminous nature/amount of reports received on a daily basis that require briefing.

The cybersecurity information that resides in many online sources such as cybersecurity vendor bulletins, peer forum posts, cyber threat information sharing platforms, cybersecurity blogs and various databases forms a significant portion of information sources for cybersecurity analysis. Moreover, Cybersecurity security analysts depend on these documents for understanding their assets' vulnerabilities, prioritizing patches, tracing clues during forensic efforts, and understanding emerging threats(Bridges et al., 2017).

The cybersecurity skills gap in Kenya has delayed the efforts of adequately addressing cyber-attacks. The few cybersecurity professionals available are sometimes overworked and do not find enough time to concentrate on key areas when addressing the cybersecurity challenges. A survey conducted by Tripwire early 2020 revealed that 83% of cybersecurity professionals felt overworked(*Survey: Only 39% of Orgs Have Ability to Retain Cyber Security Talent*, n.d.). Cybersecurity reporting is time bound and can only be effective if a report is submitted on time. A cybersecurity statistics report released on March 2021 by Forbes and Purplesec revealed that 230,000 new malware samples, 100,000 malicious websites and 10, 000 new malicious files are produced daily.

Most of the technological/ICT solutions affected are shared globally and therefore every cybersecurity professional from any part of the world should be concerned about any cyber threat being propagated in the global cyberspace. As a result, it is nearly impossible for these professionals to manually go through these hundreds and thousands of reports in order to gain insights and come up with briefs that can necessitate the next cause of action.

A quick and effective response to cyber-incidents should include automation of repetitive tasks such as report summarization to generate cyber threat briefs, that support and guide in prompt and quick decision making process. Autonomous cybersecurity report summarization takes the burden off the security team, so that they have enough time to focus more on addressing cyber-attacks.

1.2 Problem of Research

(i) Voluminous manual data correlation. It’s not convenient and practical to manually summarize thousands of cybersecurity reports generated daily in order to come up with an effective and actionable brief within the limited time.

(ii) Delayed Cybersecurity Response Time. Taking away the burden of manually performing repetitive tasks and minimize the Remediation Mean Time through human machine collaboration that leads to improved productivity, increased capacity and reduced risk.

(iii) Shortage of Cybersecurity skill gap. Enhance Cyber defense efficiency by saving manpower as this allows faster prevention of new and unknown threats through automation of repetitive tasks.

1.3 Problem Definition

The increasing lengthy and voluminous cybersecurity reports produced daily is becoming harder to generate meaningful and timely cybersecurity briefs(Dr. Emily Hand, n.d.).The few cybersecurity professional working in this industry are overwhelmed with the number of reports they have to sift through on a daily basis so as to identify a malicious activity or a potential cyber threat activity that can impact their organization or state. As a result, the intention of this research was to make this more proactive and easier by automating the process of cybersecurity briefing through the application of Natural language processing techniques.

By automating this repetitive process, the cybersecurity professional will have free time to pay more attention to other compelling tasks(emergencies) such as handling cyber incidents. Scenarios such as these(emergencies), the time required to prepare cybersecurity briefs for strategic decision making is limited.

1.4 Research Questions

(i) How do you establish a baseline for evaluating automatic text summarization process?

(ii) Can any ROUGE variants be used to evaluate both abstractive and extractive summaries?

(iii) Is ROUGE the best method for evaluating highly paraphrased summaries?

(iv) Is LSTM better than lexRank in automated cybersecurity briefing through text summarization process?

1.5 Objectives

1.5.1 Overall goal

To automate cybersecurity brief generation in order to support time bound strategic decision making process.

1.5.2 Specific research Objectives

(i) To apply natural language processing techniques in generating automated cybersecurity briefs in order to speed up strategic decision making process.

(ii) To examine the most effective ways of automating and improving cybersecurity text summarization.

(iii) To assess the variants of the standardly used summarization evaluation metrics and come up with deductions based on the assessment.

1.6 Scope

This research focuses on summarization of reports within the cybersecurity industry. The cybersecurity reports source coverage includes: cybersecurity vendor bulletins, peer forum posts, cyber threat information sharing platforms, cybersecurity analytical reports.

1.7 Significance

This paper is relevant to Smart Africa Agenda on Cybersecurity and Big Data analytics, specifically in helping governments to prevent or proactively deter crime, boost National security by protecting critical ICT infrastructures and enhance the level of cybersecurity awareness(Smart Africa, 2018).

This research also contributes to the achievement of Smart Africa Agenda in the Kenyan context by promoting the implementation of the Computer Misuse and Cybercrime Act 2018, Part III section 40 on reporting of cyber threats.

2. LITERATURE REVIEW

2.1 Introduction

The development and generation of cybersecurity reports is growing day to day . The few cybersecurity analysts are overwhelmed since they have to be vigilant, analyze and consider all cybersecurity reports shared on the global cyberspace to come up with advisory briefs that are used on a daily basis for cyber defense.The need to generate accurate summaries(briefs) in a short period without losing the meaning of the original text is becoming inevitable.

This prompted the review of the past research work related to text summarization especially in the cybersecurity industry.This paper identified and analyzed methods, datasets and trends in automatic text summarization research from 2013 todate.Much attention was given to research papers that talked about text summarization in cybersecurity. The literature reviews below are arranged in the order of relevance.

2.2 Related Work

2.2.1 Interactive Summarization for Data Filtering and Triage

(Robertson et al., 2020) Claims that there is an increasing demand for content filtering and flagging on the social media platforms in relation to cybersecurity.This work proposes a two novel perspectives on this problem.They propose utilization of topic-based summarization algorithms and topic conditioning approach to facilitate multiple summarization based on different highlghted topics.Its also demonstrated how this approach can be integrated within the process of a human analyst to improve both the quality of filtered data and the efforts.

2.2.2 Extracting Rich Semantic Information about Cybersecurity Events

(Satyapanich et al., 2019) Proposes that semantic schemas can be used to decribe cybersecurity events. He further states that using news articles anotated with these types of events, they detail a deep learning based infromation extraction which mines useful data with high accuracy. It‘s said that the cybersecurity event set considered can also enable the extension of news event types such as Denial of service.

2.2.3 A Hybrid Approach for Multi-document Text Summarization

(Sidhpurwala et al., 2020) This work by Sidhpurwala explores how the application of reduction algorithm, Text Rank, and Latent Semantic Analysis for summarization can be optimized and compares it with the approach proposed that creates a hybrid system that consolidates all the mentioned algorithms.

2.2.4 Distilling Public Data from Multiple Sources for Cybersecurity Applications

(Dr. Emily Hand, 2020) The main aim of this work was to demonstrate how publicly available data from multiple sources can be utilized to create cybersecurity applications that will assist in defending against cyber threats. This was illustrated through training a text summarizer tool that aided in digesting cybersecurity articles and data from various social media platforms in identification of bot accounts or fake users.

2.2.5 A Vietnamese based supervised learning text summarizer

(Thu & Huu, 2013) The major objective of this work is to demonstrate how combining reducing features with neural network for learning in text summarization could effectively reduce computational complexity.

2.2.6 Query-oriented text summarization

(Fors-Isalguez et al., 2018) Fors-Isalguez proposes a method for query oriented summarization that takes the multi-objective optimization problem approach with Pareto front consideration based on sentence embedding representation. The method was evaluated using the TAC dataset, with the results obtained showing a contribution to improved performance significantly.

2.2.7 An Automatic Multidocument Text Summarization Approach Based on Naïve Bayesian Classifier

(Ramanujam & Kaliappan, 2016) This research aimed at introducing a new concept on time step method for multi-document text summarization using Naïve Bayesian Classification approach. The overall aim was to produce a coherent looking summary. This was achieved through extracting more important information from various documents ingested, using scoring strategy to calculate word/term frequency. In order to illustrate the efficiency of the proposed approach, comparison between the proposed method and the existing MEAD approach was done. The results showed that the proposed approach performed better in precision, recall and f-score in text summarization process than the existing clustering approach.

2.2.8 INSHORTS

(Heckman et al., 1967) It's a news aggregator and discovery platform that summarizes large text into just 60 words. It does this through the use of Rapid 60 which is an AI-backed algorithm that automatically summarizes text of news articles to 60 words. When articles are feed into Inshort, the algorithm generates shorts of 60 words with the headline and the card image automatically.

2.2.9 A Scalable Summarization System Using Robust NLP

(Prabhala, 2014) This tool assists in summarizing content from various textual sources with the capability of recognizing the main topics being discussed. It the extracts and analyzes texts automatically considering the most used, and important words and expressions in the texts.

2.2.10 The Automatic Creation of Literature Abstracts

This study was done by H.P.Luhn where he employed the use of word collections and term frequency in the automated creation of litarure abstracts [13]. The objective was to generate generic abstracts from various research papers. However, this approach had limitations in that it could only handle single documents with less than four thousand words in total.

2.2.11 Summarize Bot

This is AI-enabled and block chain tool to learn more with a little reading of long text summaries. This bot includes white papers, web pages, images and even audio data. It helps users to save reading time during research by shortening the lengthy text. Additionally, it also removes keywords and blocks issues within text, and allows users to customize the lengthy of summary as well.

2.2.12 Resoomer

This tool produces summaries of texts and allows sorting of documents on important topics by intifying crucial facts and ideas quckly.Users just need to copy and paste the text they wish to summarize in order to get the original text compressed to just 500 words.

2.2.13 Google's news aggregator

The Google news aggregator is a news summarization platform that encompasses more than 20,000 publishers.The summary constitues roughly the first 200 characters of the main article and then links it ot the larger content.

2.2.14 Flipboard

It's a web based news aggregator that takes content from news sources including social media and then summarizes it to a personalized digital magazine, and lets users flip through it.

2.2.15 Copernic summarizer

This tool is capable of analyzing a text of any length in any of four languages and create a document summary as short as may be required. This tool has also the capability of summarizing email messages, html files, pdf and Microsoft word documents.

2.2.16 Intellexer Summarizer

This tool takes a document or a set of documents as the input and produces outputs a shorter document that contains the main important content and ideas. Intellexer's unique feature is the ability to create different kinds of summaries i.e. theme-oriented that produces summaries relevant to a given topic, structure-oriented that generates summary content depending on the input document structure and concept-oriented that produces a summary with respect to a number of user defined concepts that elevates the sentence importance.

2.2.17 Assessing the accuracy of automated text summarization evaluation

This work by Karolina Owczarzak discusses and compares the text summarization evaluation metrics, with a key focus on the accuracy and performance of these metrics. It's said that there is mistrust in the use of automated evaluation measures since their accuracy and correct application is not well understood. Its further stated that the evaluation measures relate with human judgement is too general and incomplete. This is because the same evaluation measure can be used to measure some manual evaluation scores for summarization job as well, giving poor correlation with manual scores for certain tasks (Lin,2004; Liu and Liu,2010).

2.2.18 Biomedical-domain pre-trained language model for extractive summarization

This work published by Yong-ping Du on July 2020, proposes a novel model known as BioBERTSum trained on biomedical corpora as encoder for extractive text summarization task on a single document.

2.2.19 HITS-based attentional neural model for abstractive summarization

On 1st March 2021, Xiaoyan Cai proposed a HITS based attention mechanism that takes advantage of the word and sentence level information to refine the attention value by considering the words from the original document as the authorities.

2.2.20 Leveraging Multimodality with Guided Attention for Abstractive Text

Yash Kumar on 20th May 2021 published his work that tries to address the poor performance of multimodal inputs to ensure adequate quality and uniformity in both textual and video summaries produced.

2.2.21 Deep contextualized embeddings for quantifying the informative content in biomedical text summarization

M. Moradi on 1st Feb 2020 proposed a model that leverages on the Biredirectional Encoder representations from the BERT transformer model to capture the context of the text in the summarization process. The study provided a starting point towards investigating the contextualization process of the biomedical text summarization.

2.2.22 Summary evaluation using ROUGE family

It’s the most widely and standardly used summary evaluation metric in determing the quality of system generated summaries (Lin, 2004b). It compares machine generated summaries against reference model summary (i.e. human generated summary). This is aimed at estimating whether the relevant concepts are covered in the automatically generated summaries. ROUGE has various variants which include ROUGE-1 to ROUGE-4 i.e. from uni-gram (single words) to four-gram (four words) which is based on overlap between system and reference summaries. The sequence of overlapping words that do not follow each other are computed using ROUGE-L.

2.3 Research Gap

Existing research has not exhaustively explored the contextualization and automation of text summarization process in cybersecurity industry. Although there have been many successes in text summarizations that focus on news aggregation through obtaining data sets, methods, and techniques for publication, not many papers can provide a comprehensive research of automated cybersecurity briefing through text summarization from various sources.

Despite the efforts by many researchers basing their work on improving extractive summarization with some shifting their focus on abstractive techniques, current summarizers are still far from perfect and challenges still remain unresolved(Patil, 2017). For instance, evaluation of summary results is a difficult task because there does not exist an ideal summary for a document(s) for evaluation.

The absence of a given human standard or automatic evaluation metric makes it very hard to compare different systems and establish a baseline(Neto et al., n.d.). As a result, the need to also improve methods of evaluating automatic text summarization and come up with more consistent evaluation methods will become essential.

2.4 System Design Architecture

Figure 1 below illustrates the automated cybersecurity brief generation process:

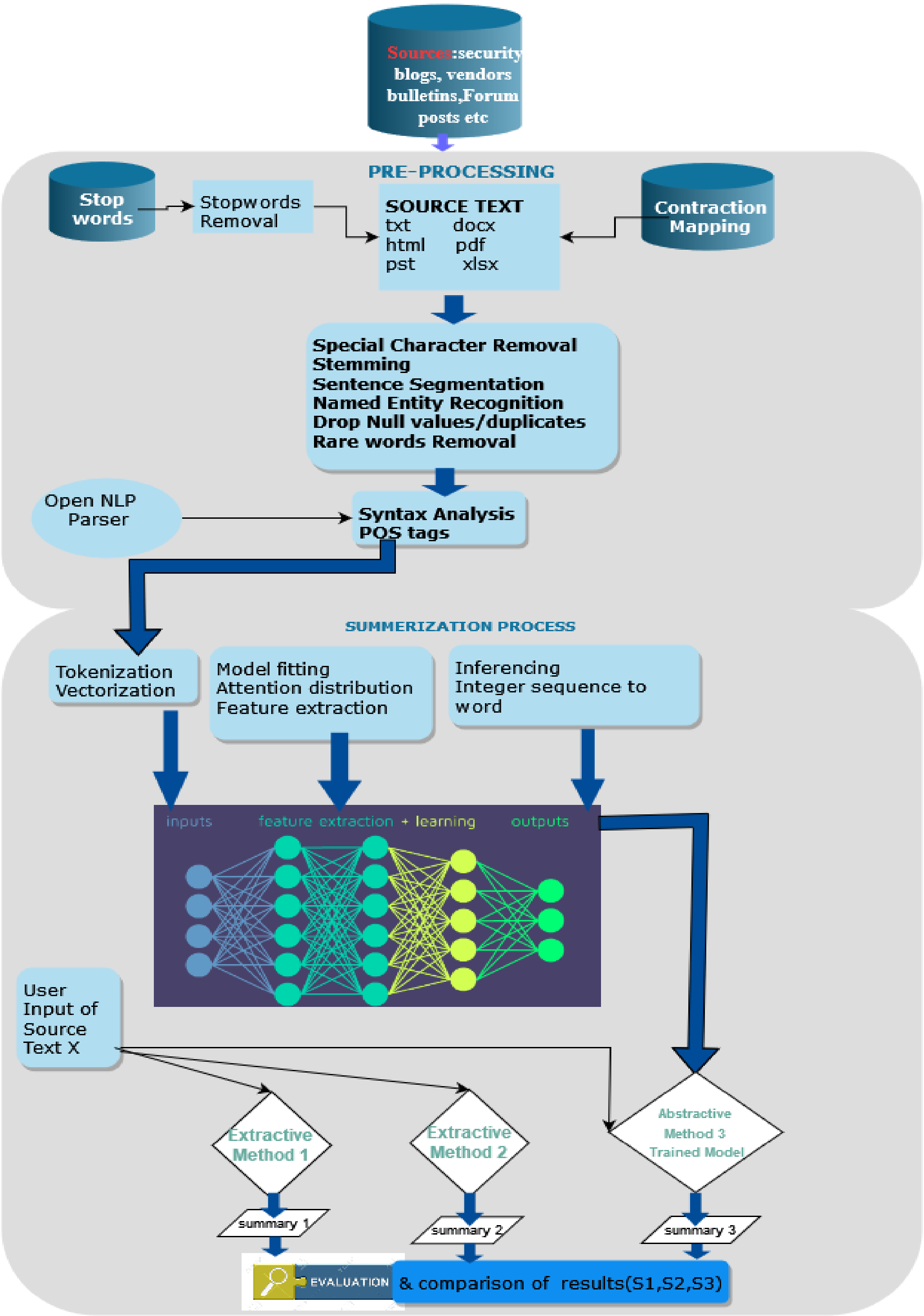


Figure 1:Automated Cybersecurity Briefing Design

3. METHODOLOGY

3.1 Introduction

In order to effectively meet the research objectives through answering research questions, a coherent process was used in which qualitative, quantitative and experimental research was applied. This is because the default calculation involved data modeling that made sense of extracted data through machine learning and statistical techniques, getting feedback from industry experts on the trained model after interacting with it and testing the model on real data/ unseen data. The Automated Cybersecurity Briefing Using Deep Learning generates a summary containing key sentences and includes all important key details from the original text. One of the main methods that were applied in summary generation is extracting and abstracting.

3.2 Methodology Overview

The methodology that was used in this project is the Cross-Industry Standard Process for Data Mining(CRISP-DM). The model entailed six phases that described the planning, organizing and implementing a machine learning data science project life cycle. The six phases include:

Business Understanding. This was the first phase that focused on understanding the project's objectives based on the requirements. At this stage, the business success criteria and feasibility study was done to determine the availability of resources in terms of data availability, project risks and contingencies.

Data Understanding. This was the second phase that focused on data identification, collection and analysis in order to accomplish the project's objectives. The data was collected and loaded into the analysis tool. The description of data was also added, including the properties among format, number of records and fields. The relationship was also established at this phase. Then finally, data was verified to determine whether it meets the requirement and passes the quality checks.

Data preparation. This was the third phase where data was prepared for final analysis. The first step of data preparation was collecting the data set that could help in achieving the research objectives. The data was then cleaned, loaded for correction and removal of unnecessary/less important features. Next, the data was then constructed by deriving new attributes and characteristics that helped in the analysis. Data was then integrated in cases where it was obtained from multiple sources and combined into one repository for use. The data was then reformatted, ready to be used in model building.

Modeling. This was the fourth phase where the selection of the modeling technique was done such as the algorithm used for data analysis. The splitting of data set into training, testing and validation set was done. The model was then built and assessed, then interpreted based on the previously set procedures.

Evaluation. The fifth phase involved evaluation of the model to find out whether it meets the business success criteria that was set at phase 1 of the methodology. A monitoring and maintenance plan was developed in order to reduce issues experienced during the operation phase. Finally, the final product was released as it underwent continuous review. The Overview of this methodology is presented in figure 2.

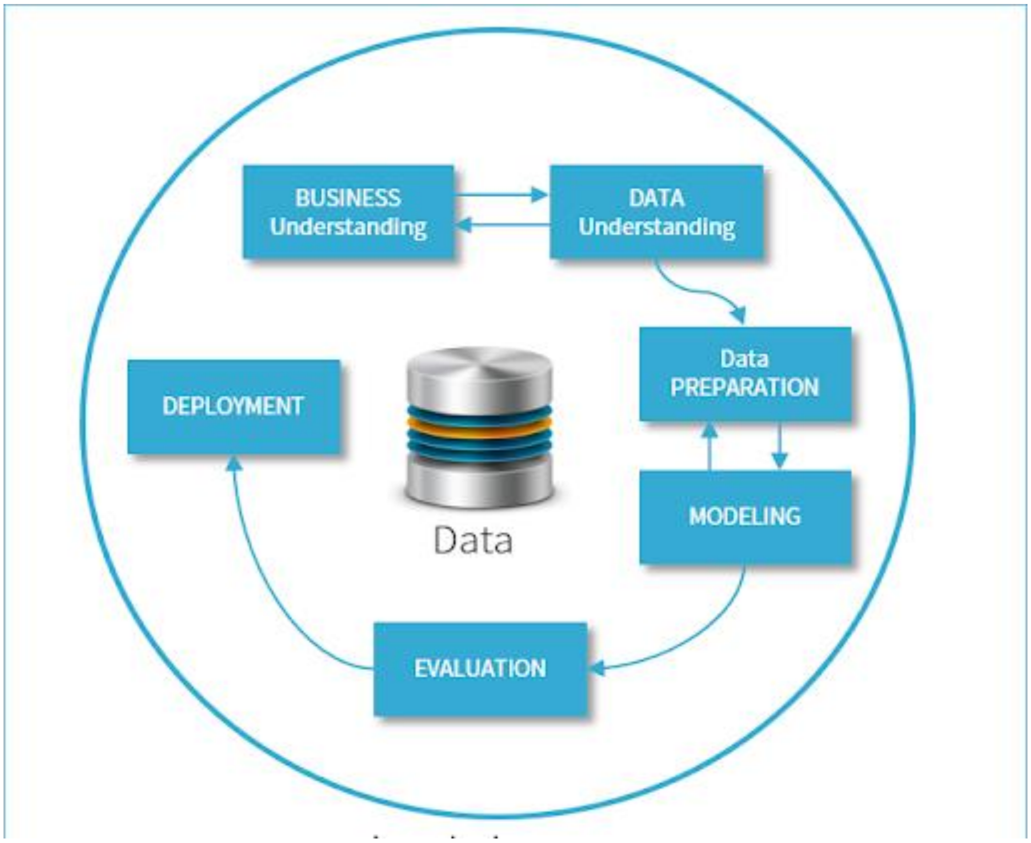


Figure 2:CRISP-DM Methodology. Source: (Huber et al., 2019)

3.3 Approach

3.3.1 Business Undertanding

The understanding of the project involved consulting with cybersecurity experts specifically the cybersecurity analysts about cybersecurity report summarization process. An automated data exploration strategy was used in this research. The exploration was done on both local development environment(Anaconda) and cloud based development environment (Google Colab) where data sets in different file formats such as csv, xlsx and txt were loaded then visualized in order to gauge whether the data properties can help in achieving the objectives. The following tools were used for textual data collection: Content grabber and Common crawl among others.

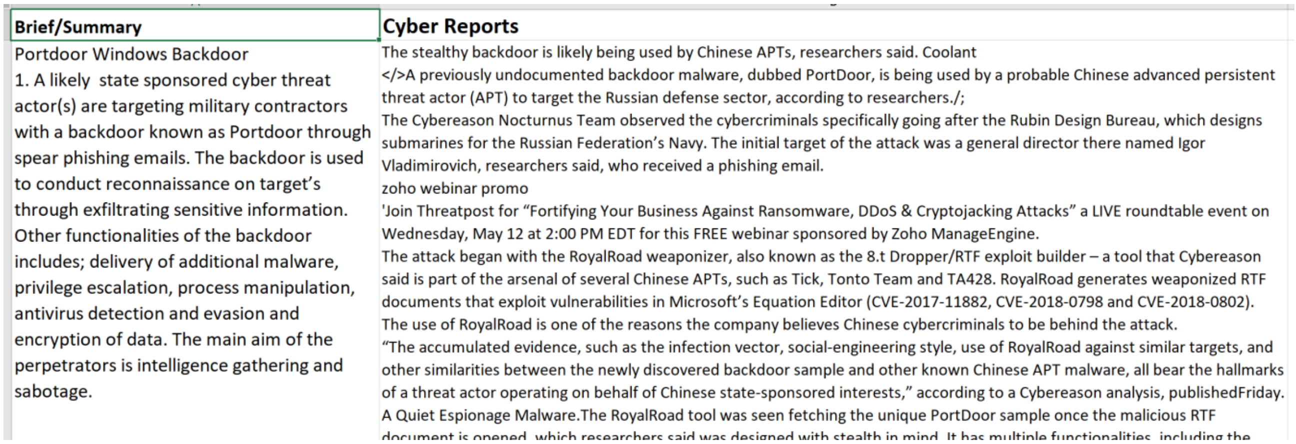
3.3.2 Data Understanding

Dataset Description

The understanding of the project requirements involved consultation with Cybersecurity Experts, specifically Cybersecurity Analysts, about creating cybersecurity strategic briefs. Contact with Head of Cybersecurity Analysis Team at Ekraal Innovation Hub showed that there is need for summarization of the daily voluminous cybersecurity reports produced within the Global Cyberspace. This will assist in supporting strategic decision making process that's most often time bound. In text summarization, cybersecurity datasets, particularly textual datasets in form of reports from advanced persistent attacks, malware reports repository, cyber incidents, web blogs, cybersecurity news bulletins and expert summaries were explored during model building, testing and evaluation.

The data structure was in form of lengthy reports that's not summarized and a corresponding summary that's cybersecurity briefs/summaries created by human analyst. The 'Cyber Reports 'column contains lengthy unsummarized report/text with its corresponding brief/summary on the same row. Out of the whole dataset, 80% was primarily used to train the model, whereas 20% dataset was used to test the accuracy of the model through comparison techniques. The data structure is as shown in table 1 below.

Table 1:Dataset format used for training and testing the model



Brief/Summary	Cyber Reports
Portdoor Windows Backdoor 1. A likely state sponsored cyber threat actor(s) are targeting military contractors with a backdoor known as Portdoor through spear phishing emails. The backdoor is used to conduct reconnaissance on target's through exfiltrating sensitive information. Other functionalities of the backdoor includes; delivery of additional malware, privilege escalation, process manipulation, antivirus detection and evasion and encryption of data. The main aim of the perpetrators is intelligence gathering and sabotage.	The stealthy backdoor is likely being used by Chinese APTs, researchers said. Coolant </>A previously undocumented backdoor malware, dubbed PortDoor, is being used by a probable Chinese advanced persistent threat actor (APT) to target the Russian defense sector, according to researchers./; The Cybereason Nocturnus Team observed the cybercriminals specifically going after the Rubin Design Bureau, which designs submarines for the Russian Federation's Navy. The initial target of the attack was a general director there named Igor Vladimirovich, researchers said, who received a phishing email. zoho webinar promo 'Join Threatpost for "Fortifying Your Business Against Ransomware, DDoS & Cryptojacking Attacks" a LIVE roundtable event on Wednesday, May 12 at 2:00 PM EDT for this FREE webinar sponsored by Zoho ManageEngine. The attack began with the RoyalRoad weaponizer, also known as the 8.t Dropper/RTF exploit builder – a tool that Cybereason said is part of the arsenal of several Chinese APTs, such as Tick, Tonto Team and TA428. RoyalRoad generates weaponized RTF documents that exploit vulnerabilities in Microsoft's Equation Editor (CVE-2017-11882, CVE-2018-0798 and CVE-2018-0802). The use of RoyalRoad is one of the reasons the company believes Chinese cybercriminals to be behind the attack. "The accumulated evidence, such as the infection vector, social-engineering style, use of RoyalRoad against similar targets, and other similarities between the newly discovered backdoor sample and other known Chinese APT malware, all bear the hallmarks of a threat actor operating on behalf of Chinese state-sponsored interests," according to a Cybereason analysis, publishedFriday. A Quiet Espionage Malware.The RoyalRoad tool was seen fetching the unique PortDoor sample once the malicious RTF document is opened, which researchers said was designed with stealth in mind. It has multiple functionalities, including the

Understanding Sequence distribution

At this step, the lengthy of Cyber reports and that of the briefs was analyzed in order to understand and get an overall idea of text length. This in turn helped in fixing sequence length.The figure 3 below illustrates the text distribution for both summaries generated and text.

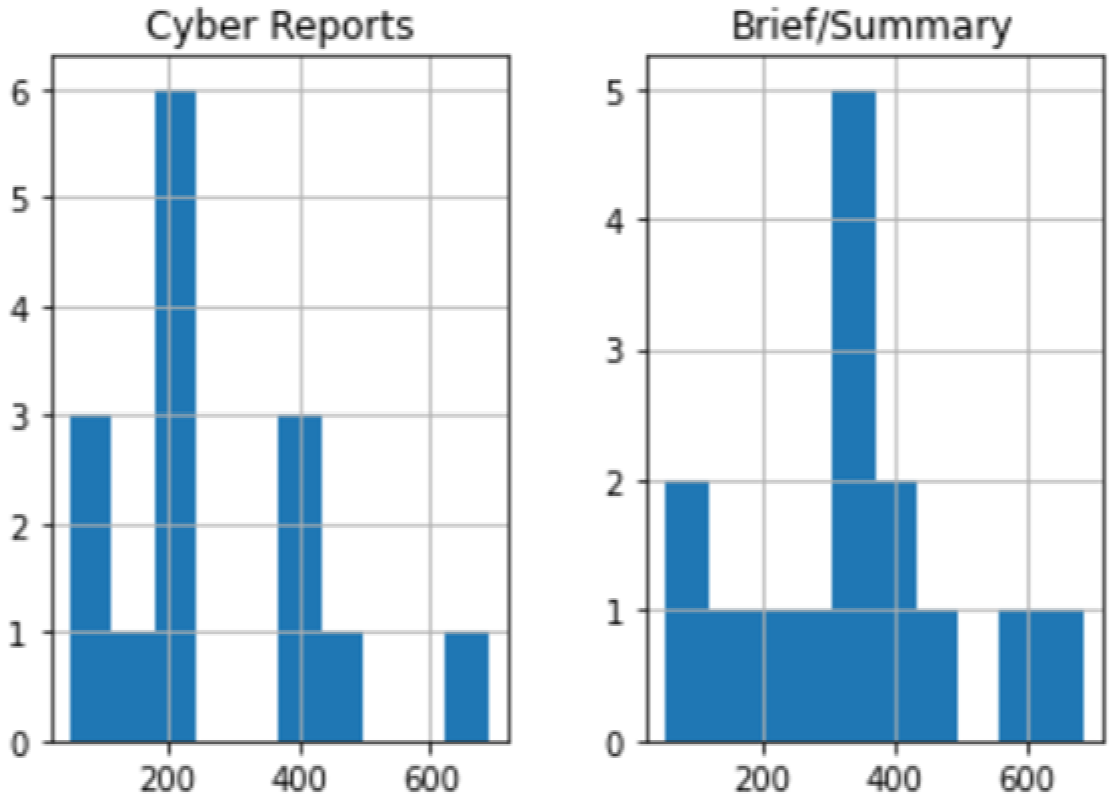


Figure 3: Plot of Data length distribution

Data Sources

The public and private dataset was used in model building. Getting access to the private dataset required seeking authorization first before utilization due to the classification and sensitivity nature of security reports. The most ideal dataset that was explored in this research was downloaded from Cyware Cybersecurity blog and the Human Expert Summaries used for testing that were acquired from Ekraal Innovation Hub together with other data from various open sources which include:

Microsoft Malware Classification Challenge- The data sets contained a set of known malware files which is a mixture of 9 family names of the malware samples.

UNSW-UB15 dataset- It contained nine families of attacks which include: DDoS & DoS, Exploits, Fuzzers, Backdoors, Reconnaissance and Worms.

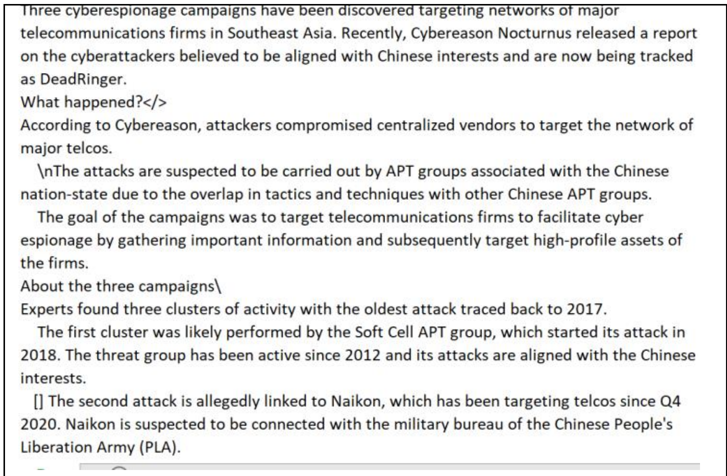
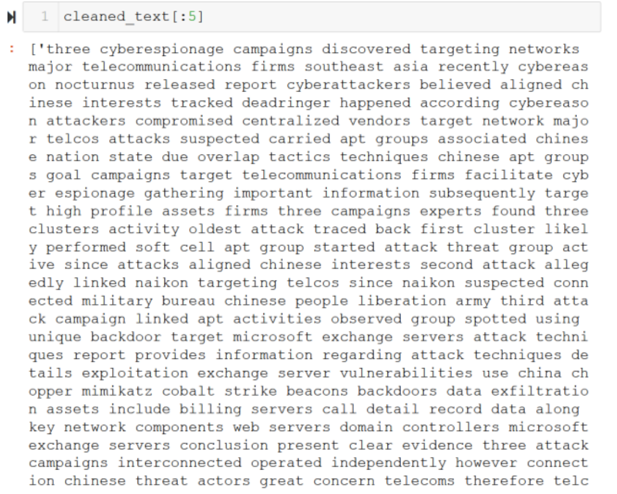
Threat Research – This is a centralized repository for threat research data gathered from various network honeypots.

3.3.3 Data Preparation

Out of the total volume of data collected, 80% was used for model training whereas 20% was used for model testing. Before the dataset was used for model building, it underwent the following data preprocessing activities which include:

- Stop words removal. This step entailed removing words that are of less importance during processing. The major attributes used to determine stop words included but not limited to terms that have high frequency of occurrence such as conjunctions (or, and, but etc.). However, there are no definite rules of determing the stop words since the determination of those words can be adjusted depending on the case being handled and the language in use.
- Stemming. This was the second stage of data preprocessing that was used to remove affixes and change words into their basic forms.
- Tokenizing. At this stage, paragraphs, sentences or documents were split into parts/tokens. For example, “Automated text summarization is not a trivial task” was tokenized to “Automated”, “text”, “summarization”, “is”, “not”, “a “, “trivial”, “task”.
- Convert everything to lowercase
- Removal of HTML tags using Regex method. In order to ensure that the text corpus is cleaned and stripped off the HTML tags, a Regex method was employed in removing these tags as indicated in the code snippets attached at the Appendix1. Remove HTML tags.
- Contraction mapping which involved mapping words/combination of words that are shortened and replaced by the apostrophe. This procedure helped in dimensionality reduction during word vectorization. This was achieved through use of pycontractions library that was installed on Google colab as shown in the Appendices.
- Lemmatization. This stage involved normalizing words or making words that contain affection into basic forms.
- Term Weighting involved judging the words to determine their importance in summary generation process.
- Other preprocessing techniques that were used include word segmentation, word frequency determination, proper noun set and use of Bag of Words among others.

Table 2: Unclean and cleaned data

 Three cyberespionage campaigns have been discovered targeting networks of major telecommunications firms in Southeast Asia. Recently, Cybereason Nocturnus released a report on the cyberattackers believed to be aligned with Chinese interests and are now being tracked as DeadRinger. What happened?</> According to Cybereason, attackers compromised centralized vendors to target the network of major telcos. \nThe attacks are suspected to be carried out by APT groups associated with the Chinese nation-state due to the overlap in tactics and techniques with other Chinese APT groups. The goal of the campaigns was to target telecommunications firms to facilitate cyber espionage by gathering important information and subsequently target high-profile assets of the firms. About the three campaigns\\ Experts found three clusters of activity with the oldest attack traced back to 2017. The first cluster was likely performed by the Soft Cell APT group, which started its attack in 2018. The threat group has been active since 2012 and its attacks are aligned with the Chinese interests. [] The second attack is allegedly linked to Naikon, which has been targeting telcos since Q4 2020. Naikon is suspected to be connected with the military bureau of the Chinese People's Liberation Army (PLA).	 cleaned_text[:5] : ['three cyberespionage campaigns discovered targeting networks major telecommunications firms southeast asia recently cybereas on nocturnus released report cyberattackers believed aligned ch inese interests tracked deadringer happened according cybereaso n attackers compromised centralized vendors target network majo r telcos attacks suspected carried apt groups associated chines e nation state due overlap tactics techniques chinese apt group s goal campaigns target telecommunications firms facilitate cyb er espionage gathering important information subsequently targe t high profile assets firms three campaigns experts found three clusters activity oldest attack traced back first cluster likel y performed soft cell apt group started attack threat group act ive since attacks aligned chinese interests second attack alleg edly linked naikon targeting telcos since naikon suspected conn ected military bureau chinese people liberation army third atta ck campaign linked apt activities observed group spotted using unique backdoor target microsoft exchange servers attack techni ques report provides information regarding attack techniques de tails exploitation exchange server vulnerabilities use china ch opper mimikatz cobalt strike beacons backdoors data exfiltratio n assets include billing servers call detail record data along key network components web servers domain controllers microsoft exchange servers conclusion present clear evidence three attack campaigns interconnected operated independently however connect ion chinese threat actors great concern telecoms therefore telc
Unclean/unpreprocessed data	Cleaned data

Splitting Data. The dataset was divided into two sets, one set for training the model and the other for evaluating the model’s accuracy/performance with the help of “sklearn”- “train_test_split” function. In order to ensure that all the sequences are of the same length, Keras sequence padding was utilized which added 0 at the start of each sequence to achieve uniformity in all sequence lengths.

3.3.4 Modeling

Modeling was done using Recurrent Neural Network variant(LSTM). The LSTM neural network is further discussed in detail below.

3.3.4.1 Detailed Seq2Seq modeling

The main goal was to build a cybersecurity text summarization model which should accept a long sequence of words as input , that can be modelled as a many to many (seq2seq) problem and be able to generate a brief from large sequence of text.

LSTM which is a variant of Recurrent Neural Network(RNN) was preferably used as the encoder and decoder for word sequence generation. This is due to its ability to capture long term dependencies through overcoming the issue of vanishing gradient. The encoder-decoder was set up in two phases: the training phase and inference phase.

A typical architecture of the seq2seq model is as shown in figure 4 below:

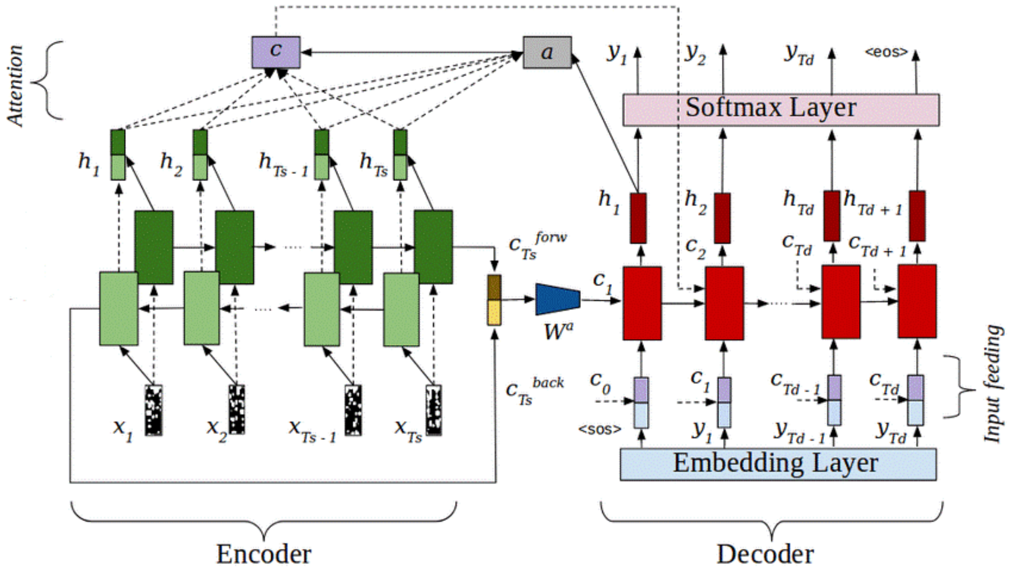


Figure 4: LSTM for both encoding and decoding. Source (Samurainote, 2019)

Figure 5 below illustrates the operation of cell state.

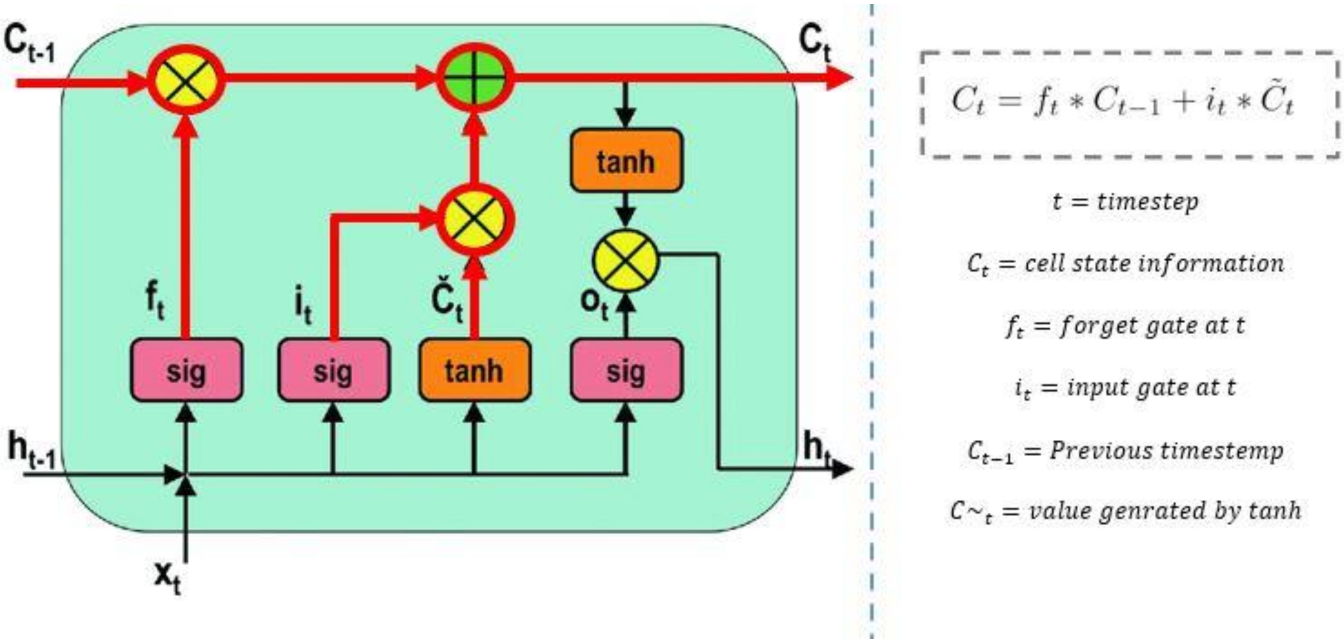

C_{t-1}
C_t
tanh
f_t
i_t
$\check{C}_t$
o_t
sig
sig
tanh
sig
h_{t-1}
h_t
x_t
$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$
$t = timestep$
$C_t = cell\ state\ information$
$f_t = forget\ gate\ at\ t$
$i_t = input\ gate\ at\ t$
$C_{t-1} = Previous\ timestemp$
$C\sim_t = value\ genrated\ by\ tanh$

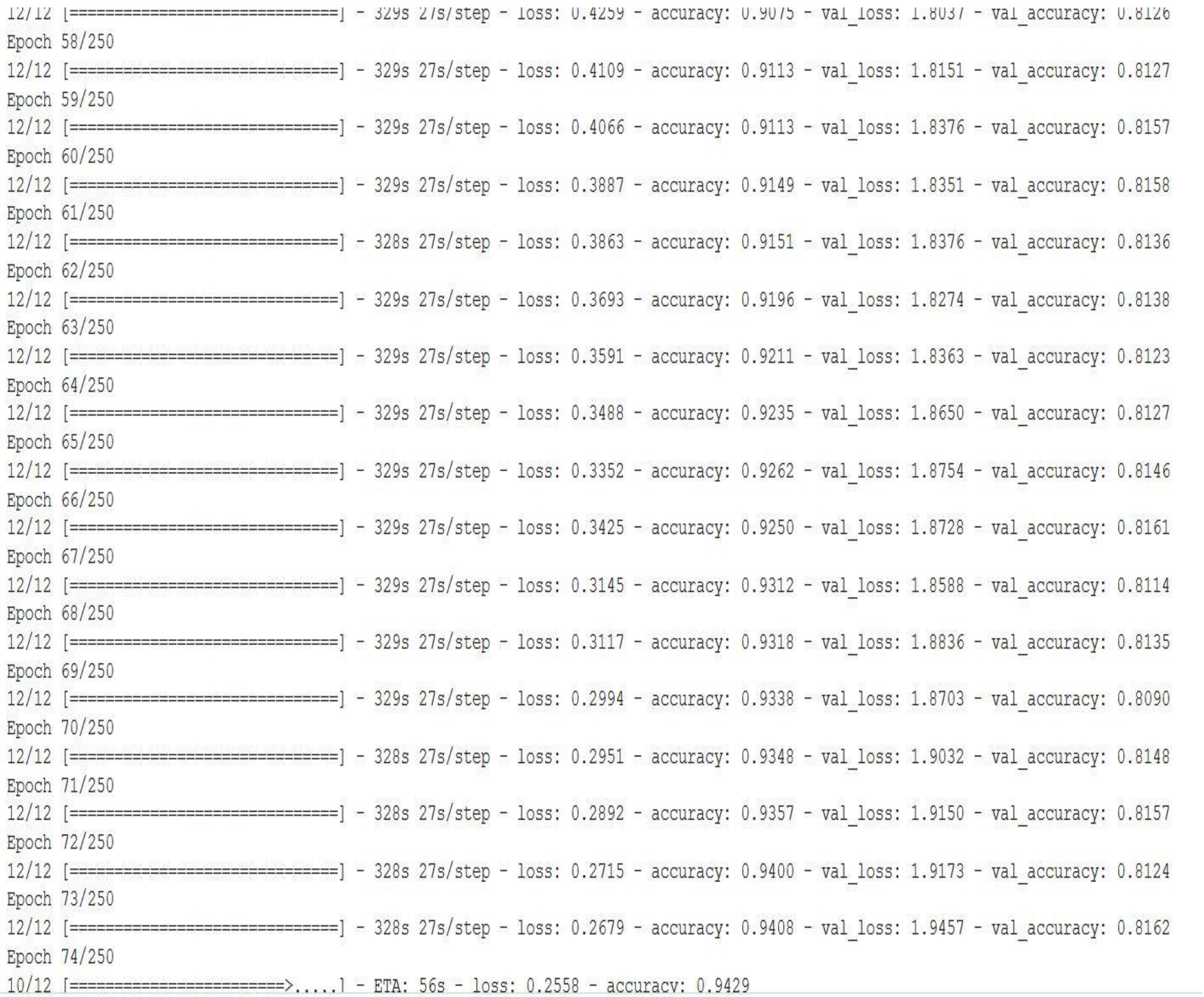
Figure 5: LSTM Cell State Operation. Source (Pluralsight, 2020)

The C(t-1) cell state is multiplied by the f(t) vector.The result will be dropped if the value of the output is zero, else the input vectori(t) is taken to update the state of the cell resulting in a new cell state C(t).Lastly, the new cell state is transferred to the next time step.

Model Training phase

At this phase, the setting up of the encoder-decoder was done and then proceeded with training the model in order to be able to predict the target sequence offset in each timestep.

The lowest loss recorded was 26% , while the highest accuracy recored was 94% at epoch 74.This meant that for better performance given the same data, the model can only be trained for 74 epochs in order to get optimal results.Figure 6 further illustrates the loss versus accuracy progress during training. When loss is plotted against epoch, there is a progressive decrease in loss with an increase in the number of epochs for both training and test data. Loss, in this case, occur as a result of a bad prediction.



```
12/12 [==============================] - 329s 27s/step - loss: 0.4259 - accuracy: 0.9075 - val_loss: 1.8037 - val_accuracy: 0.8126
Epoch 58/250
12/12 [==============================] - 329s 27s/step - loss: 0.4109 - accuracy: 0.9113 - val_loss: 1.8151 - val_accuracy: 0.8127
Epoch 59/250
12/12 [==============================] - 329s 27s/step - loss: 0.4066 - accuracy: 0.9113 - val_loss: 1.8376 - val_accuracy: 0.8157
Epoch 60/250
12/12 [==============================] - 329s 27s/step - loss: 0.3887 - accuracy: 0.9149 - val_loss: 1.8351 - val_accuracy: 0.8158
Epoch 61/250
12/12 [==============================] - 328s 27s/step - loss: 0.3863 - accuracy: 0.9151 - val_loss: 1.8376 - val_accuracy: 0.8136
Epoch 62/250
12/12 [==============================] - 329s 27s/step - loss: 0.3693 - accuracy: 0.9196 - val_loss: 1.8274 - val_accuracy: 0.8138
Epoch 63/250
12/12 [==============================] - 329s 27s/step - loss: 0.3591 - accuracy: 0.9211 - val_loss: 1.8363 - val_accuracy: 0.8123
Epoch 64/250
12/12 [==============================] - 329s 27s/step - loss: 0.3488 - accuracy: 0.9235 - val_loss: 1.8650 - val_accuracy: 0.8127
Epoch 65/250
12/12 [==============================] - 329s 27s/step - loss: 0.3352 - accuracy: 0.9262 - val_loss: 1.8754 - val_accuracy: 0.8146
Epoch 66/250
12/12 [==============================] - 329s 27s/step - loss: 0.3425 - accuracy: 0.9250 - val_loss: 1.8728 - val_accuracy: 0.8161
Epoch 67/250
12/12 [==============================] - 329s 27s/step - loss: 0.3145 - accuracy: 0.9312 - val_loss: 1.8588 - val_accuracy: 0.8114
Epoch 68/250
12/12 [==============================] - 329s 27s/step - loss: 0.3117 - accuracy: 0.9318 - val_loss: 1.8836 - val_accuracy: 0.8135
Epoch 69/250
12/12 [==============================] - 329s 27s/step - loss: 0.2994 - accuracy: 0.9338 - val_loss: 1.8703 - val_accuracy: 0.8090
Epoch 70/250
12/12 [==============================] - 328s 27s/step - loss: 0.2951 - accuracy: 0.9348 - val_loss: 1.9032 - val_accuracy: 0.8148
Epoch 71/250
12/12 [==============================] - 328s 27s/step - loss: 0.2892 - accuracy: 0.9357 - val_loss: 1.9150 - val_accuracy: 0.8157
Epoch 72/250
12/12 [==============================] - 328s 27s/step - loss: 0.2715 - accuracy: 0.9400 - val_loss: 1.9173 - val_accuracy: 0.8124
Epoch 73/250
12/12 [==============================] - 328s 27s/step - loss: 0.2679 - accuracy: 0.9408 - val_loss: 1.9457 - val_accuracy: 0.8162
Epoch 74/250
10/12 [========================>.....] - ETA: 56s - loss: 0.2558 - accuracy: 0.9429
```

Figure 6: Modeling Point of Optimal results

Encoder

The LSTM model read the whole input sequence wherein, at every time-step, single words were fed to the encoder. The information at each time-step was processed and the current/present contextual information in the input sequence at every time-step was captured. Figure 7 below illustrates this process:

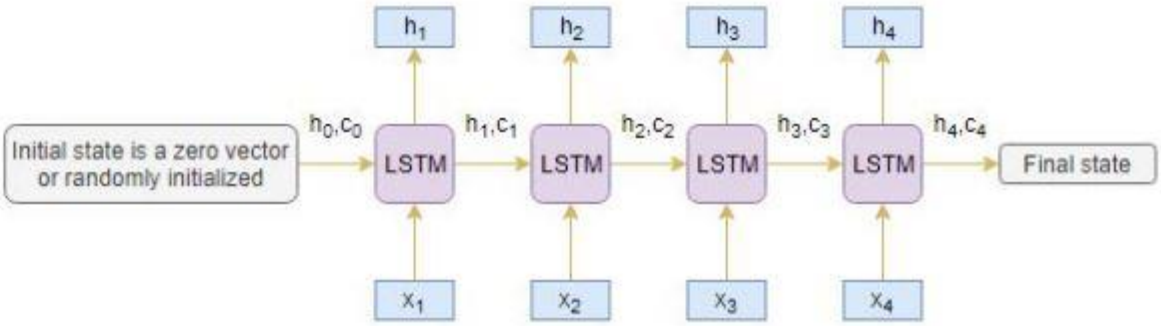


Figure 7: The Long Short Term Memory (LSTM), for encoding during training phase. Source (Samurainote, 2019)

Decoder

This is an LSTM network that reads the whole targeted sequence word by word and then predicts the similar offset sequence by one timestemp. In other words, the decoder is trained to predict the next word in the given sequence using the previous words.

Inference Phase

After the model was trained successfully, it was then subjected on a new source of sequence during testing where the target sequence is unknown. The architecture below on figure 8 illustrate the process:

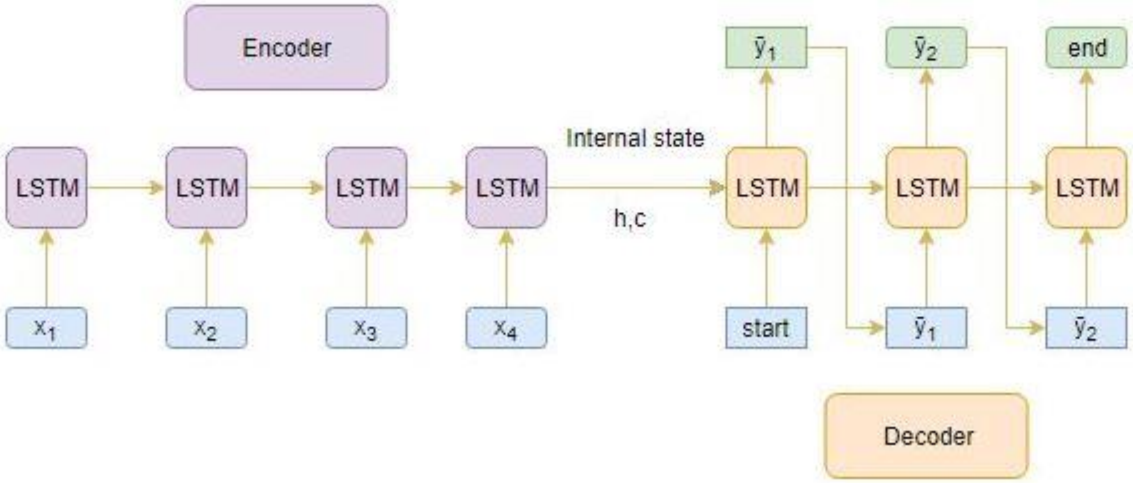


Figure 8: The Long Short Term Memory (LSTM), for decoding during inference phase. Source (Samurainote, 2019)

3.3.4.2 Discussion on the techniques/methods that were used during automated cybersecurity briefing based on text summarization

From the last ten years, most text summarization work has been using the common 6 approaches and techniques which include rule based, fuzzy based, graphics, statistics and machine learning. This research combined statistics and machine learning approach as discussed below:

Machine learning Method

The approach that was utilized widely in this automatic text summarization was machine learning which is the commonly used technique in automated text summary generation (Widyassari et al., 2020). It was the favorite technique used because, it's the latest approach whose performance can be automated and improvements made with time due to the ability to learn continuously. Summary is generated from various documents using the term frequency count, title extractions, the sentence position and cue phrases without paying much attention to the meaning of document sentences. This approach produced a coherent summary that is almost close to that produced by human.

Statistical Method

This method was used in combination with machine learning in areas such as frequency count of the sentence positions, phrases and terms/words. Sentences with highest frequency scores were extracted to form the summary [2]. The concept of position feature played a critical role in statistical text summarization process as said by Kupiec, Pedersen, and Chen in 1995 [4] in creating a statistical summarization that uses the Bayesian classification algorithm for summarization.

3.3.5 Evaluation

3.3.5.1 Evaluation of the model Using ROUGE

ROUGE being one of the standardly used measures to evaluate the summaries generated automatically, it was chosen to be the de facto summary evaluator. Since the ideal summary differ from one person to another, the summary model evaluation is not an easy task. However, the variants of ROUGE metric usually used for evaluating unsupervised summarization models were utilized. These metrics rely on calculating similarity between the summary under evaluation with a list of reference summaries from the human experts.

However, before settling on which ROUGE variant to use on which method, a prior further assessment of the automatic evaluation metrics for content selection standardly used in summarization research was done.

The metrics compares one or more multiple reference summaries created by expert with the machine generated summaries. The set of metrics that were used for assessing the automatic text summarization process include:

- ROUGE-1- Compares the overlap of unigrams
- ROUGE- 2- Compares the overlap of bigrams i.e. two adjacent words
- ROUGE-L- Longest Common Subsequence(LCS) that checks on in-sequence matches that show the level of sentence structure similarity

For each metric, F1 score, recall and precision were calculated.

The performance metric of summarization model was based on accuracy. The function created accepts inputs, computes the metrics and gives results by comparing machine generated summary with reference sample summaries created by human experts.

After the model was trained with adequate data, an experimental analysis was done with key focus on the quality of the summary generated. The qualitative analysis assessed the linguistic quality, readability and then compared the machine/model generated summary with sample summaries from cybersecurity analysts and the existing models like abstractive BERT and extractive lexRank.

Steinberger and Jezek says that during text summarization process, human annotators are used to evaluate the text quality (2009). The annotators are used to set the scale value determined by each summary. Summary evaluation in terms of extracting sentences, content generated and task based was also considered as they have been used in the previous studies.

The precision, recall and f-score measurement methods were used to gauge the level of accuracy. For content evaluation, each actual word in every sentence and not the whole sentence was compared. The equations used in computing the above measures were:

$$Precision=TP/TP+FP$$

$$Recall=TP/TP+FN$$

$$F{-}score=2Precision/Recall\ Precision + Recall$$

Where; TP-True Positive, FN- False Negative, TN-True Negative, FP-False Positive

3.3.6 Deployment

3.3.6.1 Resources Required

The experimentation was done on Google Colaboratory cloud platform when training the model. However, use of local development environment for testing and evaluation was also utilized when sizeable amount of dataset was used. The local requirements set up include Recurrent Neural Network for automatic feature extraction, personal computer with Intel(R) HD Graphics 5500, Intel(R) Core(TM) i7-5600U CPU @2.60GHz(4CPUs), ~2.6GHz running on Ubuntu 19 operating system.

The Google Colab platform was largely utilized because it offers free GPU that meets the high computational power required when processing voluminous datasets. Some of the benefits of using this platform is zero configuration requirements, easy sharing and free access to computing resources like the GPUs.Python was used as the development language together with other data processing libraries as shown in Table 3:

Table 3:Resources Utilized

Tool	Description
Anaconda IDE	An integrated environment for distributed python programming language for machine learning apps, data science and large scale data processing.
Jupyter Notebook	Interactive data science environment across many programming languages that doesn't only work as an IDE, but also as a presentation or education tool
Dataset	Cybersecurity data
Python	High level programming language
Keras	Free and open-source software library for machine learning
TensorFlow	It's a free and open-source software library for machine learning
Numpy	Collection of mathematical functions that support the operations of multi-dimensional arrays and matrices
NLTK corpus	A package of large corpus reader classes that is used for diverse collection of corpora access
Attention Layer	This is vectorization of words, simply the results of the dense layer when using the softmax function. This enables the text summarization through deep learning to hold context of the original text that is used later for summary generation.
Scikit-learn	A python machine learning library
Matplotlib	Machine learning library for interactive visualization
Beautiful Soup	Web scrapping package used for parsing html and xml files

4 RESULTS AND DISCUSSIONS

4.1. Introduction

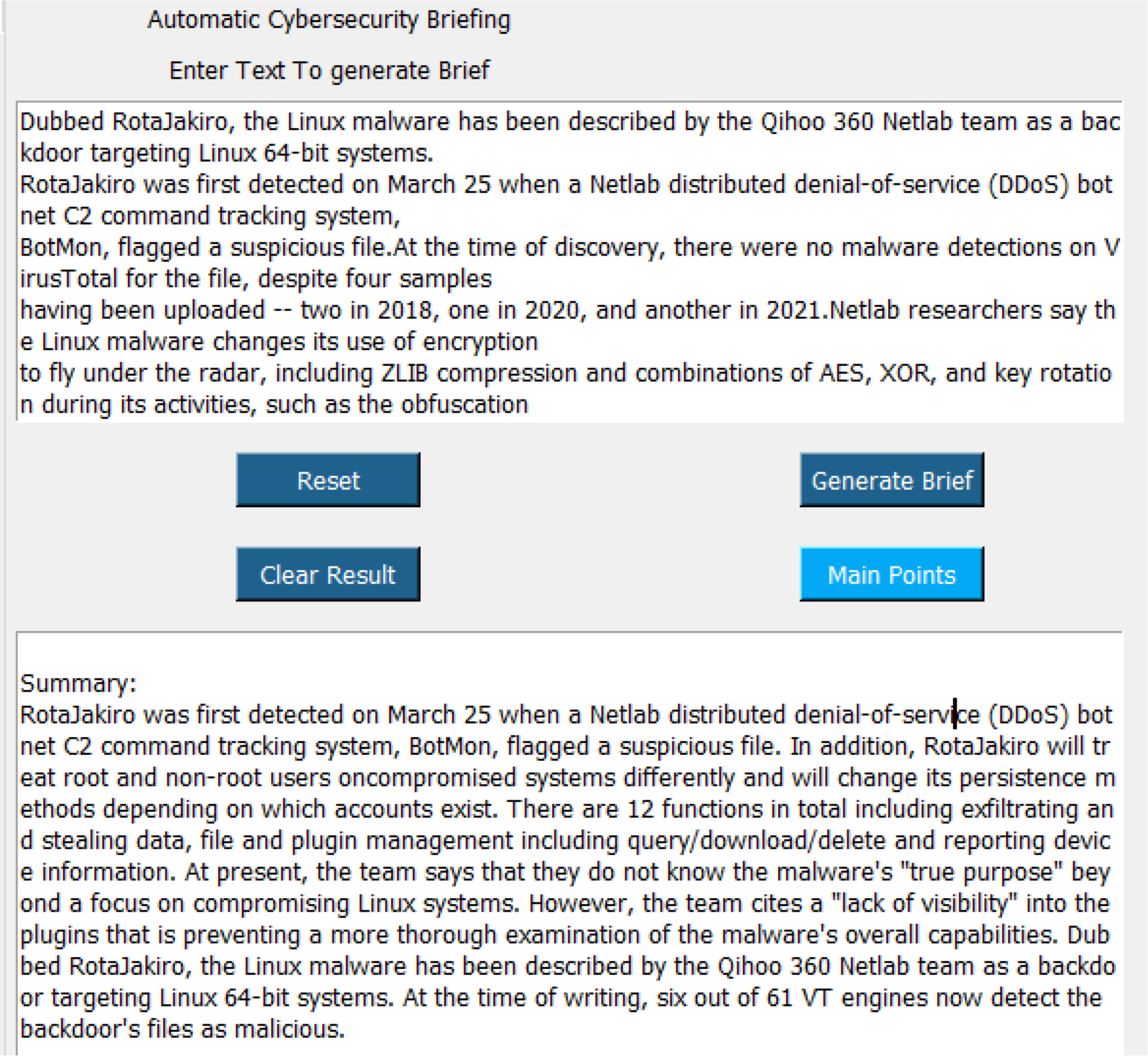
This chapter covers the results and discussions based on the model's performance and evaluation. The brief generation was done using the Recurrent Neural Network, the LSTM variant through feature extraction and deep learning.

4.2. Results

In order achieve consistency, same original text used in evaluating the performance of both the trained and existing related model. However, it should be noted that the trained model was subjected to validation test by industry experts where they loaded reports to the model and compared the generated summary with their summaries from the same reports.

4.2.1 Model Output

Figure 9 below shows the brief generated when a lengthy text is pasted and its equivalent summary.

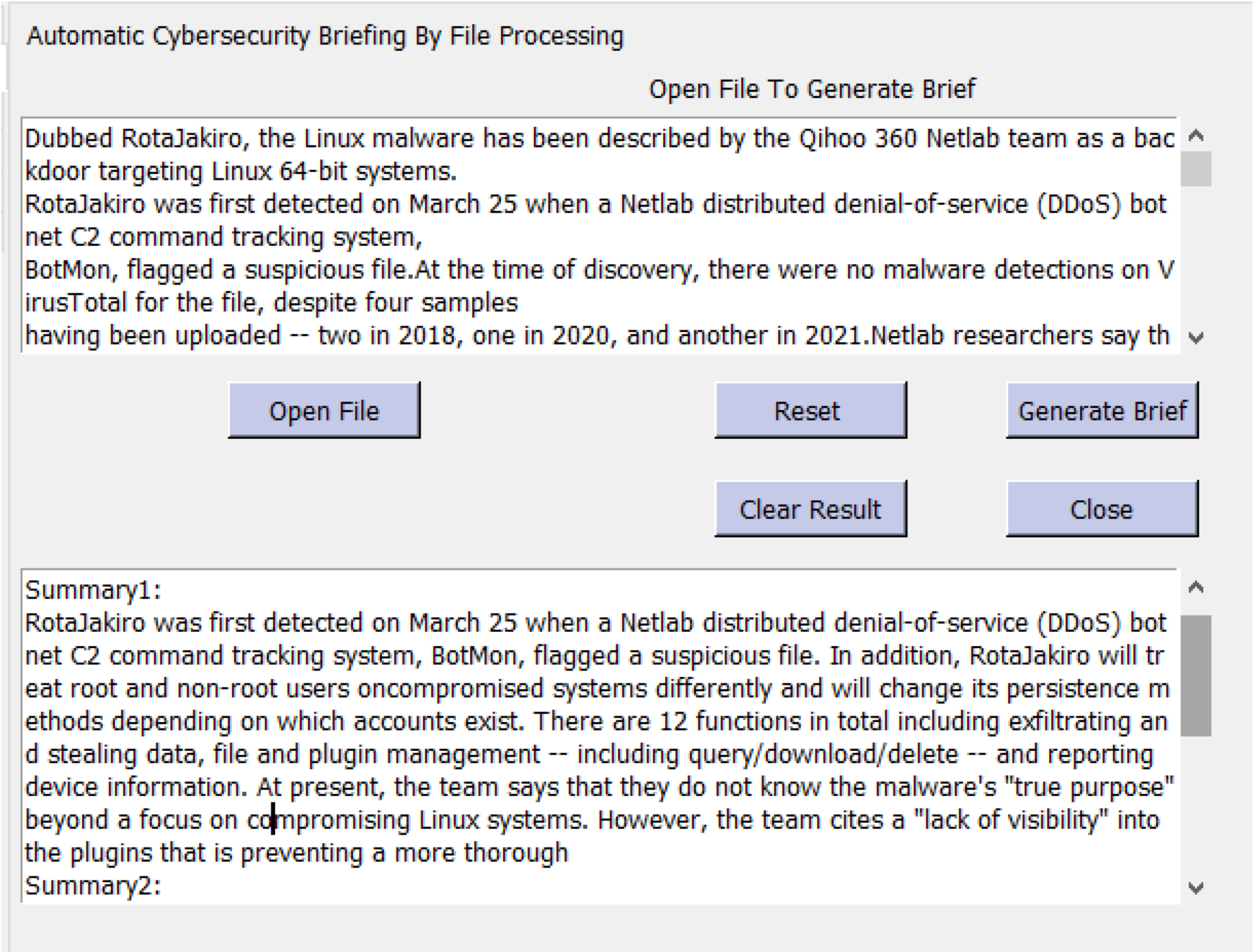

Automatic Cybersecurity Briefing
Enter Text To generate Brief
Reset
Generate Brief
Clear Result
Main Points

Dubbed RotaJakiro, the Linux malware has been described by the Qihoo 360 Netlab team as a backdoor targeting Linux 64-bit systems.
RotaJakiro was first detected on March 25 when a Netlab distributed denial-of-service (DDoS) botnet C2 command tracking system,
BotMon, flagged a suspicious file.At the time of discovery, there were no malware detections on VirusTotal for the file, despite four samples
having been uploaded -- two in 2018, one in 2020, and another in 2021.Netlab researchers say the Linux malware changes its use of encryption
to fly under the radar, including ZLIB compression and combinations of AES, XOR, and key rotation during its activities, such as the obfuscation

Summary:
RotaJakiro was first detected on March 25 when a Netlab distributed denial-of-service (DDoS) botnet C2 command tracking system, BotMon, flagged a suspicious file. In addition, RotaJakiro will treat root and non-root users oncompromised systems differently and will change its persistence methods depending on which accounts exist. There are 12 functions in total including exfiltrating and stealing data, file and plugin management including query/download/delete and reporting device information. At present, the team says that they do not know the malware's "true purpose" beyond a focus on compromising Linux systems. However, the team cites a "lack of visibility" into the plugins that is preventing a more thorough examination of the malware's overall capabilities. Dubbed RotaJakiro, the Linux malware has been described by the Qihoo 360 Netlab team as a backdoor targeting Linux 64-bit systems. At the time of writing, six out of 61 VT engines now detect the backdoor's files as malicious.

Figure 9: Brief generation from a lengthy text

The following figure 10 illustrates the summary generated from a files. The files were loaded from directory and their equivalent summaries were formed. In this case, a user can load documents into the model in order to generate briefs.

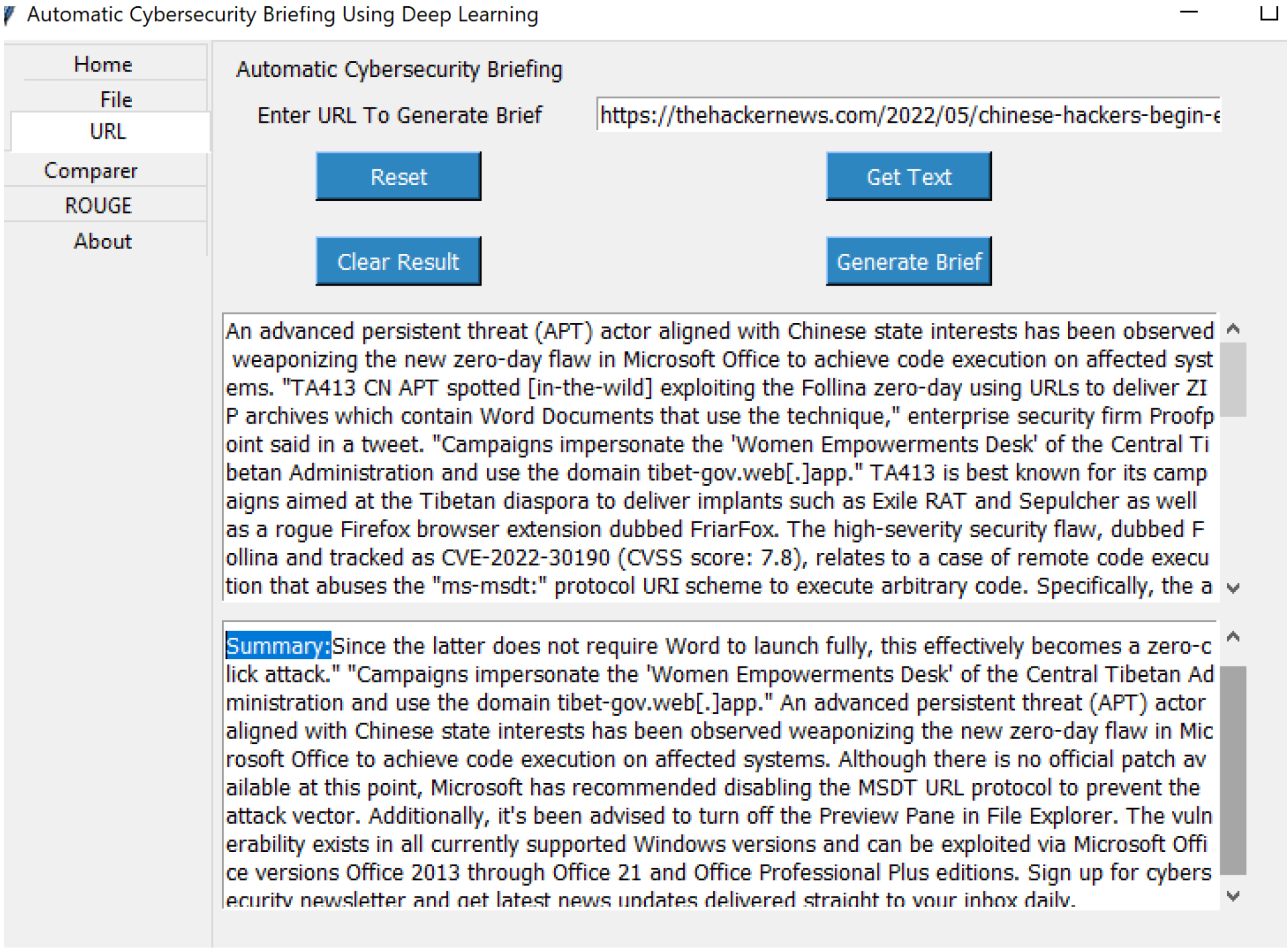

Automatic Cybersecurity Briefing By File Processing
Open File To Generate Brief
Open File
Reset
Generate Brief
Clear Result
Close

```
Dubbed RotaJakiro, the Linux malware has been described by the Qihoo 360 Netlab team as a bac
kdoor targeting Linux 64-bit systems.
RotaJakiro was first detected on March 25 when a Netlab distributed denial-of-service (DDoS) bot
net C2 command tracking system,
BotMon, flagged a suspicious file.At the time of discovery, there were no malware detections on V
irusTotal for the file, despite four samples
having been uploaded -- two in 2018, one in 2020, and another in 2021.Netlab researchers say th
```

```
Summary1:
RotaJakiro was first detected on March 25 when a Netlab distributed denial-of-service (DDoS) bot
net C2 command tracking system, BotMon, flagged a suspicious file. In addition, RotaJakiro will tr
eat root and non-root users oncompromised systems differently and will change its persistence m
ethods depending on which accounts exist. There are 12 functions in total including exfiltrating an
d stealing data, file and plugin management -- including query/download/delete -- and reporting
device information. At present, the team says that they do not know the malware's "true purpose"
beyond a focus on compromising Linux systems. However, the team cites a "lack of visibility" into
the plugins that is preventing a more thorough
Summary2:
```

Figure 10:Brief generation from files/reports

Additionally, a user can insert a URL from cybersecurity blogs to generate a brief. Before a brief is generated, the text from the webpage undergoes pre-processing then the clean text is summarized. Figure 11 below shows text from URL, cleaned text and its equivalent summary:


Automatic Cybersecurity Briefing Using Deep Learning
Home
File
URL
Comparer
ROUGE
About
Automatic Cybersecurity Briefing
Reset
Get Text
Clear Result
Generate Brief

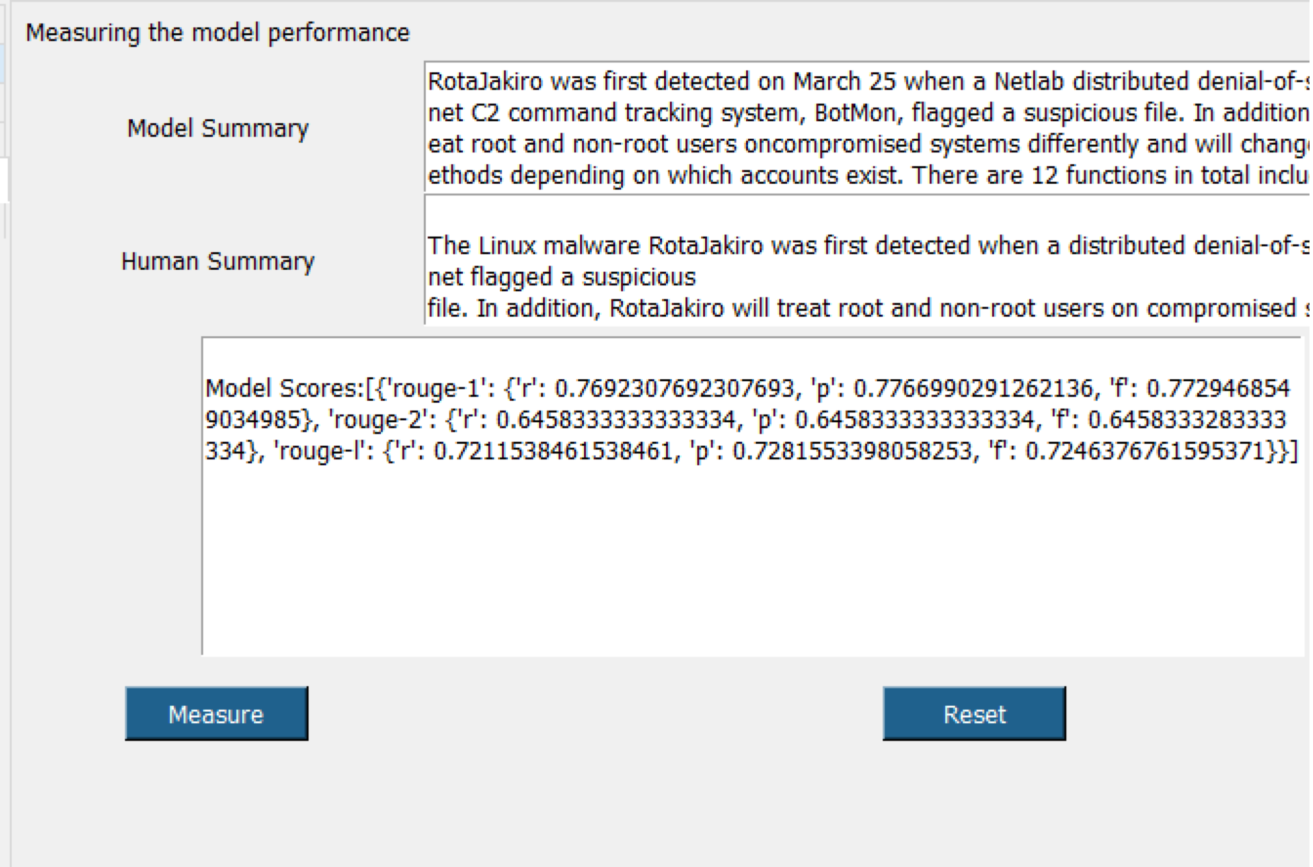
Enter URL To Generate Brief https://thehackernews.com/2022/05/chinese-hackers-begin-e

An advanced persistent threat (APT) actor aligned with Chinese state interests has been observed weaponizing the new zero-day flaw in Microsoft Office to achieve code execution on affected systems. "TA413 CN APT spotted [in-the-wild] exploiting the Follina zero-day using URLs to deliver ZIP archives which contain Word Documents that use the technique," enterprise security firm Proofpoint said in a tweet. "Campaigns impersonate the 'Women Empowerments Desk' of the Central Tibetan Administration and use the domain tibet-gov.web[.]app." TA413 is best known for its campaigns aimed at the Tibetan diaspora to deliver implants such as Exile RAT and Sepulcher as well as a rogue Firefox browser extension dubbed FriarFox. The high-severity security flaw, dubbed Follina and tracked as CVE-2022-30190 (CVSS score: 7.8), relates to a case of remote code execution that abuses the "ms-msdt:" protocol URI scheme to execute arbitrary code. Specifically, the a

Summary:Since the latter does not require Word to launch fully, this effectively becomes a zero-click attack." "Campaigns impersonate the 'Women Empowerments Desk' of the Central Tibetan Administration and use the domain tibet-gov.web[.]app." An advanced persistent threat (APT) actor aligned with Chinese state interests has been observed weaponizing the new zero-day flaw in Microsoft Office to achieve code execution on affected systems. Although there is no official patch available at this point, Microsoft has recommended disabling the MSDT URL protocol to prevent the attack vector. Additionally, it's been advised to turn off the Preview Pane in File Explorer. The vulnerability exists in all currently supported Windows versions and can be exploited via Microsoft Office versions Office 2013 through Office 21 and Office Professional Plus editions. Sign up for cybersecurity newsletter and get latest news updates delivered straight to your inbox daily.

Figure 11:Brief generation from a web blog

Figure 12 gives a graphical overview of the evaluation process. This where the model summary is evaluated against the human summary (expert summary used as the ideal summary) and metrics/model scores based on the ROUGE variants (ROUGE-1, ROUGE-2 and ROUGE-LCS) are generated.



RotaJakiro was first detected on March 25 when a Netlab distributed denial-of-s
net C2 command tracking system, BotMon, flagged a suspicious file. In addition
eat root and non-root users oncompromised systems differently and will chang
ethods depending on which accounts exist. There are 12 functions in total inclu

The Linux malware RotaJakiro was first detected when a distributed denial-of-s
net flagged a suspicious
file. In addition, RotaJakiro will treat root and non-root users on compromised s

```
Model Scores:[{'rouge-1': {'r': 0.7692307692307693, 'p': 0.7766990291262136, 'f': 0.772946854
9034985}, 'rouge-2': {'r': 0.6458333333333334, 'p': 0.6458333333333334, 'f': 0.6458333283333
334}, 'rouge-l': {'r': 0.7211538461538461, 'p': 0.7281553398058253, 'f': 0.7246376761595371}}]
```

Figure 12:Measuring Model performance

4.2.2 Discussion on the results and accuracy of the model

The Recurrent Neural Network(LSTM) was trained for 250 epochs. During the evaluation process, the same original document (text **x**) was ingested to the three summarizers (method 1,2&3) in order to obtain different summaries (summary **y**) from single original text for consistency. The performance of the LSTM Abstractive based Model when comparing the overlap of unigrams (ROUGE 1) between the machine summary and ideal summary is satisfactory. However, the accuracy of the other two variants, ROUGE 2 is slightly lower since overlap of two or more words is minimal. The accuracy that was recorded is as shown in Table 4 below.

Table 4: Accuracy of the abstractive based trained Model based on the generated Summary Y(Automated Cybersecurity Briefing Model, method 3)

LSTM Abstractive Based Model(Trained)		
Rouge Variant	**Metric**	**Accuracy**
ROUGE-1- Compares the overlap of unigrams	Recall	87%
	Precision	81%
	F-Score	84%
ROUGE-2- Overlap of the bigrams	Recall	80%
	Precision	77%
	F-Score	79%
ROUGE-L- Longest Common Subsequence(LCS)	Recall	73%
	Precision	73%
	F-Score	73%

The following were the metrics obtained from an existing related abstractive summarizer. The evaluation process involved use of the same document (text **x**) to generate a summary then the metrics of the output were obtained. The accuracy that was recorded is as shown in Table 5 below.

Table 5: Accuracy of Summary Y evaluation Using Method 2 (extractive based)

LSTM extractive based Model		
Rouge Variant	**Metric**	**Accuracy**
ROUGE-1- Compares the overlap of unigrams	Recall	85%
	Precision	80%
	F-Score	82%
ROUGE-2- Overlap of the bigrams	Recall	79%
	Precision	76%
	F-Score	78%
ROUGE-L- Longest Common Subsequence(LCS)	Recall	60%
	Precision	62%
	F-Score	61%

Further comparison was made between the automated cybersecurity briefing(ACB) model and other existing extractive based summarizers (lexRank &Resoomer). The ACB outperformed these two summarizers when same original summary was loaded to these three summarizers and the results(summary) evaluated. Table 6 below contains the evaluation results.

Table 6:Comparing the accuracy of the cybersecurity briefing model with other existing related solutions (method 1)

Rouge Variant	Metric	Accuracy in percentage		
		ACB Model	**lexRank**	**Resoomer**
ROUGE-1 Compares the overlap of unigrams	Recall	86%	77%	77%
	Precision	81%	77%	78%
	F-Score	84%	77%	77%
ROUGE-2- Overlap of the bigrams	Recall	80%	64%	65%
	Precision	77%	64%	65%
	F-Score	79%	64%	64%
ROUGE-L- Longest Common Subsequence(LCS)	Recall	84%	72%	72%
	Precision	86%	72%	73%
	F-Score	84%	72%	72%

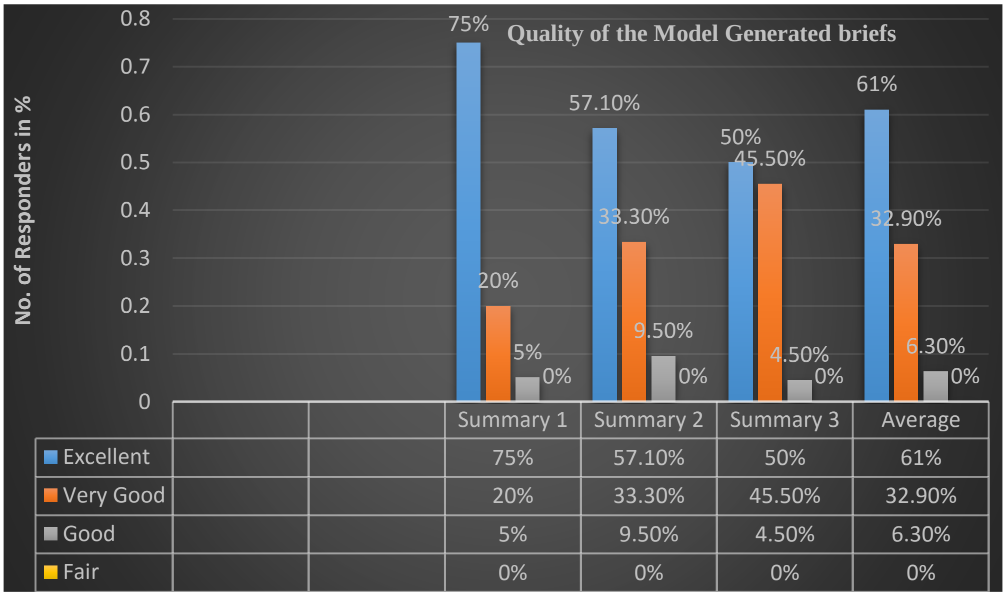
4.2.1 Industry Experts Validation on the results

Additionally, use of industry experts to validate the quality of briefs generated was employed. Out of the total number of respondents (60 cybersecurity analysts), 75% confirmed that cybersecurity briefing automation will be very important significant in the cybersecurity business whereas 20% agreed that it is important, and 5% agreed that it is somehow important.

Another aspect of concern was whether automating cyber briefing has a positive impact on prompt strategic decision making process where 68.2 percent of the total number of respondents confirmed that if the daily briefing on cybersecurity is automated, then the effectiveness of strategic cybersecurity decision making process will be improved to a great extent while 31.8 % said that the improvement will be somewhat.

The major goal of this industry expert results validation was to assess the model generated **brief's quality**. Out of the total number of cybersecurity analysts who provided input, 68.2% agreed that the briefs were excellent, 18.2% said it was very good, and 13.6 percent said it was good. A graphical representation is as shown in figure 10 below:

Figure 13: Validation of the briefs generated by the industry experts



			Summary 1	Summary 2	Summary 3	Average
🟦 Excellent			75%	57.10%	50%	61%
🟧 Very Good			20%	33.30%	45.50%	32.90%
■ Good			5%	9.50%	4.50%	6.30%
🟨 Fair			0%	0%	0%	0%

Figure 14: Validation of the briefs generated by the industry experts

4.3. Achievements

The overall objective was to develop a model that would generate cybersecurity briefs from cybersecurity reports received from various sources. The research has generated a working model that is able to produce summaries/briefs given a lengthy file/text through deep learning with a fairly acceptable performance accuracy. As compared to other summarization models, the model performs relatively well in the cybersecurity context. With continuous training of the model, its performance would surpass the expected performance while reducing the error rate to as low as possible. Through word vectorization and feature extraction, the Model was able to learn the patterns of brief generation in order to produce a summary through abstractive method given an input i.e. reports that requires summarization. Transfer learning using trained model was also used to further extract feature characteristics during learning. During training, word vectors are updated accordingly, optimized, and iterated over multiple epochs to minimize the loss function. Recurrent Oriented Understudy Gisting Evaluation was also assessed and used to measure the level of the model's performance.

4.4 Limitations

Training data was expensive and scarce to find. It was also not easy to determine what a good summary was and the best evaluator to be used for effectiveness measurement. The outstanding limitation in summarization process was a near absence of a universally accepted metric for evaluating summarization systems. This is because the evaluation of a summary is subjective since it entails judgements like readability, style, coherence and completeness.

Abstractive summarization methods can compress long texts more strongly compared to the extractive methods. However, coming up with abstractive programs is not easy since the usage of the required natural language processing techniques in the development process are still growing.

4.5 Conclusion

Abstractive summarization process based on LSTM Recurrent Neural Network performed better compared to other extractive summarization methods such as lexRank.

Despite ROUGE being widely adopted in evaluating text summarization due to its ability to correlate well with human judgements, it has been proven to be biased towards surface lexical similarities. Its therefore not suitable for evaluating summaries that have been significantly paraphrased or abstractive summarization. In the future, more works needs to be done on evaluating paraphrased summarizations.

Abstractive text summarization is based semantic text understanding, which means that the final summary is not strictly limited to the words in the original text source. As a result, metrics such as ROUGE-AR that employ use of latent semantic analysis(LSA) and part of speech tagging through incorporating anaphor resolution methods among other intrinsic methods will perform better when used to evaluate abstractive summaries. In the future, more works needs to be done on evaluating paraphrased summarizations.

The absence of a given human standard or automatic summarization evaluation metric in summarization research makes it very hard to compare different systems and establish a baseline.

Based on the above results, it can be concluded that the metric/measure to employ is determined on the task you're attempting to evaluate/measure. ROUGE-1 and ROUGE-2 may be useful if you're working on extractive summarization using a somewhat verbose system and reference summaries. ROUGE-L alone may be sufficient for very brief summaries, especially if stemming and stop word removal are used.

4.6. Future Work

There is need to acquire more data for effective brief/summary generation. More feature extraction techniques should be applied in order to further enhance the model accuracy and reduce loss as a result of incorrect prediction. The brief/summary generation can also be made real-time in order to make decision support through briefs timely and accurate. Due to voluminous nature of the dataset used to train the model, which requires more computing resources, the model can be hosted in a cloud platform for faster processing.

4.7Acknowledgment

The authors wish to supervision panel of University of Nairobi, Dr. Chepken Christopher, colleagues and family for their immense support, guidance and encouragement during all this time.

5 REFERENCES

1. Aone, C., Okurowski, M. E., Gorlinsky, J., & Larsen, B. (1997). A scalable summarization system using robust NLP. In *Intelligent Scalable Text Summarization*.

2. Berndtsson, M., Hansson, J., Olsson, B., & Lundell, B. (2008). Thesis Projects. In *Thesis Projects*. https://doi.org/10.1007/978-1-84800-009-4

3. Bridges, R. A., Huffer, K. M., Jones, C. L., Iannacone, M. D., & Goodall, J. R. (2017, December). Cybersecurity Automated Information Extraction Techniques: Drawbacks of Current Methods, and Enhanced Extractors. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 437-442). IEEE.

4. Carrasco, J. M. G., & Rubio, F. M. Methodology and Scientific Documentation.

5. Christensen, L. B., Johnson, B., Turner, L. A., & Christensen, L. B. (2011). Research methods, design, and analysis.

6. Demeyer, S. (2011). Research methods in computer science. *IEEE International Conference on Software Maintenance, ICSM, March*, 600. https://doi.org/10.1109/ICSM.2011.6080841

7. DIVISION, C. O. T. C. P. A. S. (2018). *National Security Agency Cybersecurity Report*. *June*, 1–8.

8. Dr. Emily Hand, P. D. (2020). *Distilling Public Data from Multiple Sources for Cybersecurity Applications* [University of Nevada]. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiHrKus2ZHyAhU0EWMBHdlcAWMQFjAAegQIAxAD&url=https%3A%2F%2Fscholarworks.unr.edu%2Fhandle%2F11714%2F7450&usg=AOvVaw2KYfr8Fya0ibOtxfNUac98

9. Fanfani, M. (2003). Editing. In *Lingua Nostra* (Vol. 64, Issues 1–2). https://doi.org/10.5840/radphilrev20013213.

10. Fors-Isalguez, Y., Hermosillo-Valadez, J., & Montes-Y-Gómez, M. (2018). Query-oriented text summarization based on multiobjective evolutionary algorithms and word embeddings. *Journal of Intelligent and Fuzzy Systems*, *34*(5), 3235–3244. https://doi.org/10.3233/JIFS-169506.

11. Higginbotham, D. J. (2000). Formulating research questions. *The efficacy of augmentative and alternative communication: Toward evidence-based practice*, 43-57.

12. How to write a research methodology. (n.d.).https://www.scribbr.com/dissertation/methodology/.

13. Inshorts Pte. ltd. (2021). *Inshorts*. http://www.inshorts.com/.

14. Jangra, J., Khatri, A., & Ralen, J. (2018). *A novel approach based on text summarization for Online Hotel Review*. *Iccs*, 325–329.

15. Kofod-Petersen, A. (2012). How to do a structured literature review in computer science. *Ver. 0.1. October*, *1*.

16. Neto, J. L., Freitas, A. A., & Kaestner, C. A. A. (n.d.). *Automatic Text Summarization using a Machine Learning Approach*. *i*.

17. Patil, N. R. (2017). *Automatic Text Summarization with Cohesion Features*. *8*(2), 194–198.

18. Prabhala, B. (2014). *Scalable Multi-Document Summarization Using Natural Language Processing*.

19. Rahul Lahkar, A. K. B. (2015). *Various Methodologies of Automatic Text Summarization*. https://www.ijert.org/a-survey-on-various-methodologies-of-automatic-text-summarization.

20. Ramanujam, N., & Kaliappan, M. (2016). Based on Naive Bayesian Classifier Using Timestamp Strategy. *The Scientific World Journal, Hindawi Publishing Corporation*, *2016*, 10.

21. Randolph, J., & Randolph, J. J. (2009). *A Guide to Writing the Dissertation Literature Review A Guide to Writing the Dissertation Literature Review*. *14*.

22. Robertson, J., Harrison, B., & Jhala, A. (2020). Interactive summarization for data filtering and triage. Proceedings of the 33rd International Florida Artificial Intelligence Research Society Conference, FLAIRS 2020, 252–257.

23. *S0885230820300760 @ www.sciencedirect.com*. (n.d.). https://www.sciencedirect.com/science/article/abs/pii/S0885230820300760.

24. Satyapanich, T., Finin, T., & Ferraro, F. (2019). Extracting Rich Semantic Information about Cybersecurity Events. *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019*, 5034–5042. https://doi.org/10.1109/BigData47090.2019.9006444**.**

25. Smart Africa. (2018). *Smart Africa*. *January*, 1–33. https://smartafrica.org/2019/IMG/pdf/brochure_-_smart_africa_agenda.pdf

26. *Survey: Only 39% of Orgs Have Ability to Retain Cyber Security Talent*. (n.d.). Retrieved June 22, 2021, from https://www.tripwire.com/state-of-security/featured/survey-only-39-of-orgs-have-ability-to-retain-cyber-security-talent/.

27. The, O. (2004). Guidelines for Writing a Successful MSc Thesis Proposal. *Science*, *June 2017*, 1–8.

28. Kavita Ganesan, P. (2020). *AI Implementation, ROUGE, Text Mining Concepts, Text Summarization*. http://kavita-ganesan.com/what-is-rouge-and-how-it-works-for-evaluation-of-summaries/#.W5LhLJNKidt/.

29. Thu, H. N. T., & Huu, Q. N. (2013). A semi-supervised learning method combined with dimensionality reduction in Vietnamese text summarization. *International Journal of Innovative Computing, Information and Control*, *9*(12), 4903–4915.

30. Du, Y. (2020). *Biomedical-domain pre-trained language model for extractive summarization*. https://doi.org/10.1016/j.knosys.2020.105964.

31. Cai, X. (n.d.). *HITS-based attentional neural model for abstractive summarization*. https://www.semanticscholar.org/paper/HITS-based-attentional-neural-model-for-abstractive-Cai-Shi/0e4e7816d10c01503295a7a18eef77c5eff64676.

32. Kumar, Y. (n.d.). *Leveraging Multimodality with Guided Attention for Abstractive Text*. https://www.semanticscholar.org/paper/See%2C-Hear%2C-Read%3A-Leveraging-Multimodality-with-for-Atri-Pramanick/5fd588411cad0bc8178f8920c4c28d14f4960122.

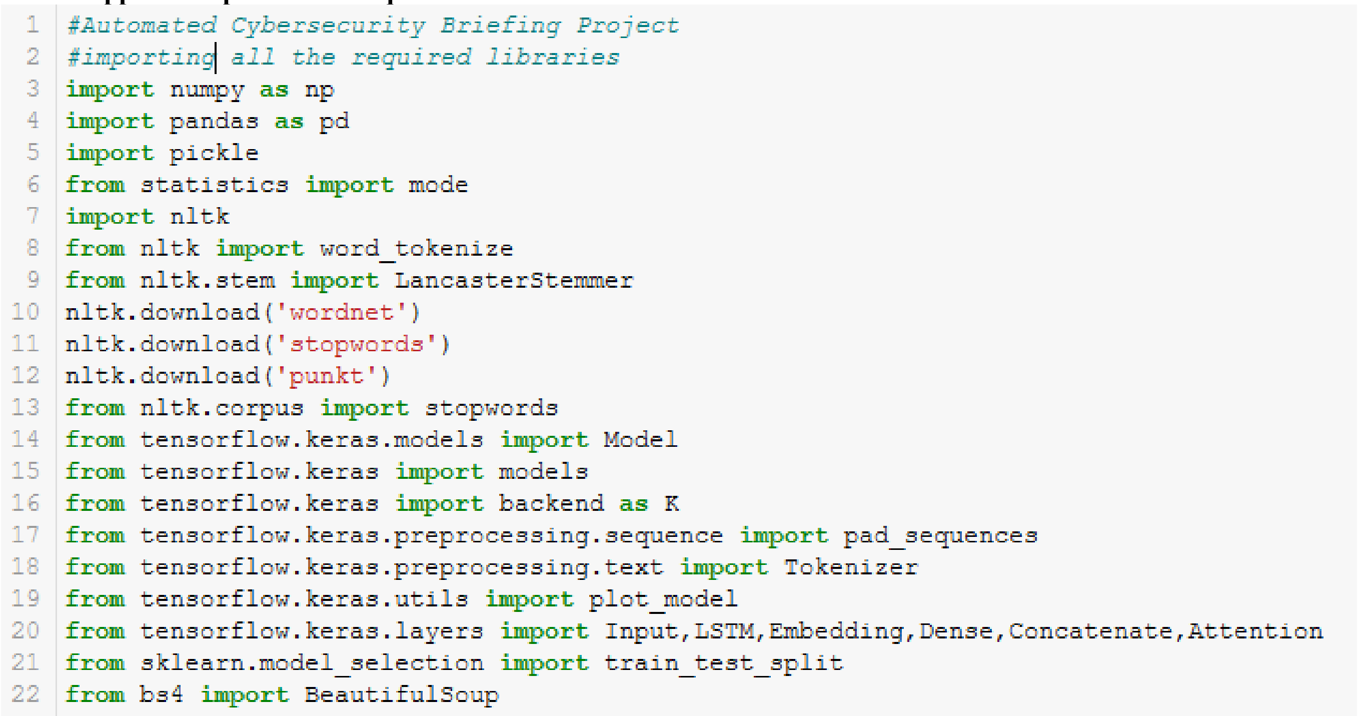
33. Moradi, M. (n.d.). *Deep contextualized embeddings for quantifying the informative content in biomedical text summarization*. https://www.semanticscholar.org/paper/Deep-contextualized-embeddings-for-quantifying-the-Moradi-Dorffner/3175467d02a809963a2aa0bf5b35789baf58365d.

34. Maples, S. (n.d.). *The ROUGE-AR : A Proposed Extension to the ROUGE Evaluation Metric for Abstractive Text Summarization*.

APPENDICES

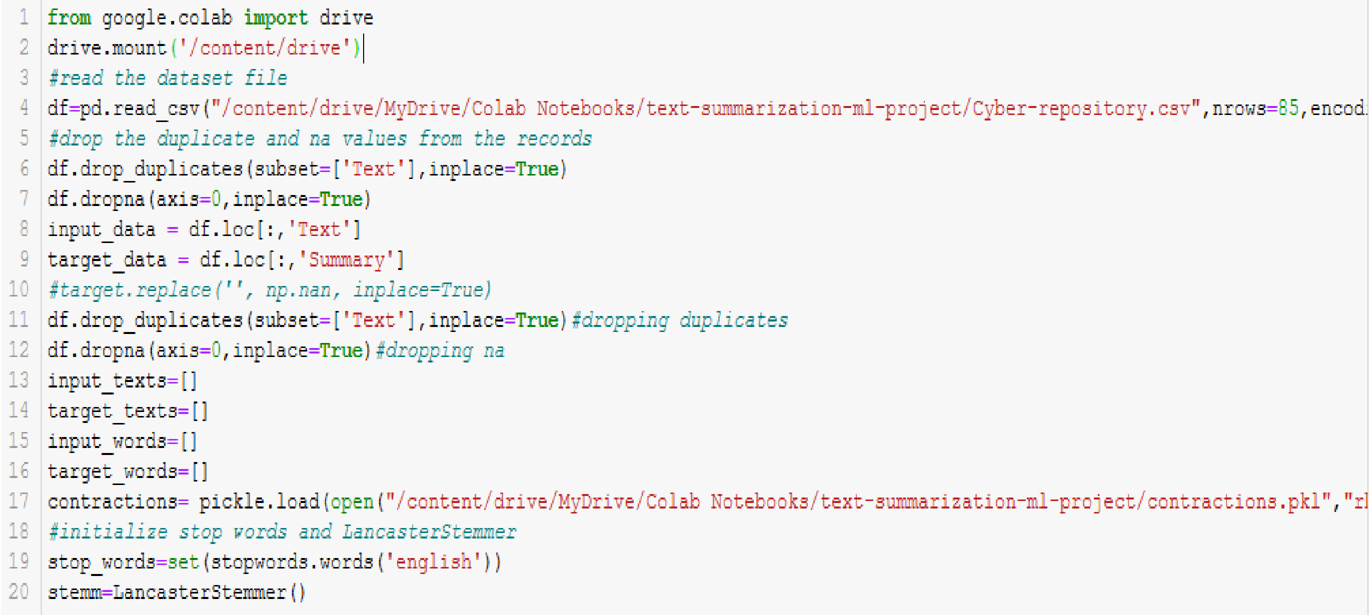
Appendix 1: Code Snippets

Code Snippet 1: Import all the required libraries



```python
#Automated Cybersecurity Briefing Project
#importing all the required libraries
import numpy as np
import pandas as pd
import pickle
from statistics import mode
import nltk
from nltk import word_tokenize
from nltk.stem import LancasterStemmer
nltk.download('wordnet')
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from tensorflow.keras.models import Model
from tensorflow.keras import models
from tensorflow.keras import backend as K
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.utils import plot_model
from tensorflow.keras.layers import Input,LSTM,Embedding,Dense,Concatenate,Attention
from sklearn.model_selection import train_test_split
from bs4 import BeautifulSoup
```

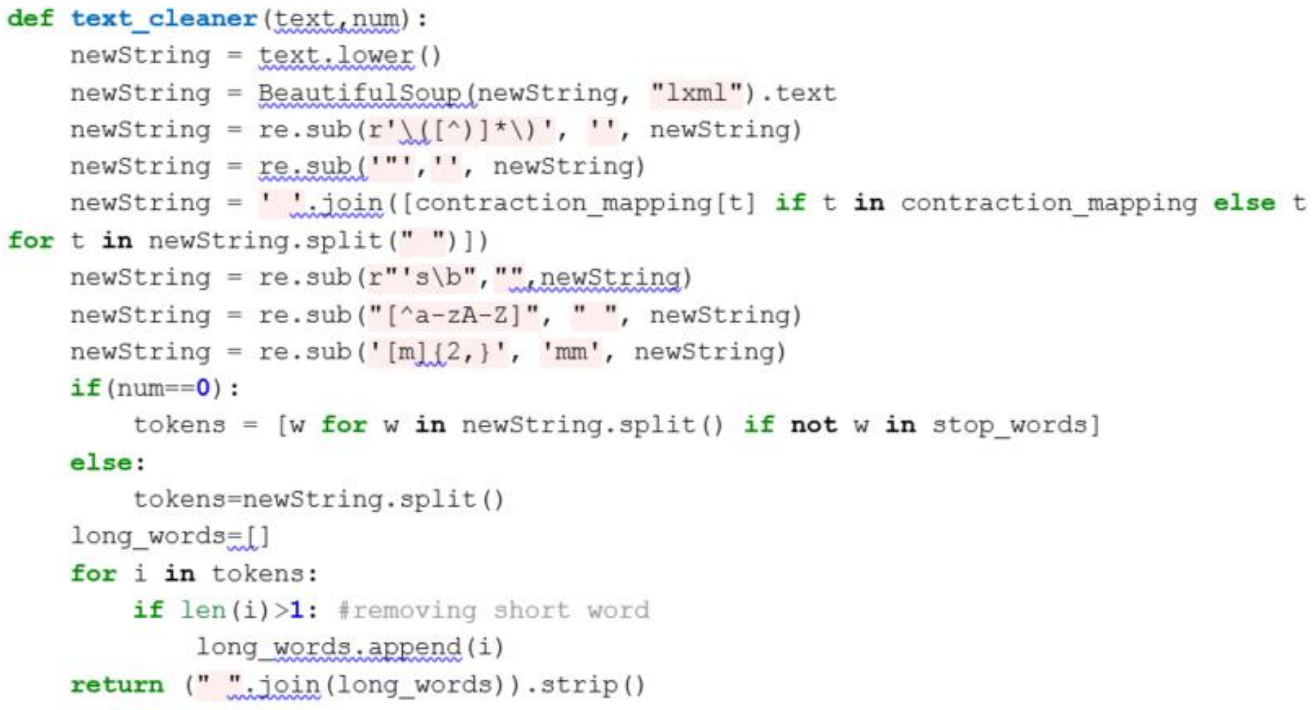
CodeSnippet 2: Reading/Loading the dataset



```python
from google.colab import drive
drive.mount('/content/drive')
#read the dataset file
df=pd.read_csv("/content/drive/MyDrive/Colab Notebooks/text-summarization-ml-project/Cyber-repository.csv",nrows=85,encod
#drop the duplicate and na values from the records
df.drop_duplicates(subset=['Text'],inplace=True)
df.dropna(axis=0,inplace=True)
input_data = df.loc[:,'Text']
target_data = df.loc[:,'Summary']
#target.replace('', np.nan, inplace=True)
df.drop_duplicates(subset=['Text'],inplace=True)#dropping duplicates
df.dropna(axis=0,inplace=True)#dropping na
input_texts=[]
target_texts=[]
input_words=[]
target_words=[]
contractions= pickle.load(open("/content/drive/MyDrive/Colab Notebooks/text-summarization-ml-project/contractions.pkl","r
#initialize stop words and LancasterStemmer
stop_words=set(stopwords.words('english'))
stemm=LancasterStemmer()
```

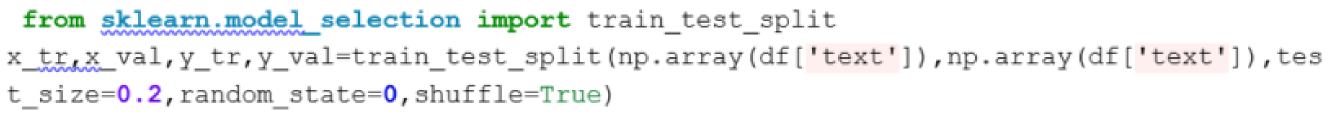
CodeSnippet 3: Data Cleaning

```
def clean(texts,src):
  #remove the html tags
  texts = BeautifulSoup(texts, "lxml").text
  #tokenize the text into words
  words=word_tokenize(texts.lower())
  #filter words which contains \
  #integers or their length is less than or equal to 3
  words= list(filter(lambda w:(w.isalpha() and len(w)>=3),words))
  #contraction file to expand shortened words
  words= [contractions[w] if w in contractions else w for w in words ]
  #stem the words to their root word and filter stop words
  if src=="inputs":
    words= [stemm.stem(w) for w in words if w not in stop_words]
  else:
    words= [w for w in words if w not in stop_words]
  return words
stop_words = set(stopwords.words('english'))
```



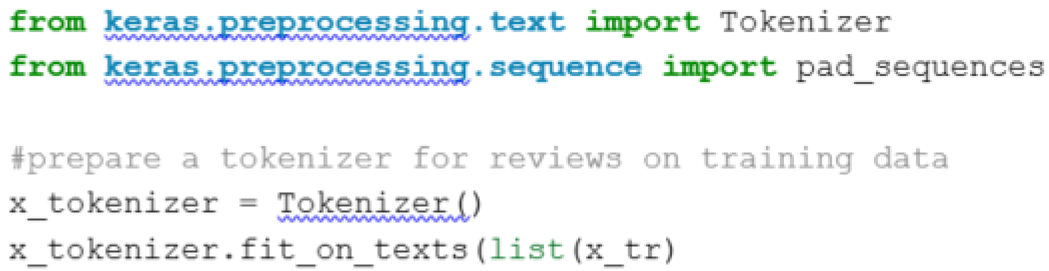
```
def text_cleaner(text,num):
    newString = text.lower()
    newString = BeautifulSoup(newString, "lxml").text
    newString = re.sub(r'\([^)]*\)', '', newString)
    newString = re.sub('"','', newString)
    newString = ' '.join([contraction_mapping[t] if t in contraction_mapping else t
for t in newString.split(" ")])
    newString = re.sub(r"'s\b","",newString)
    newString = re.sub("[^a-zA-Z]", " ", newString)
    newString = re.sub('[m]{2,}', 'mm', newString)
    if(num==0):
        tokens = [w for w in newString.split() if not w in stop_words]
    else:
        tokens=newString.split()
    long_words=[]
    for i in tokens:
        if len(i)>1: #removing short word
            long_words.append(i)
    return (" ".join(long_words)).strip()
```

Code Snippet 4: Splitting the dataset into a training and validation set.



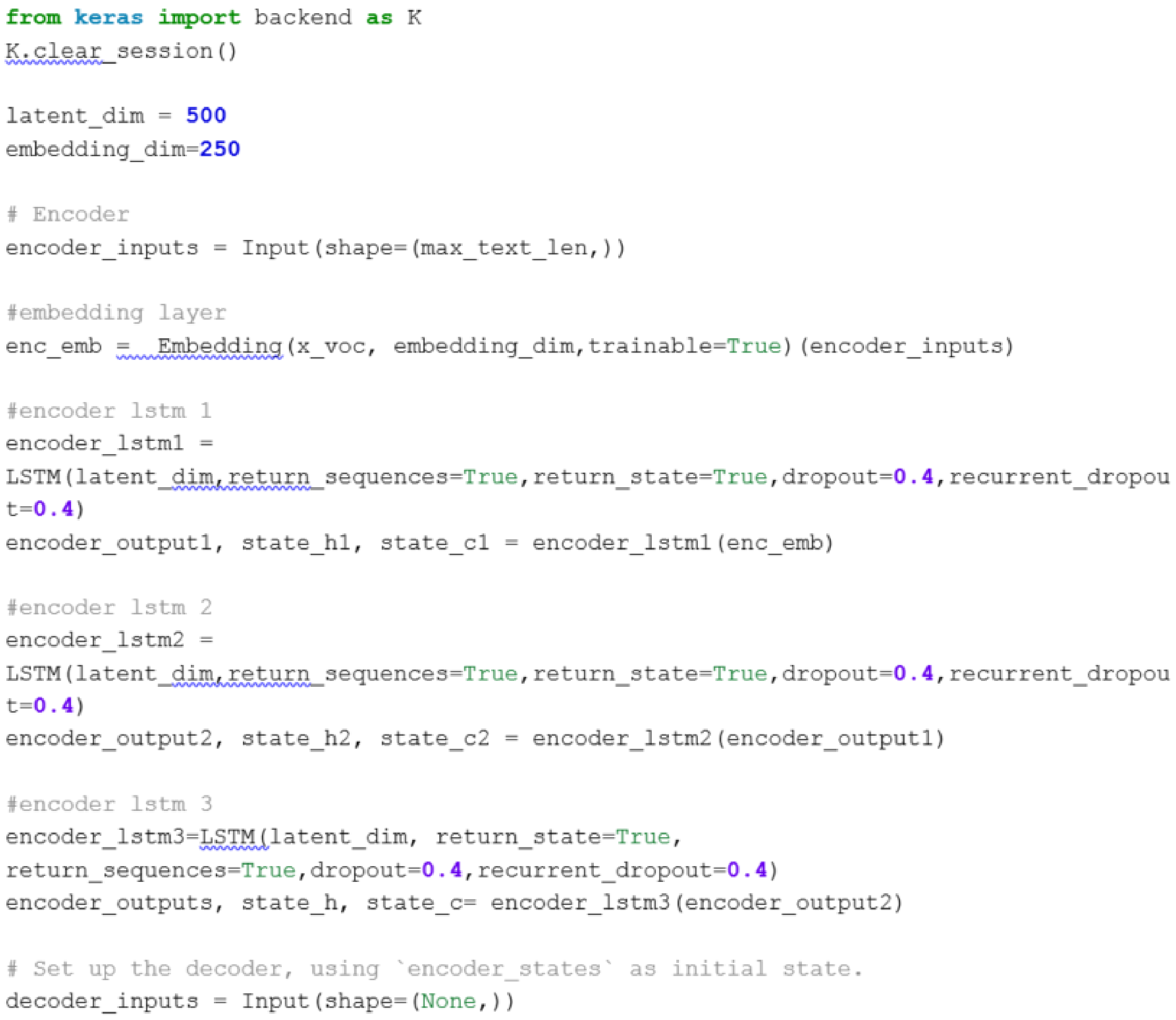
```
 from sklearn.model_selection import train_test_split
x_tr,x_val,y_tr,y_val=train_test_split(np.array(df['text']),np.array(df['text']),tes
t_size=0.2,random_state=0,shuffle=True)
```

Code Snippet 5: Converting word sequence into an integer sequence using text tokenizer

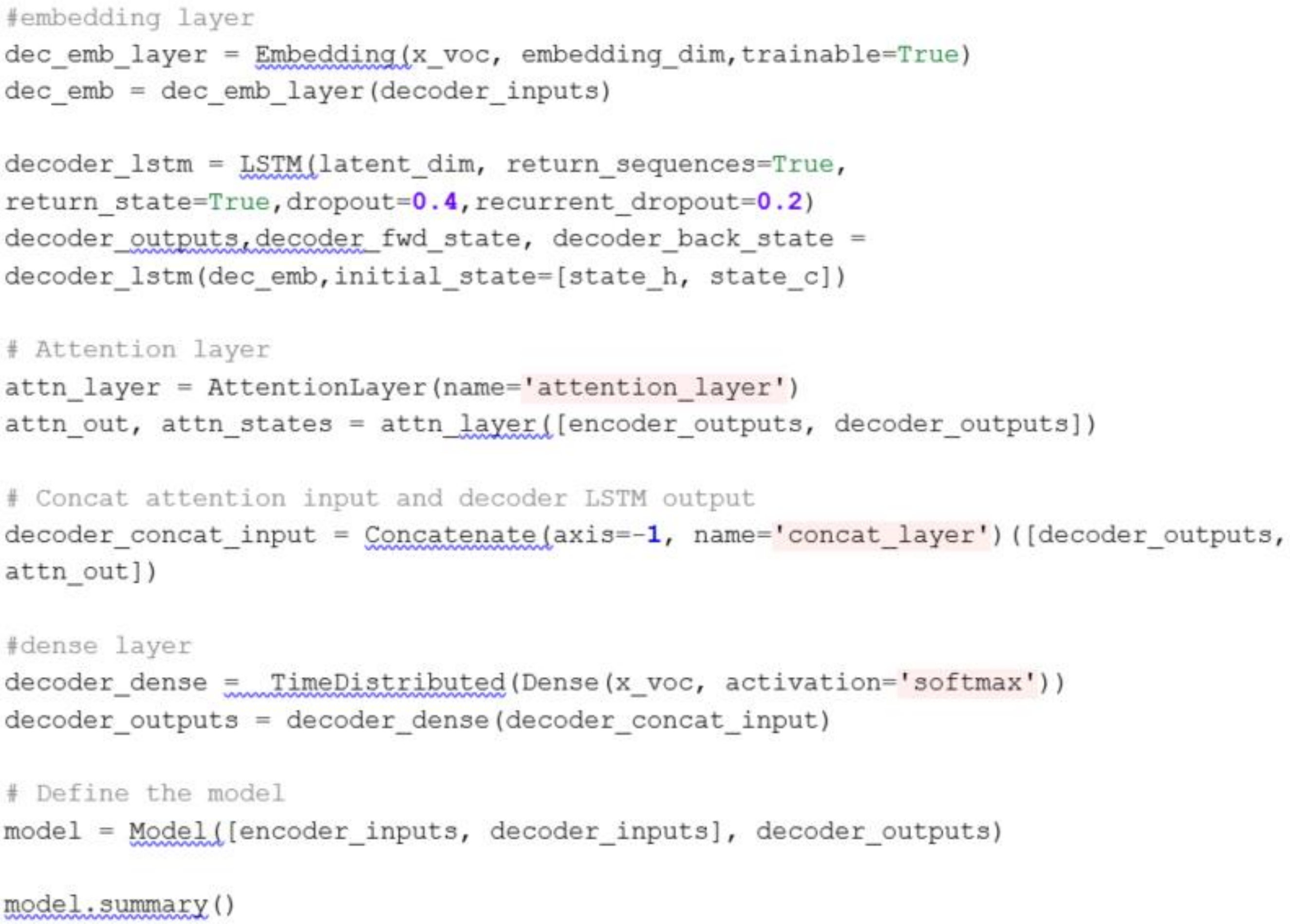


```
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences

#prepare a tokenizer for reviews on training data
x_tokenizer = Tokenizer()
x_tokenizer.fit_on_texts(list(x_tr))
```

Code Snippet 6: Building the Model

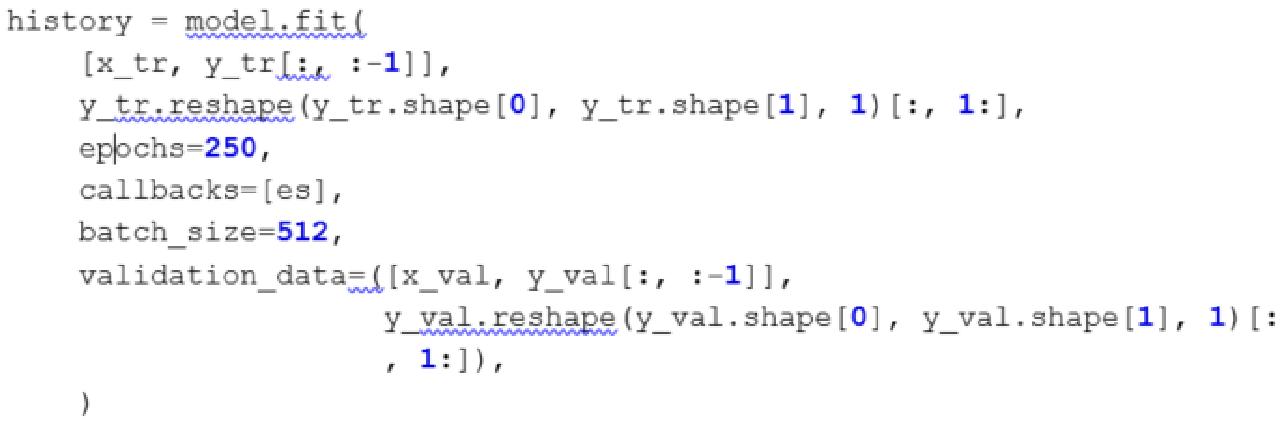


```
from keras import backend as K
K.clear_session()

latent_dim = 500
embedding_dim=250

# Encoder
encoder_inputs = Input(shape=(max_text_len,))

#embedding layer
enc_emb =  Embedding(x_voc, embedding_dim,trainable=True)(encoder_inputs)

#encoder lstm 1
encoder_lstm1 =
LSTM(latent_dim,return_sequences=True,return_state=True,dropout=0.4,recurrent_dropou
t=0.4)
encoder_output1, state_h1, state_c1 = encoder_lstm1(enc_emb)

#encoder lstm 2
encoder_lstm2 =
LSTM(latent_dim,return_sequences=True,return_state=True,dropout=0.4,recurrent_dropou
t=0.4)
encoder_output2, state_h2, state_c2 = encoder_lstm2(encoder_output1)

#encoder lstm 3
encoder_lstm3=LSTM(latent_dim, return_state=True,
return_sequences=True,dropout=0.4,recurrent_dropout=0.4)
encoder_outputs, state_h, state_c= encoder_lstm3(encoder_output2)

# Set up the decoder, using `encoder_states` as initial state.
decoder_inputs = Input(shape=(None,))
```



```
#embedding layer
dec_emb_layer = Embedding(x_voc, embedding_dim,trainable=True)
dec_emb = dec_emb_layer(decoder_inputs)

decoder_lstm = LSTM(latent_dim, return_sequences=True,
return_state=True,dropout=0.4,recurrent_dropout=0.2)
decoder_outputs,decoder_fwd_state, decoder_back_state =
decoder_lstm(dec_emb,initial_state=[state_h, state_c])

# Attention layer
attn_layer = AttentionLayer(name='attention_layer')
attn_out, attn_states = attn_layer([encoder_outputs, decoder_outputs])

# Concat attention input and decoder LSTM output
decoder_concat_input = Concatenate(axis=-1, name='concat_layer')([decoder_outputs,
attn_out])

#dense layer
decoder_dense =  TimeDistributed(Dense(x_voc, activation='softmax'))
decoder_outputs = decoder_dense(decoder_concat_input)

# Define the model
model = Model([encoder_inputs, decoder_inputs], decoder_outputs)

model.summary()
```

Code Snippet 7: Fitting the model using a batch size of 512 on 250 epochs

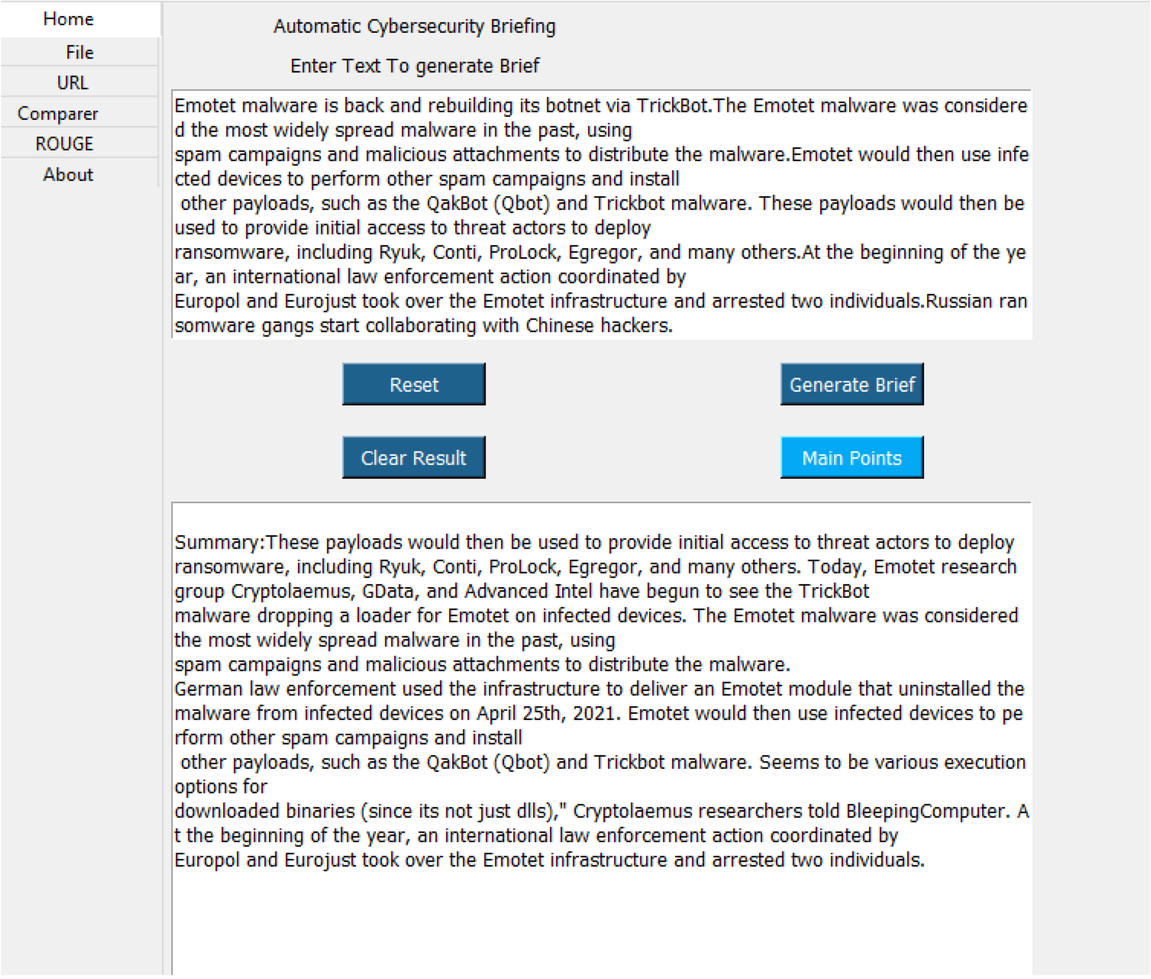


```
history = model.fit(
    [x_tr, y_tr[:, :-1]],
    y_tr.reshape(y_tr.shape[0], y_tr.shape[1], 1)[:, 1:],
    epochs=250,
    callbacks=[es],
    batch_size=512,
    validation_data=([x_val, y_val[:, :-1]],
                     y_val.reshape(y_val.shape[0], y_val.shape[1], 1)[:
                     , 1:]),
    )
```

Code Snippet 8: Inference function for encoder and decoder process.

```
def decode_sequence(input_seq):
    # Encode the input as state vectors.
    e_out, e_h, e_c = encoder_model.predict(input_seq)

    # Generate empty target sequence of length 1.
    target_seq = np.zeros((1,1))

    # Populate the first word of target sequence with the start word.
    target_seq[0, 0] = target_word_index['sostok']

    stop_condition = False
    decoded_sentence = ''
    while not stop_condition:

        output_tokens, h, c = decoder_model.predict([target_seq] + [e_out, e_h,
e_c])

        # Sample a token
        sampled_token_index = np.argmax(output_tokens[0, -1, :])
        sampled_token = reverse_target_word_index[sampled_token_index]

        if(sampled_token!='eostok'):
            decoded_sentence += ' '+sampled_token

        # Exit condition: either hit max length or find stop word.
        if (sampled_token == 'eostok'  or len(decoded_sentence.split()) >=
(max_summary_len-1)):
            stop_condition = True

        # Update the target sequence (of length 1).
        target_seq = np.zeros((1,1))
        target_seq[0, 0] = sampled_token_index

        # Update internal states
        e_h, e_c = h, c
    return decoded_sentence
```

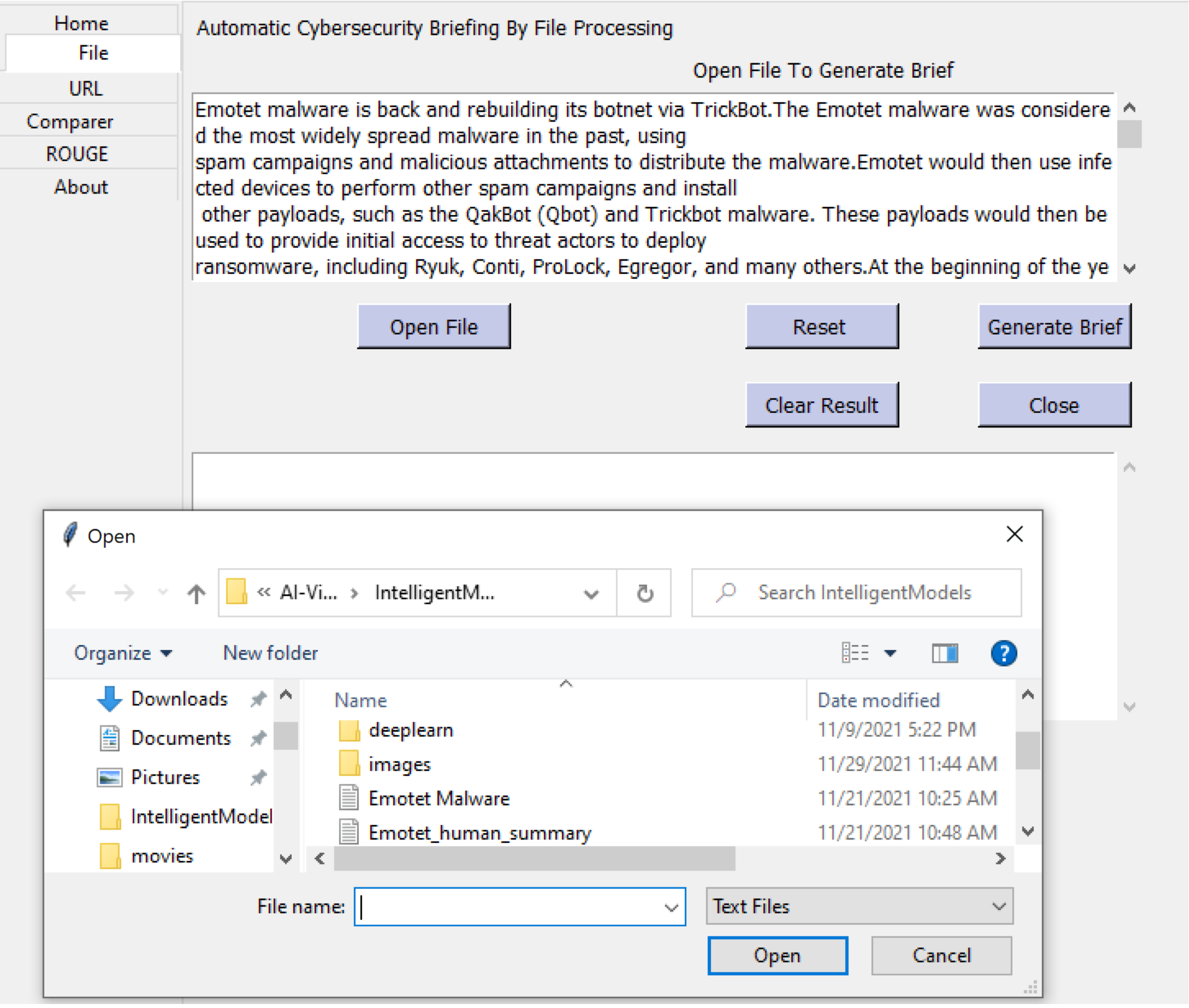
Appendix 2: User Interface



Emotet malware is back and rebuilding its botnet via TrickBot.The Emotet malware was considered the most widely spread malware in the past, using
spam campaigns and malicious attachments to distribute the malware.Emotet would then use infected devices to perform other spam campaigns and install
other payloads, such as the QakBot (Qbot) and Trickbot malware. These payloads would then be used to provide initial access to threat actors to deploy
ransomware, including Ryuk, Conti, ProLock, Egregor, and many others.At the beginning of the year, an international law enforcement action coordinated by
Europol and Eurojust took over the Emotet infrastructure and arrested two individuals.Russian ransomware gangs start collaborating with Chinese hackers.

Summary:These payloads would then be used to provide initial access to threat actors to deploy
ransomware, including Ryuk, Conti, ProLock, Egregor, and many others. Today, Emotet research
group Cryptolaemus, GData, and Advanced Intel have begun to see the TrickBot
malware dropping a loader for Emotet on infected devices. The Emotet malware was considered
the most widely spread malware in the past, using
spam campaigns and malicious attachments to distribute the malware.
German law enforcement used the infrastructure to deliver an Emotet module that uninstalled the
malware from infected devices on April 25th, 2021. Emotet would then use infected devices to perform other spam campaigns and install
other payloads, such as the QakBot (Qbot) and Trickbot malware. Seems to be various execution
options for
downloaded binaries (since its not just dlls)," Cryptolaemus researchers told BleepingComputer. At the beginning of the year, an international law enforcement action coordinated by
Europol and Eurojust took over the Emotet infrastructure and arrested two individuals.

Figure 15: User interface for generating a brief from lengthy text



Automatic Cybersecurity Briefing By File Processing

Open File To Generate Brief

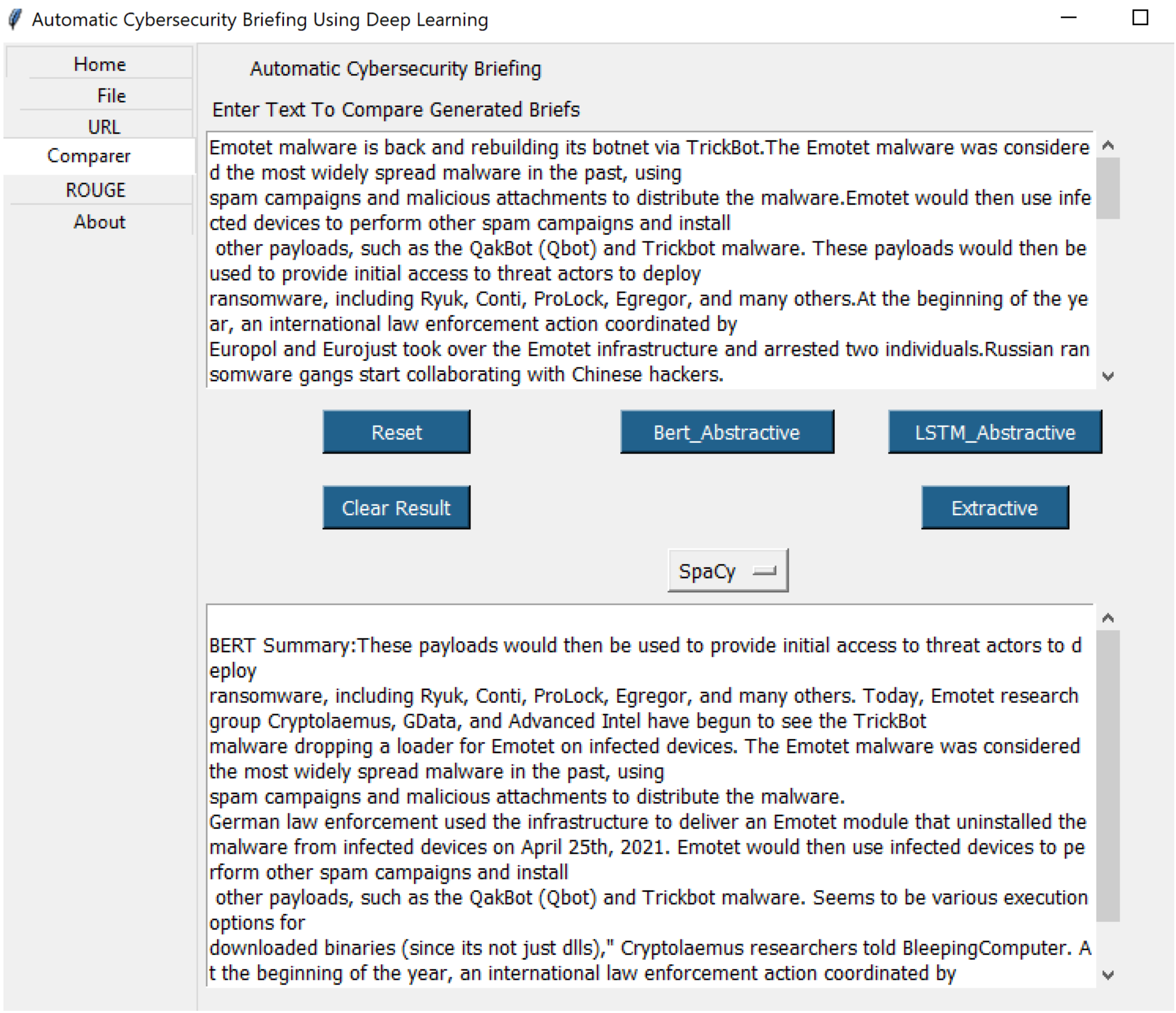
Emotet malware is back and rebuilding its botnet via TrickBot.The Emotet malware was considere
d the most widely spread malware in the past, using
spam campaigns and malicious attachments to distribute the malware.Emotet would then use infe
cted devices to perform other spam campaigns and install
 other payloads, such as the QakBot (Qbot) and Trickbot malware. These payloads would then be
used to provide initial access to threat actors to deploy
ransomware, including Ryuk, Conti, ProLock, Egregor, and many others.At the beginning of the ye

Name	Date modified
deeplearn	11/9/2021 5:22 PM
images	11/29/2021 11:44 AM
Emotet Malware	11/21/2021 10:25 AM
Emotet_human_summary	11/21/2021 10:48 AM

File name:

Text Files

Figure 16: Generating a brief from website URLs

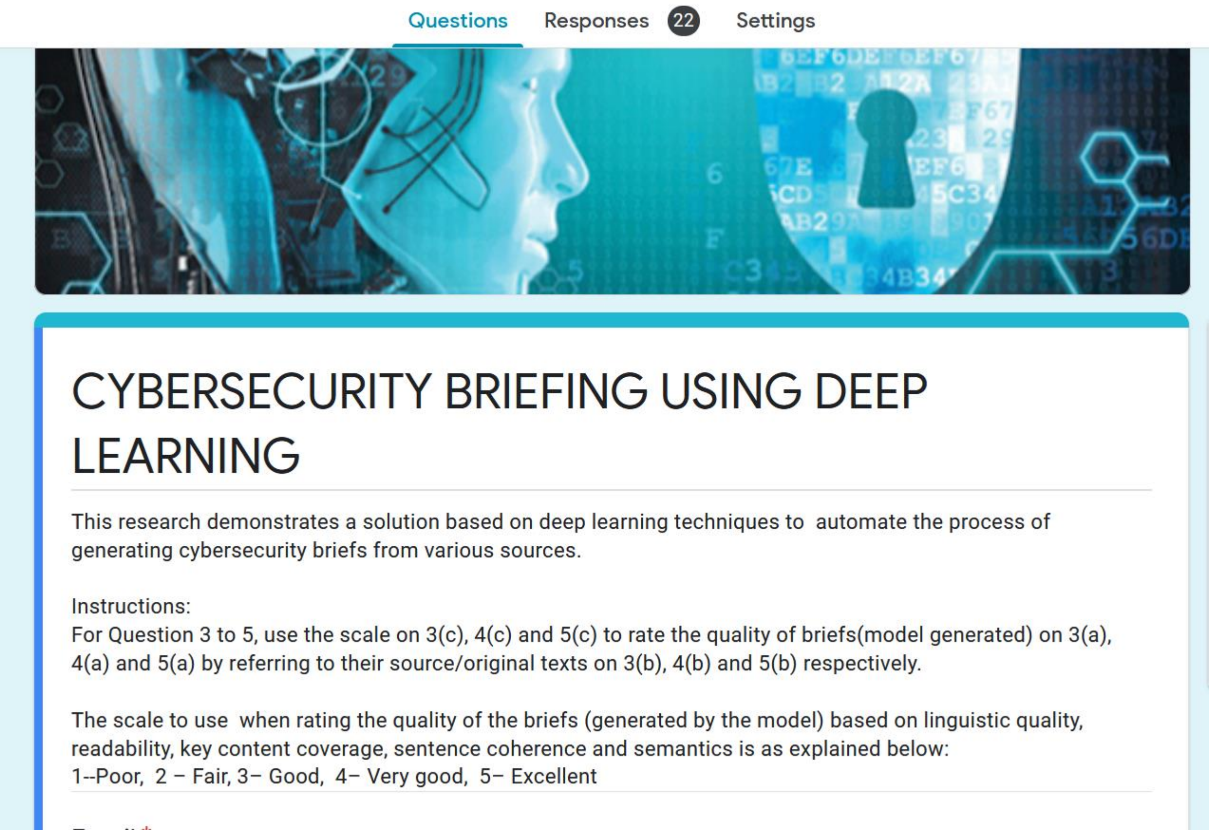

Automatic Cybersecurity Briefing Using Deep Learning
Home
File
URL
Comparer
ROUGE
About
Automatic Cybersecurity Briefing
Enter Text To Compare Generated Briefs
Reset
Bert_Abstractive
LSTM_Abstractive
Clear Result
Extractive
SpaCy

```
Emotet malware is back and rebuilding its botnet via TrickBot.The Emotet malware was considere
d the most widely spread malware in the past, using
spam campaigns and malicious attachments to distribute the malware.Emotet would then use infe
cted devices to perform other spam campaigns and install
 other payloads, such as the QakBot (Qbot) and Trickbot malware. These payloads would then be
used to provide initial access to threat actors to deploy
ransomware, including Ryuk, Conti, ProLock, Egregor, and many others.At the beginning of the ye
ar, an international law enforcement action coordinated by
Europol and Eurojust took over the Emotet infrastructure and arrested two individuals.Russian ran
somware gangs start collaborating with Chinese hackers.
```

```
BERT Summary:These payloads would then be used to provide initial access to threat actors to d
eploy
ransomware, including Ryuk, Conti, ProLock, Egregor, and many others. Today, Emotet research
group Cryptolaemus, GData, and Advanced Intel have begun to see the TrickBot
malware dropping a loader for Emotet on infected devices. The Emotet malware was considered
the most widely spread malware in the past, using
spam campaigns and malicious attachments to distribute the malware.
German law enforcement used the infrastructure to deliver an Emotet module that uninstalled the
malware from infected devices on April 25th, 2021. Emotet would then use infected devices to pe
rform other spam campaigns and install
 other payloads, such as the QakBot (Qbot) and Trickbot malware. Seems to be various execution
options for
downloaded binaries (since its not just dlls)," Cryptolaemus researchers told BleepingComputer. A
t the beginning of the year, an international law enforcement action coordinated by
```

Figure 17:Comparing the abstractive and extractive summaries

Appendix 3: Questionnaire to Gather Cybersecurity Analysts' Views on the Quality of Model Generated Briefs



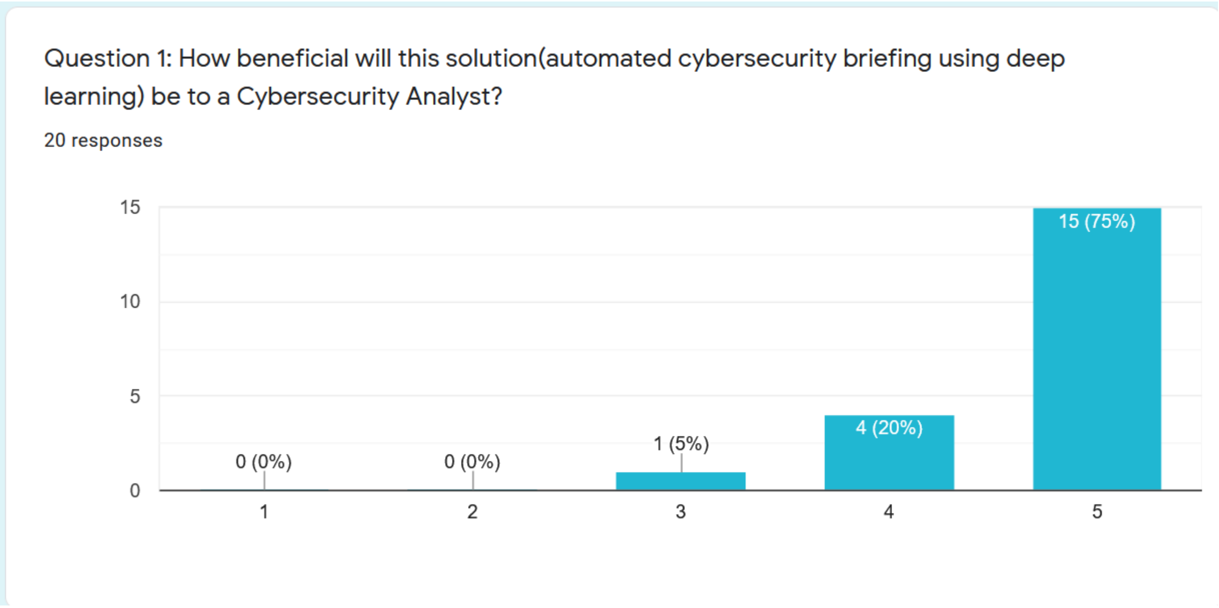
CYBERSECURITY BRIEFING USING DEEP LEARNING

This research demonstrates a solution based on deep learning techniques to automate the process of generating cybersecurity briefs from various sources.

Instructions:
For Question 3 to 5, use the scale on 3(c), 4(c) and 5(c) to rate the quality of briefs(model generated) on 3(a), 4(a) and 5(a) by referring to their source/original texts on 3(b), 4(b) and 5(b) respectively.

The scale to use when rating the quality of the briefs (generated by the model) based on linguistic quality, readability, key content coverage, sentence coherence and semantics is as explained below:
1--Poor, 2 – Fair, 3– Good, 4– Very good, 5– Excellent

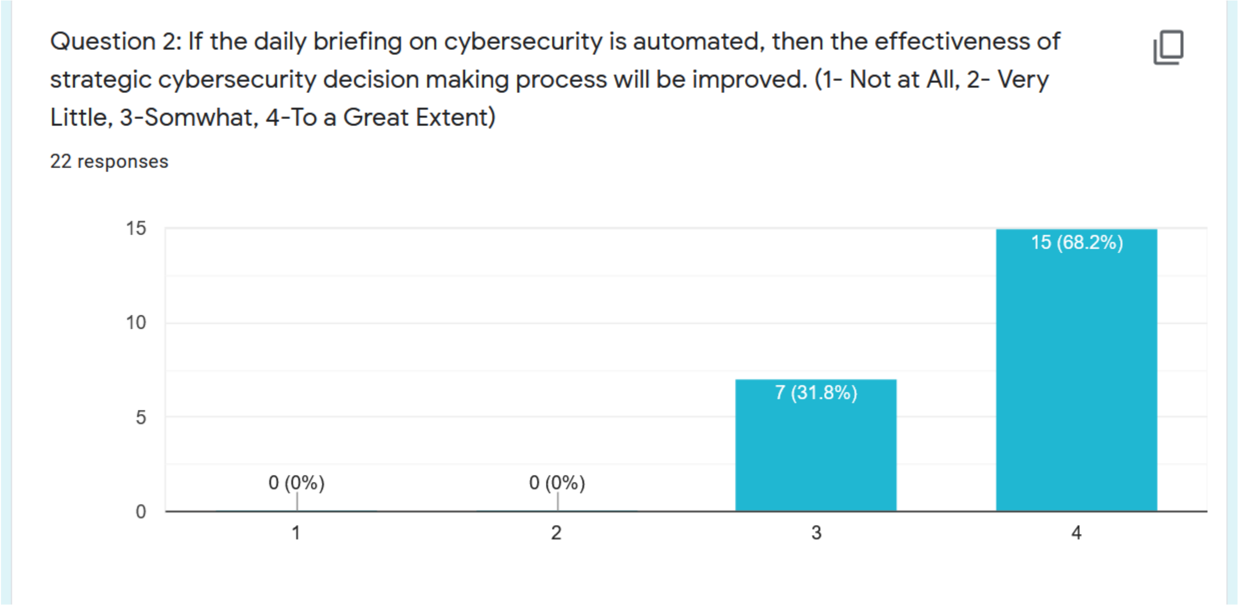
Figure 18: Survey questionnaire to collect cybersecurity analyst views on the quality of the model generated brief



Question 1: How beneficial will this solution(automated cybersecurity briefing using deep learning) be to a Cybersecurity Analyst?

20 responses

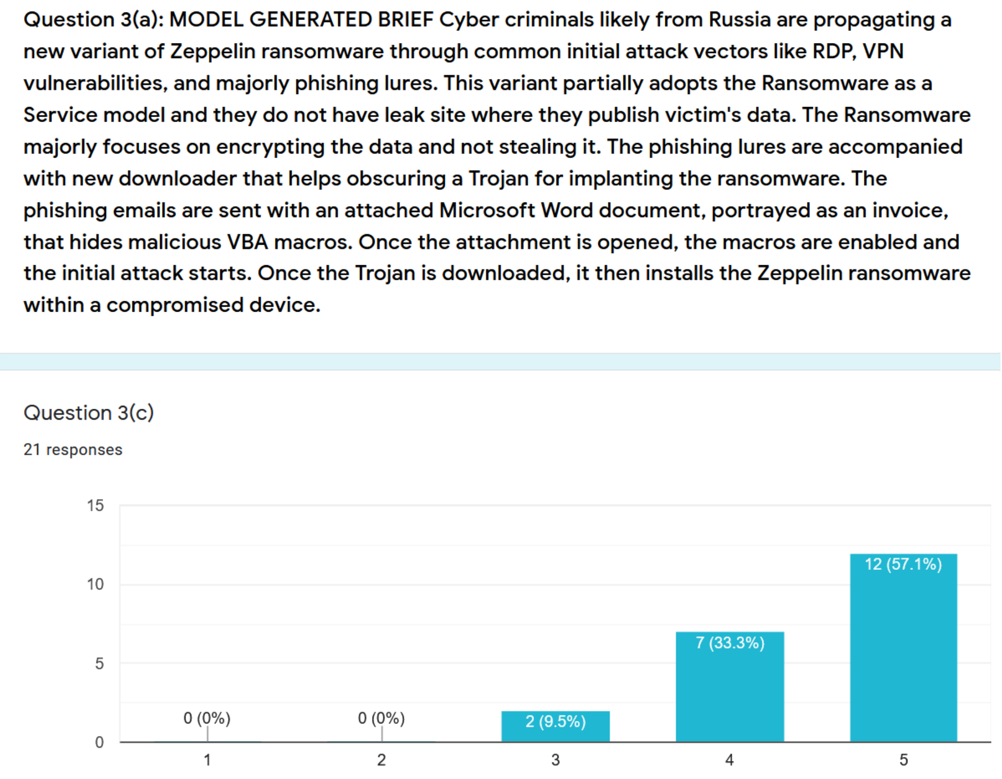
Figure 19: Question one to gather analyst views on the importance of automated cyber briefing



Question 2: If the daily briefing on cybersecurity is automated, then the effectiveness of strategic cybersecurity decision making process will be improved. (1- Not at All, 2- Very Little, 3-Somwhat, 4-To a Great Extent)

22 responses

Figure 20:Question two aimed at gauging the impact of automated cyber briefing on strategic decision making process

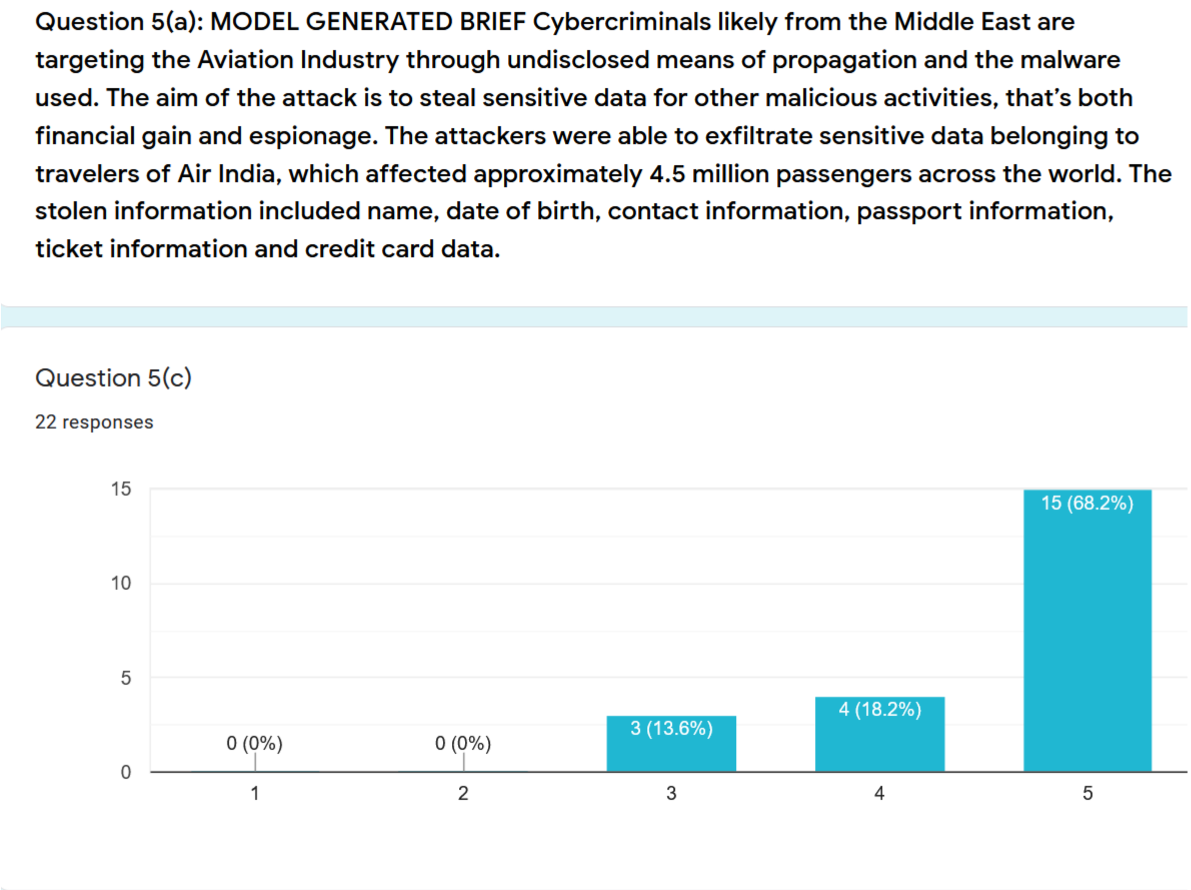


Question 3(a): MODEL GENERATED BRIEF Cyber criminals likely from Russia are propagating a new variant of Zeppelin ransomware through common initial attack vectors like RDP, VPN vulnerabilities, and majorly phishing lures. This variant partially adopts the Ransomware as a Service model and they do not have leak site where they publish victim's data. The Ransomware majorly focuses on encrypting the data and not stealing it. The phishing lures are accompanied with new downloader that helps obscuring a Trojan for implanting the ransomware. The phishing emails are sent with an attached Microsoft Word document, portrayed as an invoice, that hides malicious VBA macros. Once the attachment is opened, the macros are enabled and the initial attack starts. Once the Trojan is downloaded, it then installs the Zeppelin ransomware within a compromised device.

Question 3(c)

21 responses

Figure 21:Question three aimed at gathering cyber analsyts’ opinion on the quality of one of the generated cybersecurity brief



Question 5(a): MODEL GENERATED BRIEF Cybercriminals likely from the Middle East are targeting the Aviation Industry through undisclosed means of propagation and the malware used. The aim of the attack is to steal sensitive data for other malicious activities, that's both financial gain and espionage. The attackers were able to exfiltrate sensitive data belonging to travelers of Air India, which affected approximately 4.5 million passengers across the world. The stolen information included name, date of birth, contact information, passport information, ticket information and credit card data.

Question 5(c)

22 responses

Figure 22: Question five aimed at gauging the quality of one of the cybersecurity briefs by cyber analysts