Identification of Cyber Threats Information from Online News using Hybrid Machine Learning Algorithm

Kiflework Dinku Taye

A thesis Submitted to the department of Computer Science and Engineering,

School of Electrical Engineering and Computing

Office of Graduate Studies

Adama Science and Technology University

June, 2023

Adama, Ethiopia

Identification of Cyber Threats Information from Online News using Hybrid Machine Learning Algorithm

Kiflework Dinku Taye

Advisor: **Bahiru Shifaw (PhD)**

A thesis Submitted to the department of Computer Science and Engineering,

School of Electrical Engineering and Computing

Office of Graduate Studies

Adama Science and Technology University

June, 2023

Adama, Ethiopia

Declaration

I declare that this thesis/dissertation entitled "**Identification of Cyber Threats Information from Online News using Hybrid Machine Learning Algorithm"** is my own work and has not been submitted to any university for similar purpose. The references used in this proposal are duly recognized by proper citations.

_____	_____	_____
Name of student	Signature	Date

Certificate

I/we, the advisor(s) of this thesis, hereby certify that I have read the revised version of the thesis entitled "**Identification of Cyber Threats Information from Online News using Hybrid Machine Learning Algorithm**" prepared under my guidance by Kiflework Dinku Taye submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science and Engineering. Therefore, I recommend the submission of the revised version of the thesis to the department following the applicable procedures.

_____	_____	_____
Major Advisor/Supervisor	Signature	Date
_____	_____	_____
Co-advisor/Co-supervisor	Signature	Date

Approval Page for thesis

I/we hereby certify that the recommendation and suggestion given by the proposal review Committee are appropriately incorporated into the final thesis/dissertation entitled "**Identification of Cyber Threats Information from Online News using Hybrid Machine Learning Algorithm**" by Kiflework Dinku Taye.

Major Advisor/Supervisor	Signature	Date
Co-advisor/ Co-supervisor	Signature	Date

We, the undersigned, members of the Board of Reviewers of the proposal open defense by Kiflework Dinku Taye have read and evaluated the Thesis/dissertation entitled "**Identification of Cyber Threats Information from Online News using Hybrid Machine Learning Algorithm**" and assessed the understanding of the candidate about the proposed research. This is, therefore, to certify that the Thesis/dissertation proposal is accepted and we recommend the implementation of the Proposal.

Chairperson	Signature	Date
Internal examiner	Signature	Date
External examiner	Signature	Date

Final approval and acceptance of the thesis/dissertation proposal is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head	Signature	Date
School Dean	Signature	Date
Office of Postgraduate Studies, Dean	Signature	Date

Acknowledgment

In the first place, I would like to thanks the almighty God for helping me throughout my studies. Next I would like to express my gratitude towards my advisor **Dr.Bahiru Shifaw** for his invaluable guidance, support and encouragement while making this thesis work. I am grateful to my friends for their support and entertaining fun throughout this journey. I also wish to extend my thanks to computer science and engineering department head and all the members of this department for their cheering direction and for providing a good quality education which has enabled me to grow and develop professionally. Special thanks to Adama Science and Technology University, for giving me the opportunity to pursue my Master‘s degree.

Last but not the least; I would like to thank my pure hearts families, who have supported me in different ways during my study specially, to my wife she never stopped believing in me and supporting me during the journey of my life.

Abstract

There are large volumes of data from the online generated cyber threat news that are freely available which might contain valuable information. Cyber threat information is highly increasing and can be analysed to gather informative insights of current situation. However, news is delivered in a variety of forms, and the emergence of new cyber-attacks, as well as the usage of ambiguous news items, has made detecting linked news more challenging. Thus, to handle these situations, the aim of this paper is to propose identification mechanism for cyber threat information. The system starts with identifying cyber-attack features that will be used to classify cyber threat information using Bi-directional long short time memory with Conditional Random filed (BI-LSTM-CRF) model, as well as categorize similar news articles using Latent Semantic Analysis (LSA) to eliminate ambiguous cyber threat news. Data will be collected from the news article related to the cyber-attacks that include incidents or attacks that had happened. Data can be obtained from the news websites such as Recorded Future.com, Fire Eye, Security Week, Micro Trend and Ethiopian Monitor Website. The cyber-attack features will be identified from the collected data. Futures such as type of cyber-threat, threat actor, the organization affect and Country affect. For this research 2019 cyber related news articles are collected in the form of unstructured text. Experimental results demonstrate that using Bi-directional long short time memory with Conditional Random filed is an effective way of classification performance. The model achieves an overall F-measure of 98.48% for Cyber threat information identification with accuracy 99.12%. The findings of this study should assist individuals by presenting a realistic picture of cyber-attack occurrences in our environment and providing useful information to the general public, thereby improving societal awareness about cyber-attack activities. In addition, the model requires further improvement regarding feature selection of the cyber -attack that would be difficult for machine to catagorized.

Keywords *— Conditional Random field, Bi-directional Long short Time Memory, LSA cyber-attack features, Cyber threat information*

Table of Contents

Abstract V

List of Tables VIII

List of Figures IX

List of Abbreviations and Acronyms X

CHAPTER 1 : INTRODUCTION 1

1. Background 1
2. Objectives 5
3. Scope of the study 5
4. Significance of the study 6
5. Expected Outcome of the research 6
6. Organization of Thesis Chapters 7

CHAPTER 2 : LITERATURE REVIEW AND RELATED WORKS 8

2.1. Machine learning 8

2.2. Text Classification Using Machine learning 10

2.3. Text classification based on Named Entity Recognition 13

2.4. Related works 17

CHAPTER 3 : RESEARCH METHODOLOGY 22

3.1. Overview of research design 22

3.2. Data collection 23

3.3. Text classification modelling 24

3.4. POS (Part of Speech) 27

3.5. IOB Taging 27

3.6. Feature Extraction Methods 28

3.5. Machine Learning Classification Algorithm 29

3.6. Semantic similarity algorithms 34

3.7. Performance Evaluation Metrics 39

CHAPTER 4 : PROPOSED SYSTEM NEWS CLASSIFICATION 42

4.1. The proposed Cyber threat information classification Architecture 42

4.2. Proposed Text Pre-processing 43

4.3. Named Entity Recognition (NER) 45
4.4. Identify Cyber Attack features 47
4.5. Cyber threat News similarities 48
4.6. Proposed BiLSTM-CRF model Building 50
4.7. Models Evaluation and Testing 55
CHAPTER 5 : IMPLEMENTATION AND EXPERIMENTATION 56
5.1. Implementation Environment 56
5.2. Anaconda Distribution 56
5.3. Deployment Environment 59
5.4. Dataset description 59
5.6. Machine Learning Models Implementation 63
5.7. Model testing and evaluation 67
CHAPTER 6 : SIMULATION RESULTS AND DISCUSSION 68
6.1. Experiments evaluation 68
6.2. Model building Results 71
CHAPTER 7 : CONCLUSIONS, RECOMMENDATION, AND FUTURE WORKS 80
7.1 Conclusion 80
7.2 Recommendation 80
7.3. Future Work 81
REFERENCE 82
Appendixes 85
Appendix A: Text Pre-processing 85
Appendix B: Bi-LSTM-CRF model implementation code 87
Appendix C: LSA implementation code 89

List of Tables

Table 2-1: Summary of related works 20

Table 3-1: selected page information 23

Table 3-2: Text classification comparison 32

Table 5-1: Tools and Python Packages for Implementation 57

Table 5-2: Tagged cyber- attack features 59

Table 5-3: Similarity of two sentences 63

Table 6-1: Bi-LSTM-CRF model training and testing results 72

Table 6-2 : CRF model performance measurement 75

Table 6-3 : predicted and actual value of confusion matrix 75

Table 6-4 : True Positive values 76

Table 6-5 : calculated similarity of news article 77

List of Figures

Figure 2-1: machine learning working flow (Bonaccorso, 2017) 9

Figure 2-2: The framework of supervised text classification (Veldkamp, 2012) 11

Figure 2-3: Name entity recognition system pipeline (Nadeau & Sekine, 2007) 13

Figure 2-4 : Latent Semantic Analysis Sequence diagram (Tsai & Chan, 2007) 15

Figure 3-1: Overview of a research design diagram 22

Figure 3-2 : LSI Algorithm 37

Figure 3-3 : Confusion matrix metrics 40

Figure 4-1: Cyber threat Detection Architecture 42

Figure 4-2: Text pre-processing 43

Figure 4-3: Sample text tagged with named entities 46

Figure 4-4: Cyber-attack feature identification from news sources 48

Figure 4-5: Cyber threat news similarity measure 49

Figure 4-6: Cyber text news similarity calculation 50

Figure 4-7: Building model using Bi-LSTM-CRF 51

Figure 6-1 : Types of cyber-attack 69

Figure 6-2 : Cyber-attack incidents by year 69

Figure 6-3 : Number of words part of speech tagging 70

Figure 6-4 : tagged Cyber-attack features 70

Figure 6-5 : Number of words in a sentence 71

Figure 6-6 : Lose value of training and test CRF model 73

Figure 6-7 : Training and testing CRF accuracy 74

Figure 6-8 : Cyber threat news detection 76

Figure 6-9 : Comparison between machine learning models for text news classification 78

List of Abbreviations and Acronyms

CRF	Conditional Random Field
CVE	Common Vulnerabilities Exposure
CVSS	Common Vulnerability Scoring System
DDOS	Distributed Denial of Service
DTC	Decision Tree Classifier
IE	Information Extraction
LSA	Latent Semantic Analysis
LSI	Latent Semantic Index
LSTM	Long Short Time Memory
ML	Machine Learning
NB	Navies Bayes
NER	Named Entity Recognition
NLTK	Natural Language Tool Kit
NLP	Natural Language Processing
POS	Part of Speech Tagging
SVD	Singular value decomposition
SVM	Support Vector Machine
TDM	Term-document matrix

CHAPTER 1 : INTRODUCTION

1. Background

Cyber-security is the practice of defending computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks. Cyber is important because government, military, corporate, financial, and medical organizations collect, process, and store unprecedented amounts of data on computers and other device(Ye, Harish, & Farley, 2005).

A cyber security threat is a threat that is mounted against digital devices by means of cyberspace. Cyberspace, a virtual space that doesn't exist, has become the metaphor to help us understand digital weaponry that intends to harm. Cyber-attacks can cause electrical blackouts, failure of military equipment, and breaches of national security secrets. They can result in the theft of valuable, sensitive data like medical records. They can disrupt phone and computer networks or paralyze systems, making data unavailable. It's not an exaggeration to say that cyber threats may affect the functioning of life. Cyber security risks pervade every organization and aren't always under its direct control. Business leaders are forging ahead with their digital business initiatives, and those leaders are making technology-related risk choices every day.

Cyber security threats are acts performed by individuals with harmful intent, whose goal is to steal data, cause damage to or disrupt computing systems. Common categories of cyber threats include malware, social engineering, man in the middle (MitM) attacks, denial of service (DoS), and injection attacks. Cyber security threats can originate from a variety of sources, from hostile nation states and terrorist groups, to individual hackers, to trusted individuals like employees or contractors, who abuse their privileges to perform malicious acts.

One of the most significant components of our lives is the Internet. It gives us access to real-time information from anyplace there is a network. With the advent of online newspapers, the process of obtaining information about current or past happenings has grown and become easier. Many news provider companies compete with each other to provide a better service to attract customers which in the meantime, benefits the customers to get more quality and resourceful information (Carey & Manic, 2016).

The intensification in terms of the number and types of cyber-attacks in recent years is a result of an increased number of vulnerabilities, generated from accelerated digitization and digitalization processes (Bissell, Lasalle, & Cin, 2019). Research within the cyber land scape in 2019 reveals increase in security breaches, with effects ranging from financial loss to reputational damage (Ponemon, 2017). The cyber security industry is estimated to grow significantly by integrating innovative risk prevention and mitigation strategies.

Traditional cyber-attack prevention strategies involve identifying system or infrastructure vulnerabilities by using a list of Common Vulnerabilities Exposure (CVE) codes, with their associated Common Vulnerability Scoring System (CVSS) scores (Schiappa, Chantry, & Garibay, 2019). CVE codes contain an identification number, a description, and a public reference for vulnerabilities identified by competent entities. CVSS scores contain metrics related to access vectors, access complexity, and impact in terms of confidentiality, integrity, and availability, required privileges, user interaction, scope, exploitability, remediation level, report confidence, together with relevant information about an identified cybernetic vulnerability.

The early detection of vulnerabilities represents the process of collecting, cleaning, and analyzing data to identify emerging cybernetic threats. Research on this approach advocates for its efficiency, in both public and private contexts .This paper presents a model that aims to identify emerging cybernetic vulnerabilities in cyber security news articles, as part of a system for automatic detection of early cybernetic threats based on cyber-attack features.

Types of Cyber security Threats

There are some of the most common and prevalent cyber security threats that present-day organizations need to know about .The top cyber security threats are as follows

Malware Attacks:-Malware is an abbreviation of "malicious software", which includes viruses, worms, Trojans, spyware, and ransom ware, and is the most common type of cyber-attack. Malware infiltrates a system, usually via a link on an untrusted website or email or an unwanted software download. It deploys on the target system, collects sensitive data, manipulates and blocks access to network components, and may destroy data or shut down the system altogether.

Social Engineering Attacks:-Social engineering involves tricking users into providing an entry point for malware. The victim provides sensitive information or unwittingly installs malware on their device, because the attacker poses as a legitimate actor.

Man-in-the-Middle Attack: - involves intercepting the communication between two endpoints, such as a user and an application. The attacker can eavesdrop on the communication, steal sensitive data, and impersonate each party participating in the communication.

Denial-of-Service Attack: A Denial-of-Service (DoS) attack overloads the target system with a large volume of traffic, hindering the ability of the system to function normally. An attack involving multiple devices is known as a distributed denial-of-service (DDoS) attack.

Injection Attacks: - Injection attacks exploit a variety of vulnerabilities to directly insert malicious input into the code of a web application. Successful attacks may expose sensitive information, execute a DoS attack or compromise the entire system.

Malware, phishing, ransomware, man-in-the-middle attacks, and other ways can be used by cyber criminals to conduct a cyber-attack. By hacking into a vulnerable system, a cybercriminal can steal, alter, or destroy a specific target. Cyber threats can range in sophistication from the installation of harmful software such as malware or a ransomware assault (such as Wannacry) on a small firm to the attempted takedown of key infrastructure (Kharaz, Arshad, Mulliner, Robertson, & Kirda, 2016). A data breach, in which personal data or other sensitive information is exposed, is a common by product of a cyber-attack. Financial gain fuelled cyber-attacks and breaches, which targeted banks and credit cards.

1.1. Motivation

The present-day world has become all dependent on cyberspace for every aspect of daily living. The use of cyberspace is rising with each passing day. The world is spending more time on the Internet than ever before. As a result, the risks of cyber threats and cybercrimes are increasing. The term 'cyber threat' is referred to as the illegal activity performed using the Internet. Cybercriminals are changing their techniques with time to pass through the wall of protection. Conventional techniques are not capable of detecting sophisticated attacks. There are large volumes of data from the online news sources that are freely available which might contain valuable information. Data such as cyber-attacks news keep growing bigger and can

be analysed to gather informative insights of current situation. However, news is reported in many styles, added with the emerging of new cyber-attack and the ambiguous terms used have made the detection of the related news become more difficult. Thus, to handle these situations, machine learning techniques are developing to detect cyber threats based on cyber-attack features. The number of cybercrimes and cyber threats is becoming increasing. Different researches have been done to tackle this problem. Thus, this research is motivated to contribute solutions to this frustrating scenario to information use through experimenting and predicting situations of cyber threats from online news.

1.2. Statement of the Problem

Cyber-attack is one of the issues that become more popular report in the online newspaper. Cyber Incidents are occurred in the global emergence of ransomware such as "Wannacry" and "Petya" that have caused havoc and affect government sectors, people, Financial Transactions, Lose a lot of resources such as money and valuable data or personal information. They can result in the theft of valuable, sensitive data like medical records. They can disrupt phone and computer networks or paralyze systems, making data unavailable. It's not an exaggeration to say that cyber threats may affect the functioning of life as we know it. This paper will investigate machine learning techniques which are good for the cyber-threat detection from online news. Machine learning is a partial but significant departure from traditional well-known security solutions, including user authentication and access control, firewalls and cryptography systems, which may not be effective in meeting today's cyber business needs. Growing number of cyber security incidents in various formats are emerging over time, traditional solutions have proven ineffective in managing these cyber-hazards.

The full picture of the current cyber threat incidents and activities from internet news reports is still difficult for many people to realize. The literature review and analysis of the data collected showed that various news sources used different phrases to report the same news. (Arulanandam, Savarimuthu, & Purvis, 2014), (Brown & Liu, 2016), (Abdullah, Zainal, Maarof, & Kassim, 2018). multiple news sources reported the same news will occur Ambiguity problem when retrieving the information. We propose a scheme on identifying cyber threats from online news. The scheme starts with identifying the cyber-attack features

which will be used to classify the Cyber threats using hybrid of Bi directional LSTM-CRF model and ambiguous information is identified by Latent Semantic Analysis (LSA).

1.3. Research Questions

This study addresses the following research questions:

a) How cyber threats identification from online news improve awareness of security threat?

b) Does a hybrid machine learning algorithm accordingly identify cyber threat information from online news?

c) How can the features to identify cyber threats information in online news?

d) Collect large amount of data set and remove redundancy news from collected sentences

2. Objectives

2.1. General Objective

The general objective of this study is Detecting Cyber Threat information from online sources using Machine Learning algorithm.

2.2. Specific Objectives

The specific objectives this study attempts to achieve are the following

1) To build dataset from online news sources
2) To Identify similar/ ambiguous news article
3) To Identify Cyber-attack features from build dataset.
4) To Detect cyber threat information based on cyber-attack features
5) To evaluate the performance of the system for detecting cyber threats information.

3. Scope of the study

The scope of this study is to provide a system which focus on identifying the cyber-attack features and show the process to classify and detect cyber threats from online news article. In addition redundant information are identified and removed to get valuable information. The cyber-attack features are one of the important characteristics to distinguish the news which

could increase the accuracy of detection of cyber threat news that are related to cyber-attack. The cyber-attack features are included only four features i.e. threat type, threat actor, affected country and organization. This study does not consider name abbreviations instead full name to label the data. Additionally, cyber-attack features is manually identify from the collected dataset.

4. Significance of the study

The proposed research aims at developing a Hybrid model for classifying cyber threat information from online news based on cyber-attack features. The findings of this study will benefit cyber security researchers or experts in such a way that experts and researchers will have machine learning to classify cyber threat information. Cyber security is important because government, military, corporate, financial, and medical organizations collect, process, and store unprecedented amounts of data on computers and other devices. This research implemented hybrid approach combining a bidirectional LSTM model and a CRF model. Identifying cyber-attack features with dictionary that able to increase the accuracy of detection of the cyber-attack news. Since the study is master thesis it has also learning out comes for the researcher. It helps the researcher to examine problems and solving it in scientific manner. Generally the system is detected cyber threat news based on cyber-attack features such us the types of attacks, hackers who influence organization in a country. As a result, the system help the people show the actual occurrences of cyber-attack activities.

5. Expected Outcome of the research

The expected outcome of the research is to develop a new system that able to detect cyber-attack news from online news articles. The output will eventually show the whole Picture of cyber-attack activities that occur in our cyber space. Thus, it can help cyber security organization to generate report regarding cyber-attack activities or help in providing insight for any decision-making process. This research is trying to increase the awareness of public by showing the numbers and figures of the current cyber threats occurrences so that people will be more alert and able to avoid from being a victim.

6. Organization of Thesis Chapters

This research paper is organized into seven chapters in the following ways:

Chapter One: discussed in the above

Chapter Two: presents the literature review

Chapter Three: discusses the research methodology for this research work, methods used to build the dataset, pre-processing steps (data cleaning and transformation), and machine learning approach, finally, it presents the evaluation methods.

Chapter Four: discusses the proposed framework

Chapter Five: Discusses the Implementation and Experimentation of the proposed solution,

Chapter Six: Discusses the Results of the proposed machine learning approaches and also discusses the major results obtained by comparing all models based on the performance metrics.

Chapter Seven: Conclusion, Recommendation, and Future Work.

CHAPTER 2 : LITERATURE REVIEW AND RELATED WORKS

In this chapter, the relevant literature related to the research topic is reviewed to clearly state the gaps and further explore the research problem. This is done through providing a technical review on cyber threat occurrences and existing method to detect cyber threat news from online news articles. One of the works published on the subject matter of Detecting cyber threat information is a paper titled as extracting crime information of public yet 'hidden. conducted by (Arulanandam et al.2014)

2.1. Machine learning

Machine Learning is the study of making machines more human-like in their behaviour and decisions by giving them the ability to learn and develop their own programs. This is done with minimum human intervention i.e., no explicit programming. The learning process is automated and improved based on the experiences of the machines throughout the process. Good quality data is fed to the machines, and different algorithms are used to build ML models to train the machines on this data. The choice of algorithm depends on the type of data at hand, and the type of activity that needs to be automated (Bonaccorso, 2017).

Machine Learning today has all the attention it needs. Machine Learning can automate many tasks, especially the ones that only humans can perform with their innate intelligence. Replicating this intelligence to machines can be achieved only with the help of machine learning. With the help of Machine Learning, businesses can automate routine tasks. It also helps in automating and quickly creates models for data analysis. Various industries depend on vast quantities of data to optimize their operations and make intelligent decisions. Machine Learning helps in creating models that can process and analyse large amounts of complex data to deliver accurate results. These models are precise and scalable and function with less turnaround time. By building such precise Machine Learning models, businesses can leverage profitable opportunities and avoid unknown risks.

Machine learning model suffering from over fitting. In order to avoid this type of problem, it is necessary to apply either regularization or dimensionality reduction techniques (Feature Extraction) (Bonaccorso, 2017). In Machine Learning, the dimensionality of a dataset is equal to the number of variables used to represent it. Using Regularization could certainly help reduce the risk of over fitting, but using instead Feature Extraction techniques can also lead to other types of advantages such as:

i. Accuracy improvements.
ii. Over fitting risk reduction.
iii. Speed up in training.
iv. Improved Data Visualization.

Feature Extraction aims to reduce the number of features in a dataset by creating new features from the existing ones (and then discarding the original features). The feature extraction working flow is as shown below.

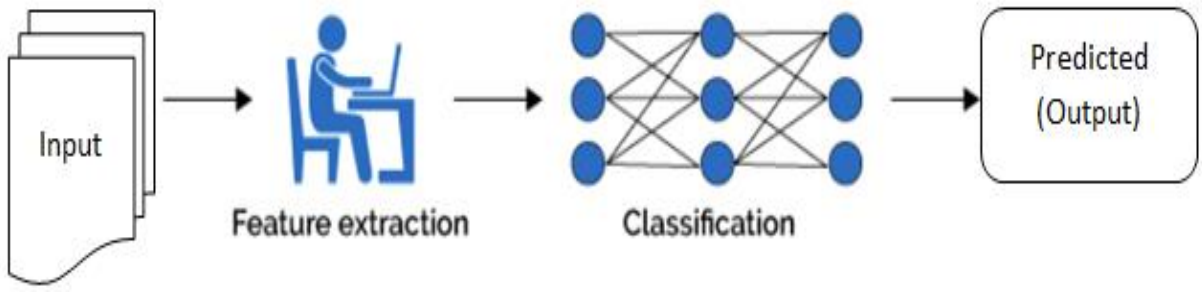


Figure 2-1: machine learning working flow (Bonaccorso, 2017)

These new reduced set of features should then be able to summarize most of the information contained in the original set of features. In this way, a summarized version of the original features can be created from a combination of the original set. Machine learning terminologies are defined below.

Model:- Also known as "hypothesis", a machine learning model is the mathematical representation of a real-world process. A machine learning algorithm along with the training data builds a machine learning model.

Feature: - A feature is a measurable property or parameter of the data-set.

Feature Vector: It is a set of multiple numeric features. We use it as an input to the machine learning model for training and prediction purposes.

Training: - An algorithm takes a set of data known as "training data" as input. The learning algorithm finds patterns in the input data and trains the model for expected results (target). The output of the training process is the machine learning model.

Prediction: - Once the machine learning model is ready, it can be fed with input data to provide a predicted output.

Target (Label): - The value that the machine learning model has to predict is called the target or label.

2.2. Text Classification Using Machine learning

News Text classification is a machine learning technique that automatically assigns tags or categories to text. Using natural language processing (NLP), text classifiers can analyse and sort text by sentiment, topic, and customer intent – faster and more accurately than humans. With data pouring in from various channels, including emails, chats, web pages, social media, online reviews, support tickets, survey responses; you name it, it's hard for humans to keep up. All channels of communication are incredibly rich sources of information for businesses, but it's challenging for them to process all the data they receive, extract valuable insights that could help them make informed decisions. Because of this, companies are turning to automated solutions like text classification. Powered by machine learning, text classification enables you to classify text in a reliable, scalable, accurate, and cost-effective way.

To begin training a classifier with machine learning, you need to transform text into something a machine can understand. This is often carried out using a bag of words, where a vector represents the frequency of a word within a predefined list of words.

Once data is vectorized, the text classifier model is fed training data that consists of feature vectors for each text sample and tag. With enough training samples, the model will be able to make accurate predictions.

Supervised Text Classification

Supervised machine learning algorithms are trained with labelled datasets that are created by humans. Raw datasets are transformed into numerical vectors that are the input variables (X)

which are easily understood by machine learning models and the output variables (Y). The supervised algorithm learns the mapping function from the input to the output. When you have new data, this technique should be able to predict the variables for that data. Supervised text classification is a commonly used approach for textual categorization, which generally involves two phases, a training phase and a prediction phase (Veldkamp, 2012).The objective of the training phase is to learn the relationship between the keywords and the class labels. The prediction phase plays an important role in checking how well the trained classifier model performs on a new dataset. The test set should consist of data that were not used during training. In the testing procedure, the keywords extracted from the training are scanned in each new input. Thus, the words that were systematically recognized are fed into the trained classifier model, which predicts the most likely label for each new self-narrative. To ensure proper generalization capabilities for the text classification models, a cross validation procedure is generally applied

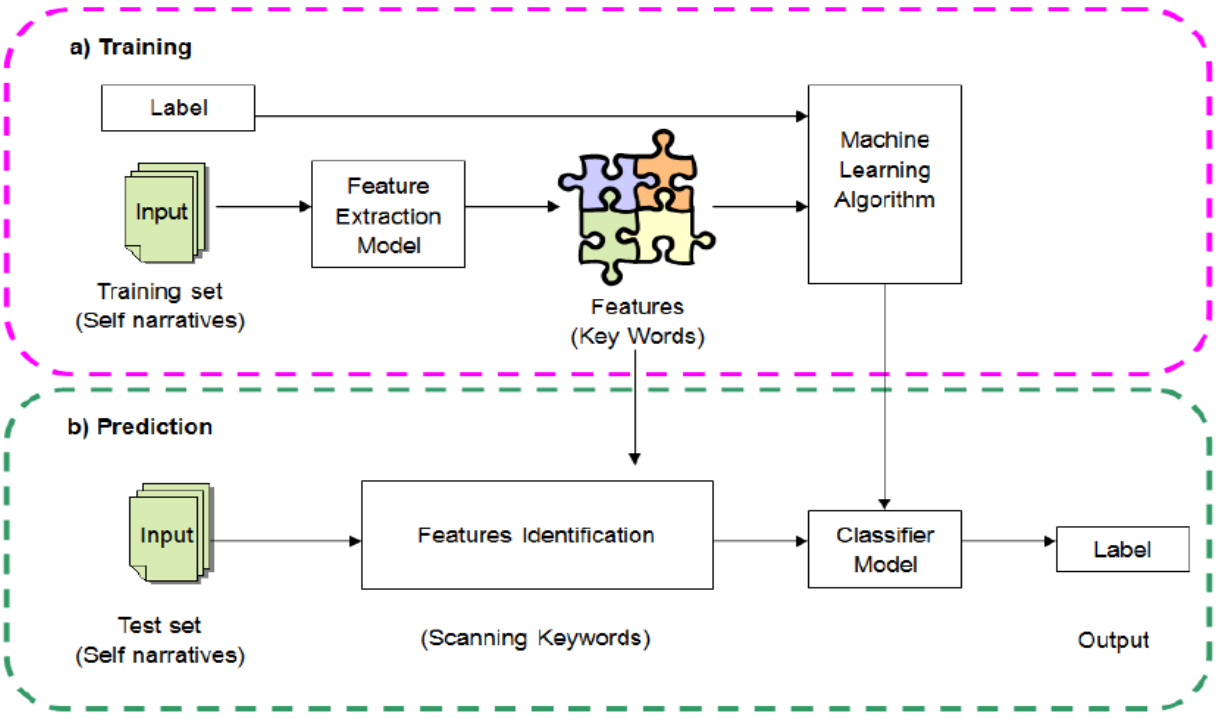


Figure 2-2: The framework of supervised text classification (Veldkamp, 2012)

Text classification using supervised machine learning has various applications, including ticket routing. In this example, incoming messages would be automatically tagged by topic, language, sentiment, intent, and more, and routed to the right customer support team based on their expertise.

Another great example is email filtering. Emails are sorted based on content, which is the result of algorithms learning to detect content that is considered spam, social, or inbox-worthy. Two of the most common supervised algorithms include Naive Bayes and support vector machines (SVM).

Unsupervised Machine Learning

In contrast, unsupervised machine learning algorithms learn independently from raw datasets. In other words, it learns from the distributions of words and syntactic patterns in vast amounts of data, which are transformed into a huge vector.

In a nutshell, unsupervised machine learning is not given any labelled outputs or a “correct” outcome. It automatically understands the underlying structure of data and groups words with similar attributes. One of the most popular techniques of unsupervised learning is clustering, which segments topics in text.

Semi-Supervised Machine Learning

Finally, semi-supervised machine learning, otherwise known as hybrid machine learning, is a combination of unsupervised and unsupervised machine learning methods trained with labelled (typically a small amount) and unlabelled data (typically a large amount).

Semi-supervised machine learning is helpful in scenarios where businesses have huge amounts of data to label. For example, semi-supervised machine learning can be used for webpage classification to automatically categorize types of webpages based on labelled input and looking at similarities between webpages, thus identifying and labelling 'unstructured’ webpages.

2.3. Text classification based on Named Entity Recognition

Named entity recognition (NER) is a natural language processing tool for information extraction from unstructured text data such as e-mails, newspapers, blogs, etc. The goal of named entity recognition (NER) systems is to identify names of people, locations, organizations, and other entities of interest in text documents (Nadeau & Sekine, 2007). An NER system should be able to identify NEs in the sentence. NER plays a key role in larger information extraction (IE) from natural language texts. NER finds practical use in a wide range of applications such as machine translation, speech recognition, chat bots, and search engines.Traditional approaches to NER include matching a list of names with documents to identify NEs and other simple rule-based approaches. A typical NER system pipeline includes pre-processing the data such as tokenization, sentence splitting, feature extraction, applying ML models on the data for tagging, and then post-processing to remove some tagging inconsistencies.

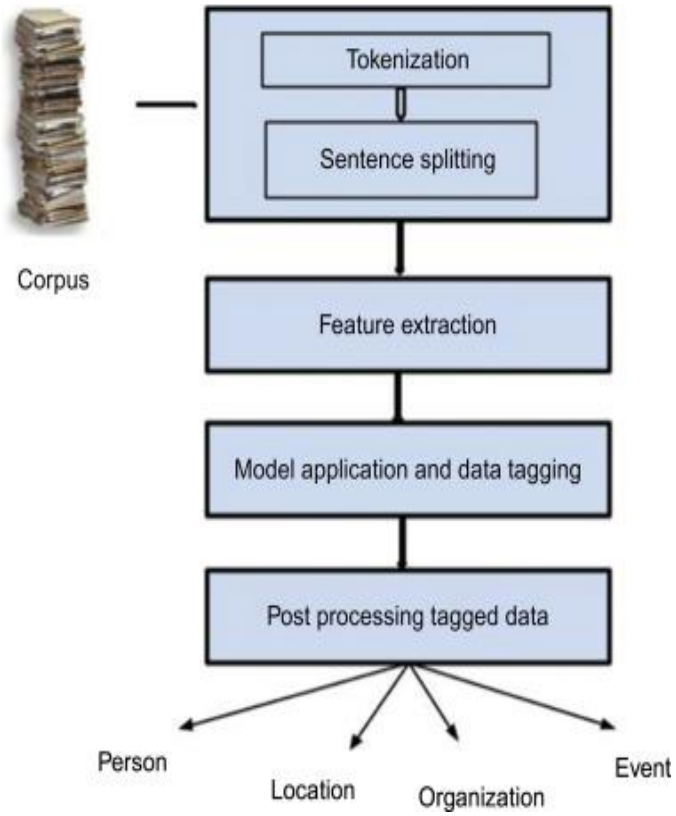


Figure 2-3: Name entity recognition system pipeline (Nadeau & Sekine, 2007)

Statistical NLP methods

NER can be treated as a standard classification problem, for example, identifying if a token is an entity or not. Statistical machine learning algorithms can be applied to solve this problem. Supervised classifiers yielded better NER systems compared to rule-based ones, as they are easier to maintain and adapt to different domains. Hidden Markov models (HMMs) and support vector machine (SVM)-based models were used in NER tasks with varying degree of success.

Maximum entropy (ME) models were highly successful. These models use posterior probability of a tag given a word within a limited window wn+in−i=wn−i…. w_{n+i} around *n*th word w_n with a window size of $2i + 1$. The models comprised of set of binary features such as lexical, word based, transition, prior, compound, and dictionary features.

Semantic Similarity of Documents Using Latent Semantic Analysis

Latent Semantic Analysis (LSA) a well-known technique for information retrieval and document classification(Tsai & Chan, 2007). LSA solves two fundamental problems in natural language processing synonymy and polysemy. In synonymy, different words may have the same meaning. In polysemy, the same word can have multiple meanings.

LSA uses a term-document matrix (TDM) which describes patterns of term (word) distribution across a set of documents. The downsizing of the matrix is achieved through the use of singular value decomposition (SVD), where the set of all the terms is then represented by a vector space of lower dimensionality than the total number of terms in the vocabulary. The consequence of the rank lowering is that some dimensions get “merged”.

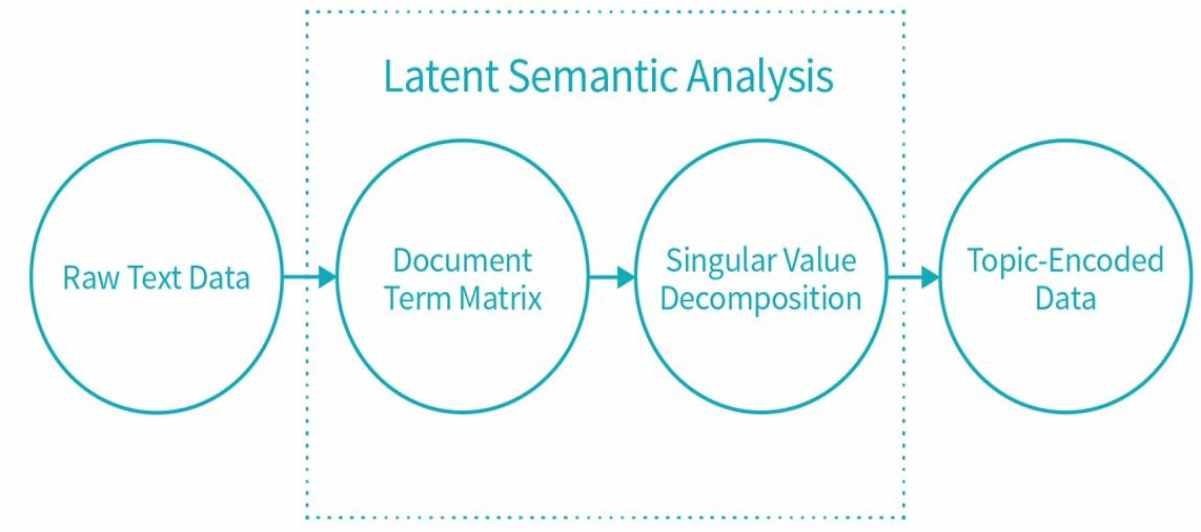
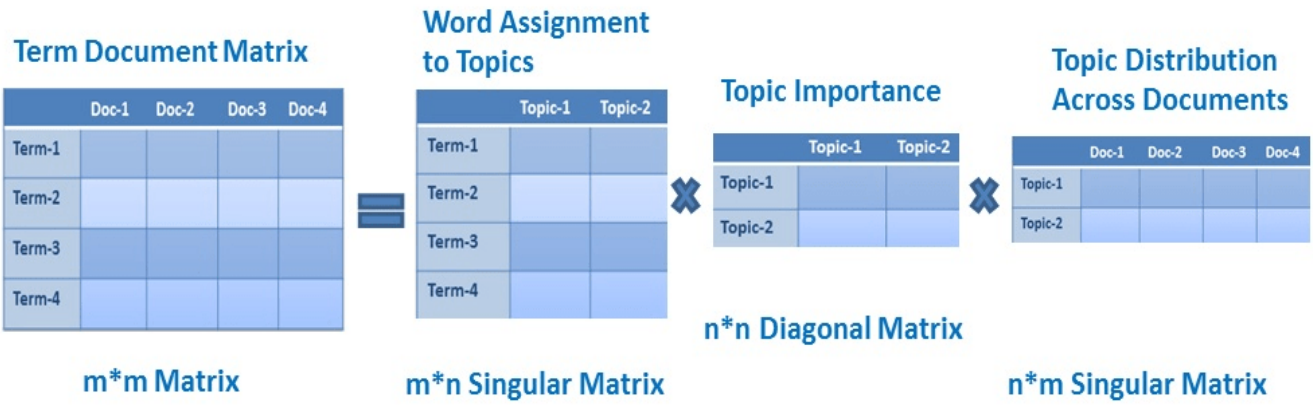


Figure 2-4 : Latent Semantic Analysis Sequence diagram (Tsai & Chan, 2007)

Singular value Decomposition (SVD) is similar to a principal component analysis that used to reduce the dimension of the original data (Tsai & Chan, 2007). Parsing the document collection generates a term-document frequency matrix. Each entry of the matrix represents the number of times that a term appears in a document. For a collection of several thousand documents, the term-document frequency matrix can contain hundreds of thousands of words. It requires too much computing time and space to analyse this matrix effectively. Also, dealing with high dimensional data is inherently difficult for modelling.

To improve the performance, singular value decomposition (SVD) can be implemented to reduce the dimensions of the term-document frequency matrix. SVD transforms the matrix into a lower dimensional, more compact, and informative form.

LSA (Latent Semantic Analysis) also known as LSI (Latent Semantic Index) LSA uses bag of word (BoW) model, which results in a term-document matrix (occurrence of terms in a document). Rows represent terms and columns represent documents. LSA learns latent topics by performing matrix decomposition on the document-term matrix using Singular value decomposition. LSA is typically used as a dimension reduction or noise reducing technique.



SVD is a matrix factorization method that represents a matrix in the product of two matrices. It offers various useful applications in signal processing, psychology, sociology, climate, and atmospheric science, statistics and astronomy.

$$M=U\Sigma V^*$$

M is an m×m matrix

U is a m×n left singular matrix

Σ is a n×n diagonal matrix with non-negative real numbers.

V is a m×n right singular matrix

V* is n×m matrix, which is the transpose of the V.

Identity matrix: It is a square matrix in which all the elements of the principal diagonal are ones, and all other elements are zeros.

Diagonal Matrix: It is a matrix in which the entries other than the main diagonal are all zero.

Singular Matrix: A matrix is singular if its determinant is 0 or a square matrix that does not have a matrix inverse.

2.4. Related works

In recent years many researches are done for detect and classify cyber security attacks by extracting addresses and crime information from online newspaper. For this research, identify cyber-attack features and classify cyber-attack news using machine learning algorithms.

Arulanandam et al.(2014) conducted a study that aimed at extracting crime information of public yet 'hidden' from online newspaper articles to make available to the general public.This paper focused on one type of crime, the theft crime and demonstrated how theft-related information can be extracted from newspaper articles from three different countries. The system developed employs Named Entity Recognition (NER) algorithms to identify locations in sentences as well as Conditional Random Field (CRF), a machine learning approach to classify whether a sentence in an article is a crime location sentence or not. The paper compared the accuracy of identifying crime location sentences using the developed model in newspapers from Australia and India on top of New Zealand and the authors achieved 84% accuracy.

According to (Fang, Gao, Liu, & Huang, 2020) In the context of increasing cyber threats and attacks, monitoring and analysing network security incidents in a timely and effective way is the key to ensuring network infrastructure security. As one of the world's most popular social media sites, users post all kinds of messages on Twitter, from daily life to global news and political strategy. It can aggregate a large number of network security-related events promptly and provide a source of information flow about cyber threats. In this paper, for detecting cyber threat events on Twitter, the research presents multi-task learning approach based on the natural language processing technology and machine learning algorithm of the Iterated Dilated Convolutional Neural Network (IDCNN) and Bidirectional Long Short-Term Memory (BiLSTM) to establish a highly accurate network model. Furthermore, they collect a network threat-related Twitter database from the public datasets to verify our model's performance. The results show that the proposed model works well to detect cyber threat events from tweets and significantly outperform several baselines. This paper focus on the two main tasks: detection of cyber threat events on tweets and the NER task. The results show that multitask model can achieve the effectiveness of two subtasks executed separately.

That simplifies the complexity of using deep neural networks dramatically. At the same time, the proposed method's performance by comparing multiple baseline methods and results show that proposed method has more outstanding performance. The use of Twitter is very active, and it is also essential to promptly detect new cyber security incidents.

According to (Arpitha, Sharan, Brunda, Indrakumar, & Ramesh, 2021) Cyber-crime is proliferating everywhere exploiting every kind of vulnerability to the computing environment. Ethical Hackers pay more attention towards assessing vulnerabilities and recommending mitigation methodologies. The development of effective techniques has been an urgent demand in the field of the cyber security community. Most techniques used in today's IDS are not able to deal with the dynamic and complex nature of cyber-attacks on computer networks. Machine learning for cyber security has become an issue of great importance recently due to the effectiveness of machine learning in cyber security issues. Machine learning techniques have been applied for major challenges in cyber security issues like intrusion detection, malware classification and detection, spam detection and phishing detection. Although machine learning cannot automate a complete cyber security system, it helps to identify cyber security threats more efficiently than other software-oriented methodologies, and thus reduces the burden on security analysts. Hence, efficient adaptive methods like various techniques of machine learning can result in higher detection rates, lower false alarm rates and reasonable computation and communication costs. The main goal is that the task of finding attacks is fundamentally different from these other applications, making it significantly harder for the intrusion detection community to employ machine learning effectively.

According to (Park & Kwon, 2022) Online social media such as Twitter has been used as an important source for predicting, detecting, or analysing critical social phenomena such as elections, disease outbreaks, and cyber-attacks. In this study, first, community detection of users related to cyber-attacks on Twitter to identify the most relevant group to the cyber-attacks. Second, to effectively identify the tweets related to cyber-attacks, the system conducts a textual similarity analysis between the tweet and the cyber-attack relevant keywords, which overcomes the limitation of lexical analysis of tweets such as keyword-based filtering and frequency of keywords. Finally, this paper proposes a novel cyber-attack

detection model by integrating both text- and graph-based models. The methodology has a distinguishing feature from the existing studies in that incorporate the semantics in Tweets to evaluate the relevance with cyber-attacks and employ community detection to identify the most relevant group to the cyber-attacks. Through extensive experiments shows the effectiveness of the proposed model. First, the text analysis in the proposed model outperforms detection accuracy of the keyword frequency-based analysis by up to 29.46%. Second, the community detection improves the detection accuracy by 28.89∼35.56% compared to the baseline criteria to select relevant users to the cyber-attacks. Through two experiments to measure the relevancy of detected communities to the cyber-attack, the results consistently show that the highest relevant community by community detection shows the highest relevancy.

This paper conducted (Iorga et al., 2020) that the drawbacks of traditional methods of cybernetic vulnerability detection relate to the required time to identify new threats, to register them in the Common Vulnerabilities and Exposures (CVE) records, and to score them with the Common Vulnerabilities Scoring System (CVSS). These problems can be mitigated by early vulnerability detection systems relying on social media and open-source data. This paper presents a model that aims to identify emerging cybernetic vulnerabilities in cyber security news articles, as part of a system for automatic detection of early cybernetic threats using Open Source Intelligence (OSINT). Three machine learning models were trained on a novel dataset of 1000 labeled news articles to create a strong baseline for classifying cyber security articles as relevant (i.e., introducing new security threats), or irrelevant: Support Vector Machines, a Multinomial Naïve Bayes classifier, and a finetuned BERT model. The BERT model obtained the best performance with a mean accuracy of 88.45% on the test dataset. Our experiments support the conclusion that Natural Language Processing (NLP) models are an appropriate choice for early vulnerability detection systems in order to extract relevant information from cyber security news articles.

According to (Rollo, Po, & Bonisoli, 2022) , Event Extraction is a complex and interesting topic in Information Extraction that includes methods for the identification of event's type, participants, location, and date from free text or web data. The result of event extraction systems can be used in several fields, such as online monitoring systems or decision support

tools. This paper introduces a framework that combines several techniques (lexical, semantic, machine learning, neural networks) to extract events from Italian news articles for crime analysis purposes. The study concentrates to represent the extracted events in a Knowledge Graph.

Table 2-1: Summary of related works

S.N	Author	Title	Method	Key Finding	Limitation
1	Rexy Arulanandam ,Bastin Tony, Roy Savarimuthu , Maryam A. Purvis, 2014	Extracting Crime Information from Online Newspaper Articles	Identify and classify crime location from online newspaper articles using named entity recognition (NER) algorithm and Conditional Random Field	extract hidden crime information from online newspaper articles and make this information available to the public	Do not eliminate duplicate articles reporting the same crime with in a newspaper
2	Fang, Gao, Liu, & Huang, 2020	Detecting Cyber Threat Event from Twitter Using IDCNN and BiLSTM	Detect cyber threat events using multi-task learning approach based on the natural language processing technology and Iterated Dilated Convolutional Neural Network (IDCNN) and Bidirectional Long Short-Term Memory (BiLSTM)	The proposed model works well to detect cyber threat events from tweets and significantly outperform several baselines.	Event detection based on Twitter is widespread and redundant tweets are occurred.
3	Arpitha, Sharan, Brunda, Indrakumar, & Ramesh, 2021	Cyber Attack Detection and notifying system using ML Techniques	designs and develops machine learning adaptive methods to detect and notifying cyber attacks	helps to identify cyber security threats more efficiently than other software-oriented methodologies,	Do not identify the types of cyber-attacks, actors, affected locations
4	Park & Kwon,	Cyber-attack	Design a novel cyber-attack detection model by	Improve cyber-attack detection	Do not identify the types of

	2022	detection model using community detection and text analysis on social media	integrating both text- and graph-based models.	from social media	cyber-attacks, actors, affected locations and affected organization
5	Iorga et al., 2020	Extracting Addresses from News Reports Using Conditional Random Fields	identify emerging cybernetic vulnerabilities in cyber security news articles using Support Vector Machines, a Multinomial Naïve Bayes classifier, and a finetuned BERT model	improve to identify cybernetic vulnerabilities	Use low dataset still need to improve better identification models
6	Rollo, Po, & Bonisoli, 2022	Online News Event Extraction for Crime Analysis	Design a framework that combines lexical, semantic, machine learning, and neural networks to extract events from Italian news articles.	Improve online event extraction	Difficult to categorize crime events.

Research gap

Generally the research to be done here is different from what is already presented in the related works, in that they didn't use large amount of data and duplicate articles are not consider with the same crime location. In addition, the studies have focused on machine learning approach to extract crime locations and cyber-attack events. Another observation in most of the literatures reviewed is news reports have no special formatting, and creating data sets requires manual labelling. This work implemented efficient methods to detect cyber threat news based on cyber-attack features and recognize cyber-attack name entities and keywords using name entity recognition and use large amount of data to increase the accuracy of the model. The research also identified similar news article with contextual meaning in order to avoid ambiguous issues.

The issue discussed in this chapter are different works related to our topic and their objective, Methodology and core finding was discussed and finally it was mentioned what makes this work different from the researches done before.

CHAPTER 3 : RESEARCH METHODOLOGY

3.1. Overview of research design

In this research, machine learning tools are used to predict cyber threat news and information. The reason why we choose machine learning tools is that, they can analysis data from different sources like image, video, and text recognition, as well as serving as the power behind recommendation engines.

To conduct identification of cyber threat information we followed different steps. Those steps are the final implementation of this study.

1. The earliest step is searching different sources to identify the research area.
2. Then based on the selected research area, the next step will be searching and analysing different papers and books to identify the research gap that is mentioned as the title of the study.
3. After the research gap is identified different research questions are formulated which are answered under this study.
4. The next step is collecting and organizing data from different sources which are the backbone for developing this system.
5. Then the process of implementing the design is started so that it is possible to solve the research gap identified.
6. After the implementation step is finished different evaluation techniques are carried out so as to check and evaluate our system performance.
7. Finally, the result of this whole step is written and explained.

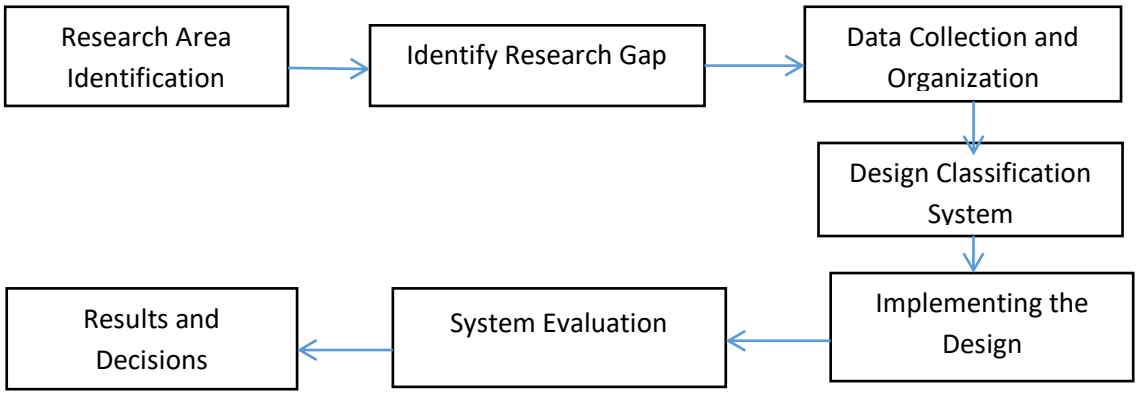


Figure 3-1: Overview of a research design diagram

3.2. Data collection

To obtain the data, we identify several websites that provide a list of cyber threat news. Data is collected from the news article related to the cyber-attacks that include incidents or attacks that had happened, the threat actors, the method used and few others. Data can be obtained from the news websites such as Recorded Future.com, Fire Eye, Security Week, Micro Trend and more. Some cyber threat news is collected from **Ethiopian Monitor** websites that a series of cyber-attack attempts were carried out targeting the dam by Egypt-based hackers named Cyber Horus Group, Anubis and Government ministries, regional bureaus, academic institutions and media house. The websites mentioned are known for their reliability and they were recommended by security experts who used the reporting services provided by these websites. For this research, 2019 cyber related news articles are collected in form of unstructured text or 24,709 words

Table 3-1: selected page information

S.No	**Name of websites/pages**	**Number of news sentences we collected**
1	Security week	852
2	Future.com	354
3	Fire Eye	257
4	Micro Trend	256
5	Cyware.com	246
6	Ethiopian monitor pages	54
Total		**2019**

After the data collected, a preparation process follows, which are collecting, cleaning, filtering, and consolidating data into one file or data table. This process used preparation tools such as MS-excel.

3.3. Text classification modelling

3.3.1. Text Pre-processing

Pre-processing is the main part of building a prediction model in our study. Pre-processing is a means of converting the noisy and huge data into relevant and clean data. As the collected data contains missing values and categorical data, it has to handle missing values and categorical data to prepare the dataset in a suitable format for the models to use it. Pre-processing is a very essential part of developing a machine-learning model since the inconsistent data affect the accuracy of the model. The following methods used in the prepossessing processes:

1. Removing (cleaning) irrelevant character, punctuations, symbol, blank value and whitespace, URL, and Emojis.
2. Removing elongation from the post and comment.
3. Performing text normalization.
4. Performing Tokenization to get the individual word or tokens in text.

Most text and document data sets contain many unnecessary words such as stop words, misspelling, slang, etc. In many algorithms, especially statistical and probabilistic learning algorithms, noise and unnecessary features can have adverse effects on system performance. In this section, we briefly explain some techniques and methods for text cleaning and pre-processing text data sets.

a) Tokenization

Tokenization is a pre-processing method which breaks a stream of text into words, phrases, symbols, or other meaningful elements called tokens. The main goal of this step is the investigation of the words in a sentence. Both text classification and text mining require a parser which processes the tokenization of the documents, for example:

sentence :

After sleeping for four hours, he decided to sleep for another four.

In this case, the tokens are as follows:

{ *"After" "sleeping" "for" "four" "hours" "he" "decided" "to" "sleep" "for" "another" "four"* }.

b) Stop Words

Text and document classification includes many words which do not contain important significance to be used in classification algorithms, such as {*"a", "about", "above", "across", "after", "afterwards", "again",. . .*}. The most common technique to deal with these words is to remove them from the texts and documents.

c) Capitalization

Text and document data points have a diversity of capitalization to form a sentence. Since documents consist of many sentences, diverse capitalization can be hugely problematic when classifying large documents. The most common approach for dealing with inconsistent capitalization is to reduce every letter to lower case. This technique projects all words in text and document into the same feature space, but it causes a significant problem for the interpretation of some words (e.g., "*US*" (United States of America) to "*us*" (pronoun)). Slang and abbreviation converters can help account for these exceptions.

d) Slang and Abbreviation

Slang and abbreviation are other forms of text anomalies that are handled in the pre-processing step. An abbreviation is shortened forms of a word or phrase which contain mostly first letters form the words, such as SVM which stands for Support Vector Machine.

Slang is a subset of the language used in informal talk or text that has different meanings such as "lost the plot", which essentially means that they've gone mad. A common method for dealing with these words is converting them into formal language.

e) Noise Removal

Most of the text and document data sets contain many unnecessary characters such as punctuation and special characters. Critical punctuation and special characters are important for human understanding of documents, but it can be detrimental for classification algorithms.

f) Spelling Correction

Spelling correction is an optional pre-processing step. Typos (short for typographical errors) are commonly present in texts and documents, especially in social media text data sets (e.g., Twitter). Many algorithms, techniques, and methods have addressed this problem in NLP. Many techniques and methods are available for researchers including hashing-based and context-sensitive spelling correction techniques, as well as spelling correction using Trie and Damerau–Levenshtein distance bigram.

g) Stemming

In NLP, one word could appear in different forms (i.e., singular and plural noun form) while the semantic meaning of each form is the same. One method for consolidating different forms of a word into the same feature space is stemming. Text stemming modifies words to obtain variant word forms using different linguistic processes such as affixation (addition of affixes). For example, the stem of the word "studying" is "study".

The benefits of using the stemming algorithm in an NLP can be summarised as follows:

1. It reduces the number of words that serve as an input to the Machine Learning/Deep Learning model.
2. It minimizes the confusion around words that have similar meanings.
3. It lowers the complexity of the input space.
4. When creating applications that search a specific text in a document, using stemming for indexing assists in retrieving relevant documents.
5. It assists in eliminating the out-of-vocabulary (OOV) problem. For example, if the vocabulary does not contain the word 'oranges', one can use the stem word 'orange' as a proxy.

6. It enhances the accuracy of the ML/DL model as the model does not have to deal with inflected word forms

h) Lemmatization

Lemmatization is a NLP process that replaces the suffix of a word with a different one or removes the suffix of a word completely to get the basic word form (lemma).

3.4. POS (Part of Speech)

Tagging Part of Speech (POS) tagging is an essential part of NER. POS tagging will assign each word in the corpus to its POS tag based on the definition and context. POS tagging aims to categorize word classes into nouns, adjectives, verbs, etc. In this work, the POS tagging task is performed using the tool provided by NLTK Services, which provides the POS tagging function. Example Sentence: "John love to play score"

'John', "NNP", 'Love', "VBZ", 'to', "To", 'Play', "VB", 'Soccer, "NN"

3.5. IOB Taging

IOB labelling (short for **Inside, Outside, Beginning labelling**) is a popular technique used in Named Entity Recognition (NER) to annotate and categorize text data.

IOB labelling involves labelling each word or token in a text sequence with a specific tag that indicates whether it is part of a named entity or not. In IOB labelling, each token in a sequence is assigned one of three tags:

- **B (Beginning):** the token marks the beginning of a named entity.
- **I (Inside):** the token is part of a named entity, but it is not the first token.
- **O (Outside):** the token is not part of a named entity.

Essential info about entities:

- geo = Geographical Entity
- org = Organization
- per = Person
- gpe = Geopolitical Entity
- tim = Time indicator
- art = Artifact

- eve = Event
- nat = Natural Phenomenon

3.6. Feature Extraction Methods

Feature extraction methods based on state-of-the-art text mining; techniques applied for reducing redundant features and dimensionality. The methods involved in selecting a subset of relevant features that would help in identifying hate, offensive, and neither from the dataset and can be used in the modelling of the detection problems. The N-gram, TF-IDF, and word2vec feature extraction used in this study because they are better and popular feature extraction methods used cyber threat information detection studies and text classification problems. The following features extraction methods are used for modelling cyber threat news detection.

N-Gram: N-grams are one of the most used techniques in cyber threat information detection. N-gram is a word prediction model using probabilistic methods to predict the next word after observing N-1 words. The most common N-grams approach consists in combining sequential words into lists with size N, where N is the number of words used in the probability sequences. E.g., if N=1, N=2, and N=3 referred them as unigram, bigram, and trigram, respectively. This study uses a word N-gram method to create N-gram of post and comment features. These three n-grams used because when the number of N increases, the model performance remains the constants.

TF-IDF: is also one of the most features modelling methods used in cyber threat information detection. TF-IDF is a measure of the importance of a word in a document within a dataset and increases in proportion to the number of times that a word appears in the document. More detail discussed in chapter two.

The feature vectors extracted by each of the above methods used for training the detection models and then by combining each feature extraction method, for instance, N-gram weighted by TF-IDF.

Word2vec: it takes a large corpus of text as input and produces a vector space that will be used when training a model for Amharic cyber threat information detection. Word2vec is a collection of models capable of capturing the semantic similarity between words based on the sentential contexts in which these words occur. It does so by projecting words into an n-

dimensional space, and giving words with similar contexts similar places in this space. Simply the model gives relationships between the words in the text. Using Word2vec help to get a feature model not only from our annotated dataset but also from the whole gathered data collected for this study.

Word embedding is one of the most preferred approaches when encoding text data to real-valued data for Machine learning tasks. When using the word embedding approach, we first break text into a list of tokens (words) and then assign a real-valued vector (embedding) of pre-decided length to each token. Other encoding approaches like one-hot, word frequency, Tf-Idf, etc. assign just a single scalar to each token. This limits the representation power of the tokens. With a single scalar, you can only represent a single meaning. With the word embedding approach where we assign a real-valued vector to the token, it gives more representation power to it. The same word can have different meanings in a different context and can have a different relationship with other words.

GloVe (Global Vectors) is an unsupervised learning algorithm that is trained on a big corpus of data to capture the meaning of the words by generating word embedding for them. These word embedding can be then used by other ML tasks that have different small datasets. The trained token embedding can be taken from **GloVe Embedding**. This saves us time to generate and train our own embedding which might not be possible with small datasets.

3.5. Machine Learning Classification Algorithm

The objective of this study is to classify cyber threat information from online news using a machine learning algorithm on categorized and labelled datasets. We use a multiple machine learning algorithm for comparison and accuracy of the detection. The algorithms are selected because of their good classification performance.

Support Vector Machine (SVM): is a supervised machine learning algorithm that can be used in classification problems. SVM algorithm performs classification by finding hyper plane in n-dimensional space that separates the classes or data points in feature space. Hyper planes are decision boundaries that help classify the data points. A hyper plane in n-dimensions is an affine subspace of dimension n-1. In general, the equation below shows the form of a hyper plane for a set of input point X.

$$w * x + b = 0 \quad (3.1)$$

Where: w are the support vectors to the hyper plane, the intercept (b) is found by the learning algorithm, and x is a set input data point. The SVM classifier for predicting a new input can be written as,

$$fw, (x) = g\ (wx + b) \quad (3.2)$$

Here, then f(x) >0 for points on one side of the hyper plane, and f(x)<0 for points on the other. This study used a one-vs-the-rest strategy of SVM classifier because SVM is inherently binary classifiers, but the goal here is to classify two datasets with three and two labelled classes. The approach of the one-vs-the-rest SVM classifier is to train a binary classifier for each class in the dataset.

Naïve Bayes (NB): is a probabilistic classifier technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, the NB classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. The equation for Bayes' Theorem

$$(c|x) = (x|c) / (c)P(x) \quad (3.3)$$

Where, P(c|x) is the posterior probability of class c given to data x, P($x|c$) is likelihood which is the probability of x data predicted of class c, P(c) is the prior probability of class c, and P(x) is the prior probability of the data x.

Decision Tree: One earlier classification algorithm for text and data mining is decision tree. Decision tree classifiers (DTCs) are used successfully in many diverse areas for classification. The structure of this technique is a hierarchical decomposition of the data space. Decision tree as classification task was introduced by D. Morgan and developed by J.R. Quinlan. The main idea is creating a tree based on the attribute for categorized data points, but the main challenge of a decision tree is which attribute or feature could be in parents' level and which one should be in child level. To solve this problem, De Mántaras introduced statistical modelling for feature selection in tree.

Conditional Random Field

Conditional random field (CRF) is a statistical model well suited for handling NER problems, because it takes context into account. In other words, when a CRF model makes a prediction, it factors in the impact of neighbouring samples by modelling the prediction as a graphical model. For example, a linear chain CRF is a popular type of a CRF model, which assumes that the tag for the present word is dependent only on the tag of just one previous word(Shabat, Omar, & Rahem, 2014). Conditional random fields are a group of models that are best suited to predict contextual tasks. For labelling, Conditional Random Field is used. It is typically used for sequence labelling or parsing information, for example, processing of language and CRFs Named Entity Recognition for the POS labelling. CRFs are used to forecast sequences. Denote x as the sequence of input states, i.e. the words of a sentence y as the output states i.e. the named entity tags

$$X = (x1,. . . . . .xn) \quad , \quad Y = (y1 , ...yn) \tag{3.4}$$

For conditional random field, we model a conditional probability

$$P(y1 , ...yn \,|x1,. . . . . .xn) \tag{3.5}$$

Below is the formula for CRF where Y is the hidden state (for example, part of speech) and X is the observed.

$$p(\mathbf{y}|\mathbf{x}) = \underbrace{\frac{1}{Z(\mathbf{x})}}_{\text{Normalization}} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \underbrace{\theta_k}_{\text{Weight}} \underbrace{f_k(y_t, y_{t-1}, \mathbf{x}_t)}_{\text{Feature}} \right\} \tag{3.6}$$

Broadly speaking, there are 2 components to the CRF formula:

Normalization: You may have observed that there are no probabilities on the right side of the equation where we have the weights and features. However, the output is expected to be

a probability and hence there is a need for normalization. The normalization constant Z(x) is a sum of all possible state sequences such that the total becomes 1. You can find more details in the reference section of this article to understand how we arrived at this value.

Weights and Features: This component can be thought of as the logistic regression formula with weights and the corresponding features. The weight estimation is performed by maximum likelihood estimation and the features are defined by us.

Bi-directional Long short time memory with Conditional Random Field

An improved bidirectional long short term memory conditional random field (BI-LSTM-CRF) model was used for named entity recognition, combining with word and char information in the process of training which has achieved good results. The output of the BiLSTM is then fed to a linear chain CRF, which can generate predictions using this improved context. This combination of CRF and BiLSTM is often referred to as a BiLSTM-CRF model. The bidirectional LSTM consists of two LSTM networks - one takes the input in a forward direction, and a second one taking the input in a backward direction. Combining the outputs of the two networks yields a context that provides information on samples surrounding each individual token.

Table 3-2: Text classification comparison

S.No	Model	Advantages	limitations
1	Support Vector Machine(SVM)	✓ SVM can model non-linear decision boundaries ✓ Performs similarly to logistic regression when linear separation ✓ Robust against over fitting problems(especially for text data set due to high-dimensional space)	✓ Lack of transparency in results caused by a high number of dimensions (especially for text data) ✓ Choosing an efficient kernel function is difficult (susceptible to over fitting/training issues depending on kernel) ✓ Memory complexity
2	*Decision Tree(DT)*	✓ Can easily handle qualitative (categorical) features ✓ Works well with decision boundaries	✓ Issues with diagonal decision boundaries ✓ Can be easily overfit

		parellel to the feature axis ✓ Decision tree is a very fast algorithm for both learning and prediction	✓ Extremely sensitive to small perturbations in the data ✓ Problems with out-of-sample prediction
3	Naïve Bayes(NB)	✓ It works very well with text data ✓ Easy to implement ✓ Fast in comparison to other algorithms	✓ A strong assumption about the shape of the data distribution. ✓ Limited by data scarcity for which any possible value in feature space, a likelihood value must be estimated by a frequents
4	Conditional Random Field(CRF)	✓ Its feature design is flexible ✓ Since CRF computes the conditional probability of global optimal output nodes, it overcomes the drawbacks of label bias ✓ Combining the advantages of classification and graphical modelling which combine the ability to compactly model multivariate data	✓ High computational complexity of the training step ✓ This algorithm does not perform with unknown words ✓ Problem about online learning (It makes it very difficult to re-train the model when newer data becomes available)
5	Bi-Directional Long Short Time Memory With Conditional Random Field(Bi-LSTM-CRF)	✓ BiLSTM is well aware of the context information in the character sequence. ✓ CRF helps improve the labelling accuracy at the sentence level. ✓ Combining the advantages of BiLSTM and CRF, the overall recognition accuracy will be improved	✓ BiLSTM-CRF take longer to train ✓ require more memory to train

3.6. Semantic similarity algorithms

Semantic similarity measures and algorithms are discussed in (Gomaa & Fahmy, 2013) and (Vijaymeena & Kavitha, 2016). They are classified as corpus-based and knowledge based. The corpus-based algorithms process text document collection, extract information from it and use the information obtained for determining similarity between text elements (words, phrases). The knowledge based algorithms derive semantic similarity by using information from semantic repositories (ontologies, semantic networks).

1) **Corpus-based similarity algorithms**

Hyperspace analogue to language inspects co-occurrences of words and puts them in semantic space as matrix with elements, which represent the degree of association between the corresponding row and column words. The algorithm measures the co-occurrence of a focus word to neighbouring words and weights it inversely proportional to the distance between the word and the focus word. The weight is inserted as element in the matrix.

Explicit semantic analysis measures the degree to which two texts are related. It implements vector-based representation of the texts and the cosine measure to assess how they are related.

Point-wise mutual information – information retrieval - semantic similarity between words is calculated by means of probabilities. High frequency of co-occurrence of words determines high score of the point-wise mutual information similarity measure. Its extension second-order co-occurrence point-wise mutual information sorts lists of neighbour words of the two input words. This provides for obtaining similarity between words, which do not co-occur frequently, but have the same co-occurring neighbouring words.

Normalized Google distance is obtained from the matches to a set of keywords returned by the Google search engine. When two words have similar meaning their Google distances are close to one another. Two words, appearing separately have infinite Google distance, while words appearing together have Google distance equal to zero.

Extraction of words with similar distribution using co-occurrences assumes that similar meaning appear in similar contexts. The method uses three words frame for determining co-occurrences. Similarity is determined by applying measures to the corresponding word

vectors. The basic measures used are similarity based on words' colocation sets and similarity based on sets of similar by distribution word sets.

Latent semantic analysis is based on the assumption that similar words appear in similar text fragments.

2) Knowledge-based similarity algorithms

There are two types of similarity measures – semantic similarity and semantic relatedness. They are based either on information content or on path length. They implement relations between words and concepts that are available in semantic repositories. Scoring function for calculating semantic similarity between texts T1 and T2 from (Vijaymeena & Kavitha, 2016) is described by equation (3.7).

$$\text{Sim}(T_1,T_2) = 1/2 \frac{(\sum w \varepsilon T1 \text{ maxsim(w,T2)} * \text{idf(w)}}{(\sum w \varepsilon T1 \text{idf(w)}} + \frac{(\sum w \varepsilon T2 \text{ maxsim(w,T1)} * \text{idf(w)}}{(\sum w \varepsilon T2 \text{idf(w)}} \quad 3.7$$

The following metrics identify words with high semantic similarity:

i. Based on the shortest *path length* between concepts obtained by the count of nodes and the maximum

taxonomy depth D – equation (3.8).

$$\text{Sim} = -\log \frac{length}{2D} \quad 3.8$$

ii. Based on function of the overlap of definitions, which is provided by dictionary and performs word

disambiguation;

iii. Based on a combination of the estimated depth of the two analyzed concepts c1 and c2 in taxonomy (Word Net) and the least common sub sumer's (LCS) depth equation (3.9).

$$\text{Sim} = \frac{2depthLCS}{depthc1 + depthc2} \quad 3.9$$

iv. Based on a function of the information content *IC* of the least common subsumer of two concepts,

which is calculated by the probability (−log $P\ c$ of the presence of the concept c in the text collection equation (3.10).

$$\text{Sim}= \frac{2IcLcs}{ICc1+ICc2} \qquad 3.10$$

Besides the corpus - and knowledge based similarity measures hybrid measures combining several ones are implemented as well. As shown in (Aggarwal, Asooja, & Buitelaar, 2012) semantic similarity of sentences is determined by implementing semantic
relatedness measure over the sentence followed by knowledge-based semantic similarity scores calculated for the words that appear in the same roles in both sentences.

3.5.1. Latent semantic analysis

Latent semantic analysis / indexing (LSA / LSI) (Boling & Das, 2014) ,(Kao et al., 2012),(Landauer, McNamara, Dennis, & Kintsch, 2013), is a major approach in the text mining processing. It has proven to be the most widely applied corpus-based similarity measure. It facilitates the capture of the most descriptive features of the document meaning and thus the elaboration of a document vector space with reduced dimensionality. The basic assumptions of LSI are (Kao et al., 2012):

1) The meaning of a word is obtained as an average of its presence in all documents.
2) The meaning of multi-word constructs is determined by the way the words are configured within it.
3) The latent (hidden) associations among words are discovered by the inspection of word co-occurrence with each single word.

Associations between points in a reduced vector space are induced on the basis of determining correlations in their distribution. This semantic knowledge LSI determines by an inductive process on the basis of the way words and phrases are distributed within the documents.

The processing steps of the LSI algorithm (Rozeva & Zerkova, 2017) are presented in Figure 3-2

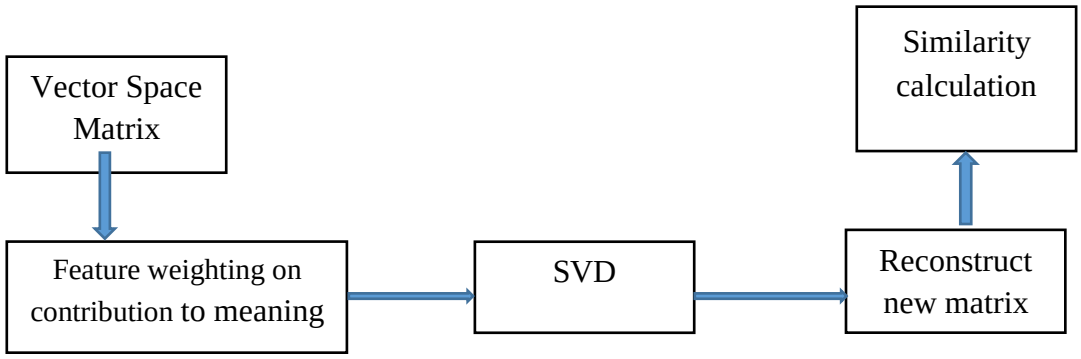


Figure 3-2 : LSI Algorithm

Step1: Implements frequency-to weight transformation of features within a document. The transformation addresses the growth rate of the understanding of the text meaning. Equation (3.11) implements the transformation.

$$weight_{ij}^{loc}=\log (freq_{ij}+1) \qquad 3.11$$

Further on the global feature's frequency within the document collection is calculated after equation (3.12).

$$Weight_{ij}^{glob} = 1 + \sum_{j=1}^{n} pij \,.\, \log(pij) / \log(n),\ \ pij = frec_{ij}^{loc}/\Sigma^{n}_{j=1} frec_{ij}^{loc} \qquad 3.12$$

The sum of local the feature's local frequencies within the documents represents its global frequency. The weighted value of a feature is calculated by equation (3.13).

$$Weight_{ij}^{feature} = weight_{ij}^{loc} \ / weight_{ij}^{glob} \qquad 3.13$$

The frequency-to-weight transformation assesses the contribution of a feature to the meaning of a text, as the co-occurrence, expressed by local and global frequencies, does not imply high informative value. The implemented weighting method attaches higher weight to features that have high local frequency and low global frequency.

Step2: Singular value decomposition of the matrix, obtained from Step1 to three matrices after equation (3.14).

$$M_{mxn} = T\Sigma D^{T} \quad 3.14$$

Matrices T_{mxm} and D^{T}_{nxn} are orthogonal, where T is the matrix representation of features and D is the matrix representation of documents. Matrix Σ is "cellular diagonal" matrix, i.e. it has the diagonal matrix, r < min (m, n), on the main diagonal and all other elements on the main diagonal are 0. After the SVD transformation of D the singular values of the matrix M are obtained on the main diagonal of D in a descending order. The singular values of M, contained in D, represent the dimensions of the meaning of the elements (words and concepts) of the analyzed text.

Step3: Construction of new lower dimensionality matrix M^s can be performed by the substitution of some of the singular values with 0. The rule for the substitution is from the lowest to the greatest singular value. The remaining non-zero singular values are s, s<r (the number of values in D). The construction of M^s is achieved by calculation by replacing Σ with Σ^s. The value of s, which represents the number of dimensions of the newly constructed matrix of the semantic representation of co-occurrence of words within the documents, is provided as parameter. The matrices of the feature T^s and document D^T representations in the lower dimensionality space are obtained by multiplication with the singular value matrix Σ^s.

Step4: Similarity of vectors in the reduced dimensionality space is determined by the cosine of the angle between them. Cosine = 1 (parallel vectors) implies that the features are synonyms, while cosine = 0 means that they are dissimilar. Calculation of the cosine of vectors v1 and v2 is performed by equation (3.15)

$$\text{Cos}(\Theta) = \frac{v1.v2}{||v1||.||v2||} \quad 3.15$$

3.5.2. Semantic Similarity calculation

1. Word-to-word similarity with word vectors w1 and w2 in reduced space is calculated from a matrix, obtained by equation (3.16).

$$M_s M_s^{T} = T_s T_s^{T} \quad 3.16$$

In equation (3.9) $^{T}{}_{s}$ $T^{T}{}_{s}$ denote the transposed matrices. Similarity between word vectors w1 and w2 is calculated by the cosine of the vectors in the i^{th} row and j^{th} column of the matrix.

2. Document-to-document similarity with document vectors d1 and d2 in reduced space is calculated from a matrix, obtained by equation (3.17).

$$M_s^T M_s = D_s D_s^T \quad 3.17$$

Similarity between document vectors d1 and d2 is calculated by the cosine of the vectors in the i^{th} row and j^{th} column of the matrix, obtained in equation (3.17).

3. Word-to-document similarity with vectors w1 and d1 is calculated by the cosine of the i^{th} row and j^{th} column of M_s, divided by $\|w!\|$ and $\|d1'\|$ where equation (3.18)

$$W1' = w1\sqrt{\Sigma}, \quad d1' = \sqrt{\Sigma}\, d1 \quad 3.18$$

Similarity between document vectors d1 and d2 is calculated by the cosine of the vectors in the i^{th} row and j^{th} column of the matrix, obtained in (3.18)

3.7. Performance Evaluation Metrics

To validate the performance of our proposed model, different performance parameters have been considered: precision, recall, accuracy, and confusion matrix as evaluation matrices.

1) Confusion matrix

Confusion matrix metrics are performance measures which help us to find the accuracy of classifier.

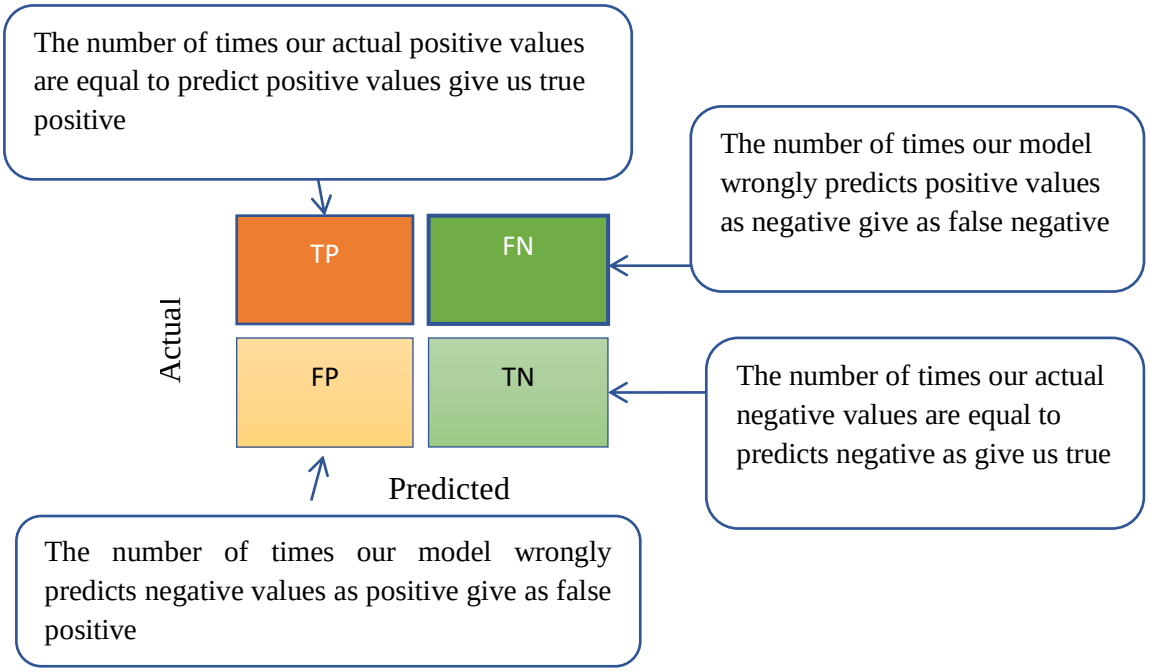


Figure 3-3 : Confusion matrix metrics

2) Accuracy

The accuracy is used to find the portion of correctly classified values. Tell us how classifier is right.

$$\text{Accuracy} = \frac{\text{Tp} + \text{Tn}}{\text{Tp}+\text{Tn}+\text{Fp}+\text{FN}} \qquad 3.19$$

3) Precision

Precision is used to calculate the models ability to classify positive values correctly.it answers the question when the model predicts positive values.

$$\text{Precision} = \frac{Tp}{\text{Fp}+\text{Tp}} \qquad 3.20$$

4) **Recall**

Recall is used to calculate the models ability to predict positive values. How often does the model actually predict the correct positive values?

$$\text{Recall} = \frac{Tp}{\text{Fp+Fn}} \qquad 3.21$$

F1-score: - The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall. A perfect model has an F1-score of 1.

$$\text{F1} = \frac{2}{1/\text{recall x } 1/\text{precision}} = 2 \text{ x } \frac{\text{precision x recall}}{\text{precision x recall}}$$

$$= \frac{tp}{tp+1/2(fp+fn)} \qquad 3.22$$

CHAPTER 4 : PROPOSED SYSTEM NEWS CLASSIFICATION

This chapter discusses the proposed solution of cyber threat information classification using machine learning techniques suitable to solve the problem of understudy. This study identifies cyber-attack features for identification of cyber threat information from online news.

4.1. The proposed Cyber threat information classification Architecture

The Bi LSTM-CRF classifier will be trained and later used to classify the news. BiLSTM-CRF model learns weights of the feature from the training datasets and produces a model in labelling the sign from the test dataset. The identified features are used to assign the sentence with proper label stating whether the sentence contain any cyber-attacks incidents or not manually(Arulanandam et al., 2014). The trained model is eventually can detect the news related to the cyber-attacks incidents accordingly.

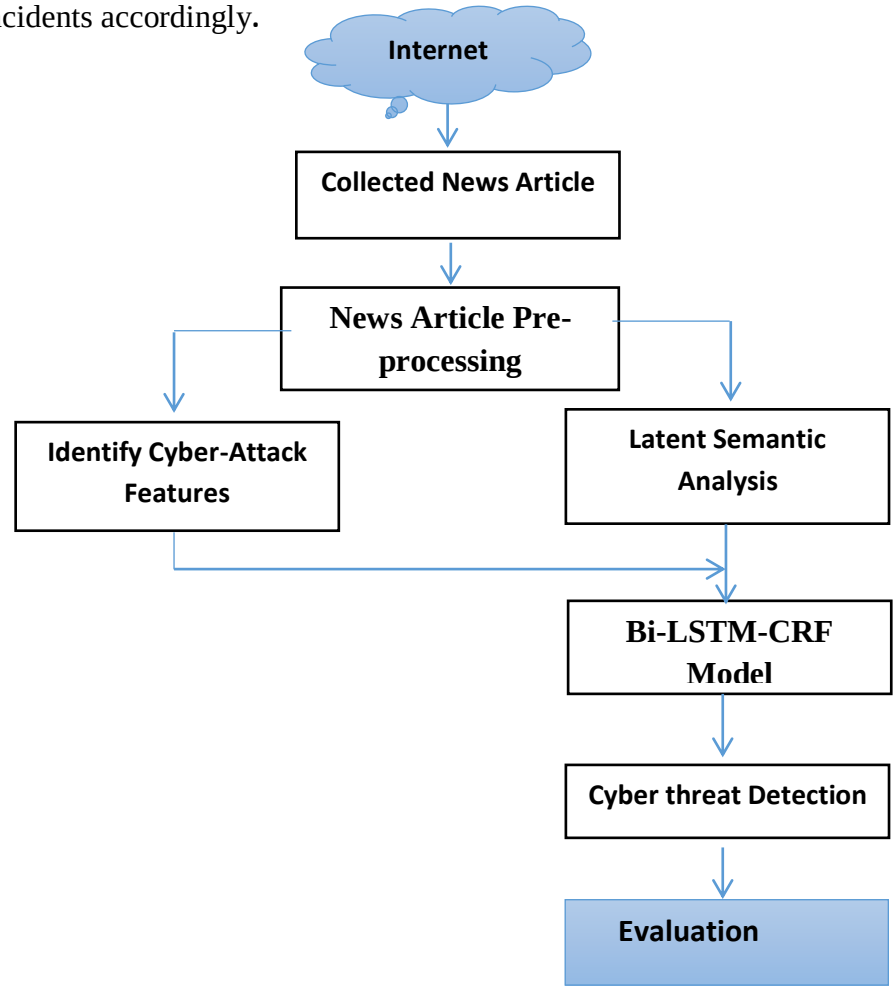


Figure 4-1: Cyber threat Detection Architecture

4.2. Proposed Text Pre-processing

Pre-processing is an important part to complete the classification model. Following dataset preparation, pre-processing steps are conducted to feed the clean dataset to the machine-learning model. This pre-processing is performed based on text pre-process techniques such as removing (cleaning) punctuations and special characters), stemming, and tokenization.

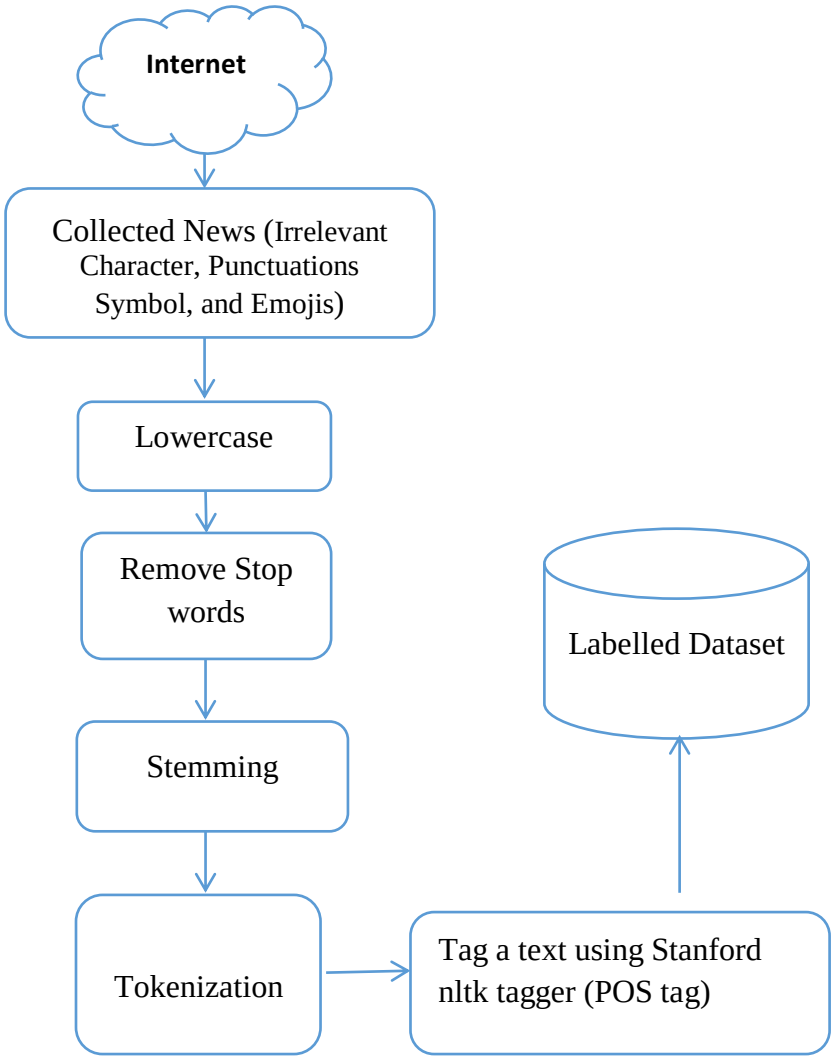


Figure 4-2: Text pre-processing

Step 1: Removing (Cleaning) Irrelevant Character, Punctuations Symbol, and Emojis

The dataset is created using text news, which is posts from online usually contain a special character, punctuations, symbol, and Emojis to express different opinions and feelings. This cleaning task removes all irrelevant special characters, symbols, and Emojis.

Pseudo code: 4.1 Removing Irrelevant Characters

Input: text in a dataset
Output: clean text
Begin:
1. Read the text in the dataset;
2. While (! end of the text in a dataset):
If the text contains special_char [,'! @#$%^&*] then
Remove special_char
If the text contains symbol [<>‹› « »=:-`~_/] then
Replace symbol and add space
If text contain English_word_num = [a-z A-Z] [0-9] then
Remove English_word_num
If text contain number = [0-9] then
Remove number
If a text contains emoji = [.....] then
Remove emoji
If a text contains extra white space then
Trim the text
3. Return clean_text;
End:

Algorithm 4. 1 Removing Irrelevant Characters (Pre-processing)

Step 2: Sentences can contain a mixture of uppercase and lower case letters. Multiple sentences make up a text document. To reduce the problem space, the most common approach is to reduce everything to lower case. This brings all words in a document to the same space, but it often changes the meaning of some words, such as “US” to “us” where the first one represents the United States of America and the second one is a pronoun.

Step 3: Remove Stop words are used to remove Common word such as “a”, “is”, “the” etc. which made the content of the text more meaningful.

Step 4: Stemming process will separate the root word from word that appears in the text in order to minimize the frequency. For an example, words like “attacking” and “attacks” will undergo a stemming process and later on, the result will be stored in database as “attack” in which preventing the redundancy of word in the database, saving spaces in memory. Porter Stemmer is often used in stemming process as excellent trade-off between speed, liability and accuracy.

Step 5: Tokenization is the process of splitting the given text into smaller pieces called tokens. Words, numbers, punctuation marks, and others can be considered as tokens.

4.3. Named Entity Recognition (NER)

Named Entity Recognition (NER) is a subfield of Natural Language Processing (NLP) that involves identifying and classifying named entities in unstructured text data. Named entities refer to real-world entities present in the text. Some common examples of named entities are persons, organizations, locations, dates, and time expressions.

POS Tagging

POS Tagging simply labelling words with the appropriate Part-Of-Speech . The part of speech describes how a word is used in a sentence. Nouns, pronouns, adjectives, verbs, adverbs, prepositions, conjunctions and interjections these are the eight main parts of speech. POS tagging is a supervised approach to learning which uses features such as the previous word, next word, first letter capitalization, etc. NLTK has a pos tags feature and operates after the tokenization process. Example Sentence: “John love to play score”

‘John’, “NNP”, ‘Love’, “VBZ”, ‘to’, “To”, ‘Play’, “VB”, ‘Soccer, “NN”

IOB Labeling in Named Entity Recognition

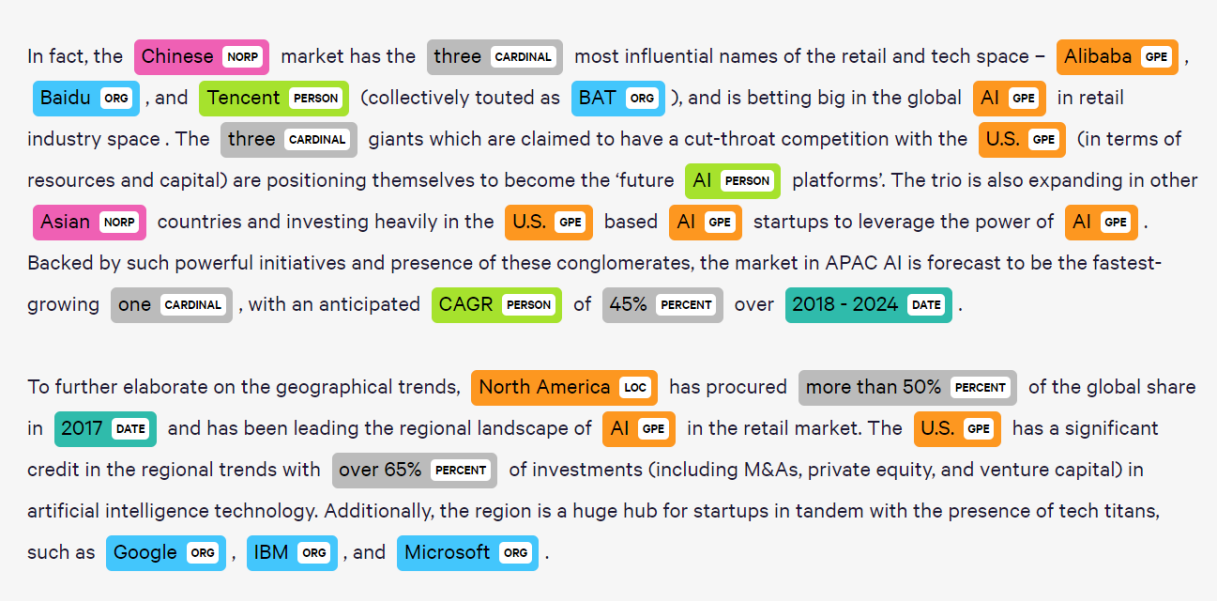


Figure 4-3: Sample text tagged with named entities

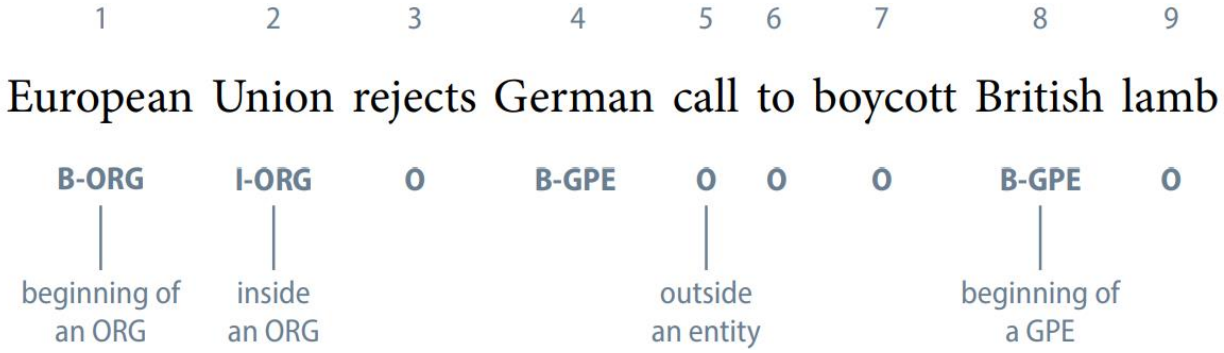
Essential info about entities:

- geo = Geographical Entity
- org = Organization
- per = Person
- gpe = Geopolitical Entity
- tim = Time indicator
- art = Artifact
- eve = Event
- nat = Natural Phenomenon

IOB labelling (short for **Inside, Outside, Beginning labelling**) is a popular technique used in Named Entity Recognition (NER) to annotate and categorize text data.

IOB labelling involves labelling each word or token in a text sequence with a specific tag that indicates whether it is part of a named entity or not. In IOB labelling, each token in a sequence is assigned one of three tags:

- **B (Beginning):** the token marks the beginning of a named entity.
- **I (Inside):** the token is part of a named entity, but it is not the first token.
- **O (Outside):** the token is not part of a named entity.



4.4. Identify Cyber Attack features

The cyber-attack features will be identified from the collected data. Features such as the name of the cyber-attacks, its types, the respective threat actors, and the organization involved and so on. These features are being compared to other researchers work and have it validated from the cyber security expert. In the same time, any related information obtained from the text is extracted and will be recorded. The information or we called it as terms or keywords are used for updating the dictionary of the cyber-attacks. All the terms currently present are collected through manual reading of the cyber-attack news from multiple sources. Then, to have a better news type classification, a features model is created consisting of several sets of cyber-attack features. The cyber-attack features are Types of cyber threat, Threat actor, Organization Affect, Country affect.

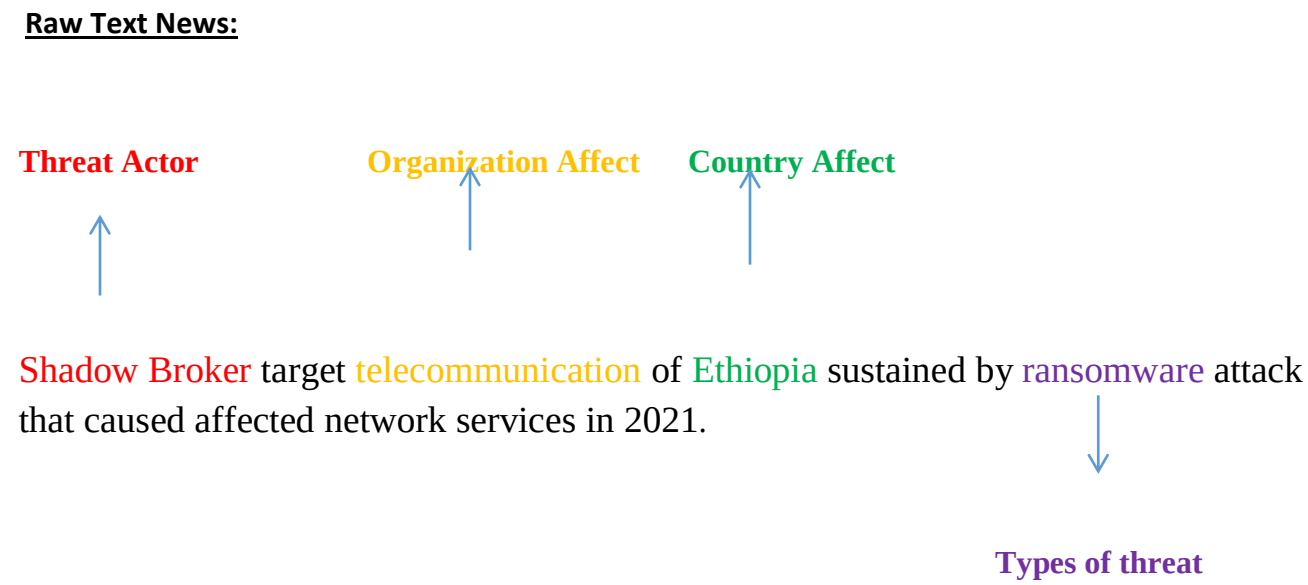


Figure 4-4: Cyber-attack feature identification from news sources

4.5. Cyber threat News similarities

Latent Semantic Analysis (LSA) is an approach to measure similarity between texts (Rozeva & Zerkova, 2017). LSA is shown to capture significant portions of the meaning not only of individual words but also of whole passages such as sentences, paragraphs, and short essays. LSA arose from the problem of how to find relevant documents from search words. LSA is designed to overcome the fundamental problem of comparing words to find relevant documents by instead comparing the concepts or meanings behind the words to retrieve documents. Singular-value decomposition is then used to arrange the matrix to reflect the major associative patterns in the text and ignore the smaller less important words and concepts.

The process of ignoring less important words contains removing extremely common words which bring little value to analysing a piece of text, such words are called stop words and are usually filtered out before processing the text.

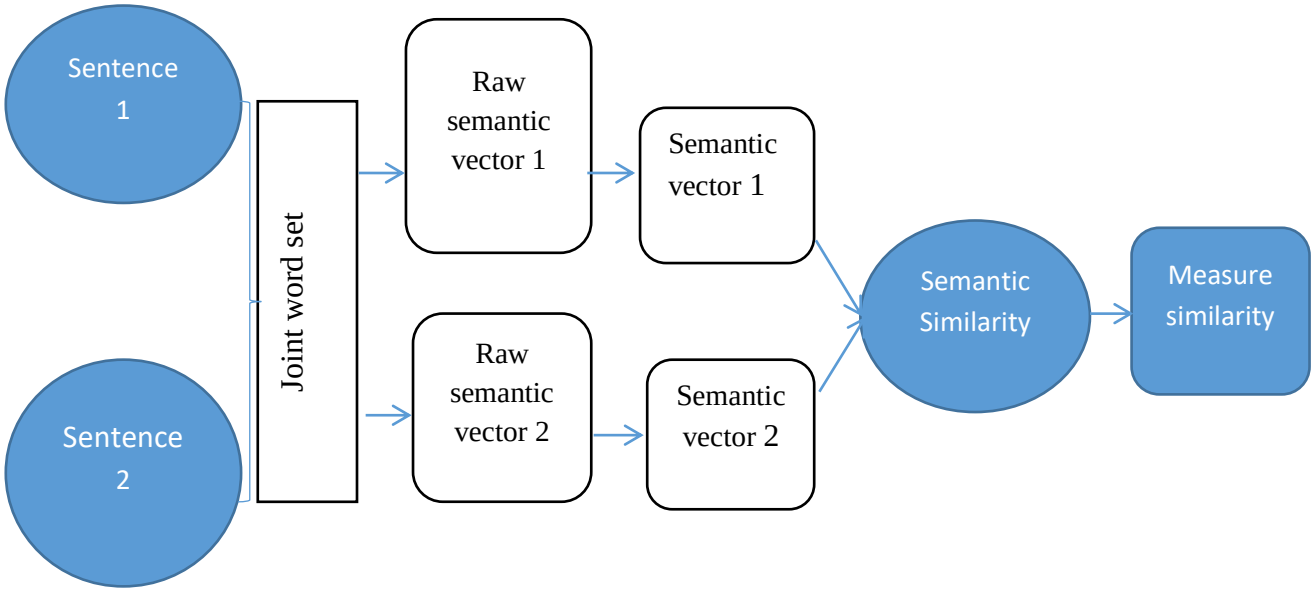


Figure 4-5: Cyber threat news similarity measure

Sentence Similarity is used to efficiently compute the similarity between texts or sentences. Figure 3.5 shows how two different sentences are computed. The significant words in each semantic vector are then weighted by using information content derived from a text corpus. Finally, the sentence similarity is computed by comparing the vectors to each other.

Steps to calculate Cosine Similarity

Cosine similarity calculates similarity by measuring **the cosine of angle between two vectors**.

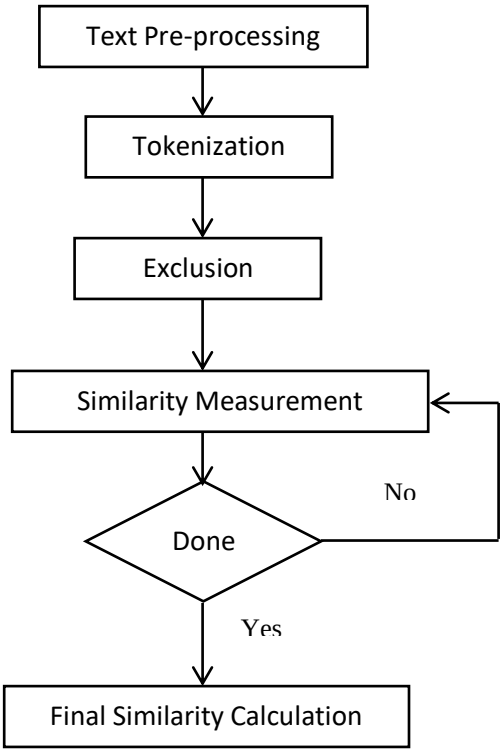


Figure 4-6: Cyber text news similarity calculation

Mathematically speaking, **Cosine similarity** is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them

Sim (v1, v2) = $\frac{\sum_{i=0}^{n} v1.v2}{||V1||.||V2||}$

where ||v1||= $\sqrt{\sum_{i=0}^{n} v1^2}$, ||v2||= $\sqrt{\sum_{i=0}^{n} v2^2}$

4.6. Proposed BiLSTM-CRF model Building

This study aims to develop a model that enables to identify cyber threat information from online news. This study builds a classification model mainly by using machine-learning models, Bi-LSTM-CRF. For creating a classifying model, a total an instant of 2019 records was collected and then remove redundant records prepared 1488 for training and testing using learning algorithms. In this study, the classification process outcome is classified from a given input or features.

One problem with the linear chain CRFs is that they are capable of capturing the dependencies between labels in the forward direction only. If the model encounters an entity like "Addis Abeba University" it will likely tag the Addis token as a name, because the model is "blind" to the Abeba token that appears downstream. One way to resolve this challenge is to introduce a bidirectional LSTM (BiLSTM) network between the inputs (words) and the CRF. The bidirectional LSTM consists of two LSTM networks - one takes the input in a forward direction, and a second one taking the input in a backward direction. Combining the outputs of the two networks yields a context that provides information on samples surrounding each individual token. The output of the BiLSTM is then fed to a linear chain CRF, which can generate predictions using this improved context. This combination of CRF and BiLSTM is often referred to as a BiLSTM-CRF model and its architecture is shown in Figure 4.7

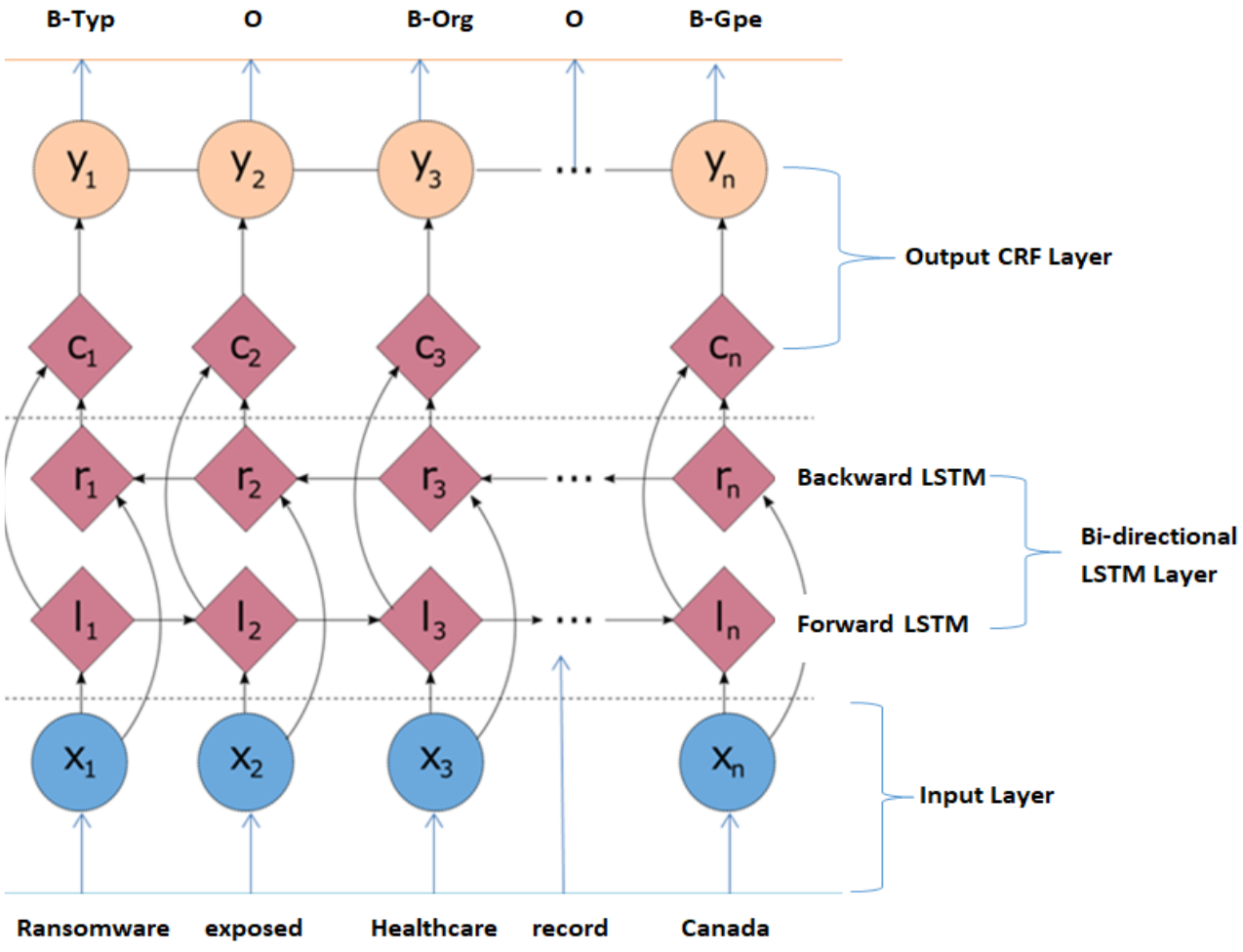


Figure 4-7: Building model using Bi-LSTM-CRF

BiLSTM is well aware of the context information in the character sequence, and CRF helps improve the labelling accuracy at the sentence level. Combining the advantages of BiLSTM and CRF, the overall recognition accuracy will be improved.

Input Layer: -the input is words; so, the effect of entity recognition depends heavily on the effect of word segmentation. A global dictionary with size v is constructed, that is, a collection of v distinctive characters in the training set, in which each character corresponds to an identifier. A sentence containing n characters is one-hot encoded as an n × v-dimension matrix, denoted as $W_i = (w_1, w_2, \cdots, w_n)$, where w_i represents the vector of the i^{th} character of the sentence.

Embedding Layer:- Texts news from the input layer are converted into vectors so that computers can calculate. The traditional one-hot representation cannot capture the semantic relationship between words, and it will also bring in dimensional disasters and data-sparse problem. Distributed representation methods, however, can map words into fixed-length, low-dimensional, dense vectors. The semantic similarity between words can be measured based on the distance of the words in the vector space, which well overcomes the shortcomings of one-hot representation. We fine-tune the initial word embedding, modifying them during back propagation of the model. Each incoming identifier of a character w_i from the input layer is mapped into a d -dimension dense vector Xi = (x1, x2, ⋯ , xd), where d is the embedding size that defines the number of features used to represent character. The vector is obtained according to Equation.

$$X_i = Hw_i \qquad 4.1$$

where H denotes a d × v -dimension pre-trained weight matrix. The vector is then transmitted to BiLSTM Layer as the input. At this point, a sentence containing n characters is mapped into a dense matrix X (with the shape of n × d) from the initial sparse one-hot matrix W (with the shape of n × v).

BiLSTM Layer:- This layer is used to extract sentence features. Every character vector Xi = (x1, x2, ⋯ , xd) is taken as the input of BiLSTM Layer, and the output hidden state of the i^{th} character $h^L_i = (h^L_1 , h^L_2 , \cdots, h^L_{d)}$ on the left is obtained through the forward LSTM. Likewise, after backward LSTM, the output hidden state $h^R_i = (h^r_1, h^r_2 , \cdots, h^r_d)$ on the right

is obtained. h^L_i and h^R_i are then spliced to obtain the output hidden state $c_i = h^L_i , h^L_i$ of each position i, and finally, the sentence level hidden state sequence C = (c1, c2, ⋯, cn) is obtained. A full connection layer is used to map the hidden state vector to k dimensions, where k is the number of labels in the label set, so as to extract features and provide a probability distribution matrix $P = (p_1, p_2, \cdots, p_n)$. Element $p_{i,j}$ represents the probability of classifying the character w_i as the j^{th} label.

The pseudo code of the BiLSTM Layer is shown in Algorithm 4.2.

Pseudo code: 4.2 building model Algorithm
Input: Pre-trained character embedding X: Output: Probability distribution matrix P of the input sequence (1) Step 1: The character vectors from X are sent into the forward LSTM layer (2) for i ∈ length (X) do (3) send Xi to BiLSTM Layer (4) end for (5) Step 2: The state of the cell in the current LSTM network is updated (6) $f_t = \sigma(W_f [h_{t-1}, x_t] + b_f)$ (7) $i_t = \sigma(W_i [h_{t-1}, x_t] + b_i)$ (8) $\tilde{c}_t = \tan h(W_f (h_{t-1}, x_t + b_c)$ (9) $c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$ (10) $o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$ (11) $h_t = o_t * \tan h(c_t)$ (12) Step 3: The character vectors from X are sent into the backward LSTM layer and the above 2 steps are repeated (13) Step 4: The forward and backward sequences of hidden layers are spliced to obtain a sentence-level hidden state sequence C rich in contextual information (14) Step 5: C is sent into a full connection layer and the prediction matrix P is obtained (15) Return P;

Algorithm 4.2. Bi-LSTM-CRF model Building Algorithm

CRF Layer: - This layer carries out sentence-level sequence labelling to ensure the generation of the globally optimal labelling sequence. The output of the BiLSTM Layer p_i is independent of each other, ignoring the strong dependence between its preceding label p_{i-1} and its subsequent label p_{i+1} . The CRF layer can automatically obtain some restrictive rules from the training data and conduct sentence-level adjustment, which will reduce the probability of illegal sequences and improve the accuracy of label sequence prediction. Considering the need to add a start state at the beginning of a sentence and a stop state at the end, the data structure of the CRF layer is a (k + 2) × (k + 2) state transfer matrix A. $A_{i,j}$ represents the transfer score from the i^{th} label to the j^{th} label. For a sentence W = (w1, w2, ⋯, w_n) as the input, assuming that a predicted label sequence y = (y1, y2, ⋯, y_n) is obtained, the score of the prediction is defined as S(W, y) =$\sum_{i=0}^{n} Ayi, yi + 1 + \sum_{i=0}^{n} Pi, yi$ Where $P_{i,yi}$ is the probability of the i^{th} position of BiLSTM outputs being y_i , and $A_{yi\ ,yi+1}$ is the transfer probability from y_i to y_{i+1}.

4.7. Models Evaluation and Testing

Model evaluation and testing are critical in our study in order to test and estimate the performance of machine learning models trained on the cyber threat news dataset. Model evaluation is very important to compare and select the best model for further classification. In this study, the effectiveness of the classifier model is evaluated by using different performance evaluation metrics appropriate for models; these metrics are precision, recall, f1_score, confusion matrix are used as discussed in chapter three.

CHAPTER 5 : IMPLEMENTATION AND EXPERIMENTATION

The section describes the implementation of the proposed solution for the cyber threat news classification by using the methodology discussed in chapter three and implementing the proposed architecture presented in chapter four. In this study, experiments were done using a well-organized dataset. Finally, a prototype for cyber threat news classification was developed by using the best model.

5.1. Implementation Environment

In this study, several tools and packages are used to implement the proposed solution. Python programming language has used this study for implementing and experimenting with each proposed solution from the data pre-processing to the final phase, which is the model-building phase and prototype development. The reason to choose python is that it is popular and it is the language of choice for developers, researchers, and data scientists that make the machine learning model development simple and clear. Besides, Python is the general-purpose programming language that can be used for processing text, numbers, images, scientific data easily. It is also simple and easy to understand.

5.2. Anaconda Distribution

Anaconda is a free and open-source distribution of the Python and programming languages for scientific computing (data science, machine learning applications, large-scale data processing and predictive analytics etc.) that aims to simplify package management and deployment. Package versions are managed by the package management system Conda on Linux, window and Mac OS. With over 12 million users worldwide, it is the industry standard for developing testing and training on a single machine enabling individual data scientists to:

1) Quickly download 1,500+ Python/R data science packages.
2) Manage libraries dependencies and environments with Conda.
3) Develop and train machine learning and deep learning models with scikitlearn, Tensor flow, and keras.
4) Analyze data with scalability and performance with, NumPy and pandas.
5) Visualize results with Matplotlib, Bokeh, Datashader, and Holoviews

Python is a high level, interpreted object oriented scripting language that is easy to learn, read and maintain. It supports both functional and object oriented programming. It can be easily integrated with other programming languages. One of the reasons why python is mostly used for machine learning applications is its syntax. Since the syntax of python is math like, it is easy for programmers to express their ideas mathematically. It has particular tools that are very helpful in developing machine-learning applications. Frameworks, libraries and extensions like Numpy make python to accomplish the tasks easily. Languages like Java, Ruby need hard coding because of their complexity whereas python is considered as a toy language because of its simple syntax and many features. Since python can do a lot thing easily, which helps for complex set of machine learning tasks it is considered as a best language for implementing machine-learning tasks.

Table 5-1: Tools and Python Packages for Implementation

Tools and packages	Version	Description
Anaconda navigator	1.10.0	Help us to launch development applications and easily manage anaconda packages, environments, and channels without the need to use command-line.
Jupyter notebook	6.0.3	It is a free web application that enables us to create and share documents. It is useful for data cleaning and transformation, modelling, data visualization, and machine learning.
Python	3.7.4	Python is a popular platform easy to learn, powerful programming language to develop a machine learning application.
Microsoft Excel	2016	For dataset preparation.
Microsoft Word	2016	For documentation purpose.
Scikit-learn	0.23.1	Is a python module used in machine learning for handling basic ML tasks clustering, regression, and classification?
Pandas	1.0.5	Pandas is an important package providing fast, flexible and expressive data structures designed to make working with “relational” or “labeled” data. It enables integrating and filtering of

		data, as well as loading it from other external sources like Excel and handling the dataset.
Numpy	1.19.5	Array processing for numbers, strings, and objects. This study uses it for handling converting the text to numeric data for features, training, and testing the model
Matplotlib	3.2.2	Publication quality figures in python. This study uses it for data and results visualization.
Seaborn	0.10.0	Statistical data visualization
Flask server	1.1.1	Micro framework for making web services in python. This study uses it for implementing the prototype for selected models
NLTK		NLTK is a toolkit build for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets
Neat text		a simple Natural Language Processing package for cleaning text data and pre-processing text data. It can be used to clean sentences.
gensim		Gensim is a free Python library designed to automatically extract semantic topics from documents, as efficiently (computer-wise) and painlessly (human-wise) as possible. Gensim is designed to process raw, unstructured digital texts ("*plain text*").
Tensor Flow		The Tensor Flow platform helps you implement best practices for data automation, model tracking, performance monitoring, and model retraining.
Keras		Keras is a neural network Application Programming Interface (API) for Python that is tightly integrated with Tensor Flow, which is used to build machine learning models.

5.3. Deployment Environment

The tools and packages that are discussed above in section 6.1 have been deployed on a personal computer Processor Intel(R) Core(TM) i5-5200U CPU @ 2.2 GHz, 4 Logical Processor(s), 4 Gigabyte of physical memory, 931.51 Gigabyte hard disk storage capacities and Microsoft Windows 10 Enterprise,64bit operating system.

5.4. Dataset description

Data is collected from different online cyber threat news article such us Securityweek.com, Future.com, BBCNews.com, FireEye.com and other annual cyber threat news reports in the form of .csv format. The data is pre-processed using neat text library. Then, identify cyber-attack features i.e. types of attack, the affected countries, the affected organization and the threat actors by manual reading of news article. After tagging and labelling the collected Text News, Classification using hybrid Bi directional long short time memory with Conditional random fie**l**d is implemented using python programming language in anaconda environment. Keras and Tensor flow which is an open source software library focused on machine learning applications. The pre-processed text is tagged using the proposed cyber-attack features model guided with the new dictionary. This process is done manually via Python programming. Our dataset is collected from online sources a total of 2019 news that contain cyber threat information. After pre-processing and removing redundancy news, 1488 clean dataset is prepared for training and testing.

According to IOB format (short for **inside**, **outside**, **beginning**) is a common tagging format for tagging tokens in a chunking task in computational named-entity recognition.

Table 5-2: Tagged cyber- attack features

Egyptian	hacker	target	Ethiopian	telecommunication	sustained	by	ransomware	attacks
B-Act	**I-Act**	**O**	**B-gpe**	**B-org**	**O**	**O**	**B-Typ**	**O**
Threat actor			**Country affected**	**Organization affected**			**Types of threat**	

Base on the proposed cyber-attack features:

i. Threat actor: **Egyptian hacker,**
ii. Types of threat: **ransomware**
iii. Organization Affected on: **telecommunication**
iv. Country Affected**: Ethiopia**

This study defines a set of features for extracting information on types of threats, threat actors, organization affected and country affected from online news documents. These features are grouped based on POS tagging.

5.5. Pre-processing Implementation

For machine learning algorithms to work, it's necessary to convert **raw data** into a **clean data** set, which means we must convert the data set to **numeric data**. We do this by encoding all the **categorical labels** to column vectors with binary values. **Missing values** or NaNs (not a number) in the data set is an annoying problem. You have to either drop the missing rows or fill them up with mean or interpolated values.

```
# importing necessary library and un pre-processed dataset

import pandas as pd

# load Dataset

data = pd.read_csv('Data_with_Feature_kifle.csv', encoding = "ISO-8859-1")

data.head()
```

Figure 5. 1 Python code for loading the panda's library and dataset

5.6. Implementing LSA model

Latent Semantic Analysis is a classical tool for automatically extracting similarities between documents, through dimensionality reduction. A term-document matrix is filled with weights corresponding to the importance of the term in the specific document (term-frequency/inverted document frequency in our case) and then is reduced via Singular Value Decomposition to a

lower dimensional space called the concept space(Korzycki, Gatkowska, & Lubaszewski, 2017).

LSA is a system and hypothesis used to extract and represent the contextual meaning of a number of words based on a larger quantity of words. This determines the relevance of a paragraph or sentence to the rest of the content. The reasoning behind this is that it enables people create a link between one word and another based on a given topic and try to prove that each word is somehow linked to the other. LSI system categorizes all words with similarities into groups and ensures that the ones placed together actually have some form of common ground. It takes into account the keyword and looks for content that contains that keyword and categorizes them in terms of relevance based on the number of times that keyword appears. However, all documents are examined individually then ranked in order of which contains the most keywords and therefore most likely more relevant to the topic. Similarity of documents is implemented using gensim package of LSI.

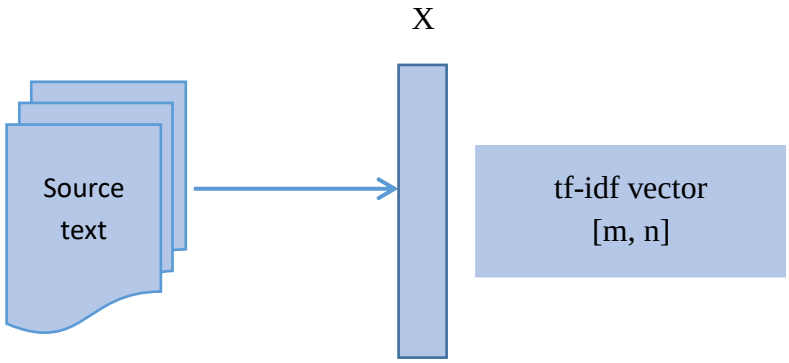
```
  #import library packages for implementing LSI model
from gensim import corpora, models, similarities


print(documents)
texts = [[word for word in document.lower().split() if word not in stoplist] for
        document in documents]
tfidf = models.TfidfModel(corpus) # step 1 -- initialize a model
print(tfidf)
lsi = models.LsiModel.load('model.lsi') # try to load above saved model
print(lsi)
index = similarities.MatrixSimilarity(lsi[corpus]) # transform corpus to LSI space and
        indexize it
print(index)
sims = index[vec_lsi] # calculate degree of similarity of the query to existing corpus
print(sims)
sims = sorted(enumerate(sims), key=lambda item: -item[1])  # sort output object as per
        similarity ( largest similarity document comes first )
print(sims)
```

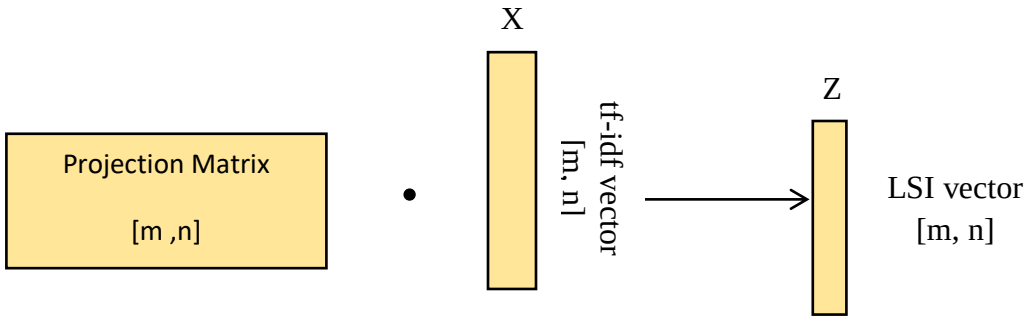
The first step in comparing the two pieces of text is to produce tf-idf vectors for them, which contain one element per word in the vocabulary. These tf-idf vectors are then projected down

to topics with LSI. Finally, the two LSI vectors are compared using Cosine Similarity, which produces a value between 0.0 and 1.0.

Step 1: Convert source text to tf-idf



Step 2: Project the tf-idf vector onto the topic space



Step4: reduce the dimensionality of the projection space using a truncated Singular Value Decomposition

Step 5: use the cosine as a similarity measure between documents.

Similar news articles are identified by using similarity calculation by latent semantic analysis.

Sample similarity calculations

Sentence similarity with semantic vectors v1 and v2 in reduced space is calculated from a matrix.

Sample: **Sentence1**: Malware attack polish bank, **Sentence2**: Spyware slowdown polish bank

Table 5-3: Similarity of two sentences

term	Malware	attack	slowdown	polish	bank
Malware attack polish bank	1	1	0	1	1
Spyware slowdown polish bank	0	0	1	1	1

Sim (v1, v2) = $\frac{\sum_{i=0}^{n} v1.v2}{||V1||.||V2||}$ where ||v1||= $\sqrt{\sum_{i=0}^{n} v1^2}$, ||v2||= $\sqrt{\sum_{i=0}^{n} v2^2}$

V1= (1,1,0,1,1) ,V2= (0,0,1,1,1) , **Sim (V1 , V2) = 0.75,** Therefore, the two sentences are similar in 75%

5.6. Machine Learning Models Implementation

To build machine-learning models using the proposed algorithm, we used a python keras learning library package and followed the conventional method that is importing the important library package for modelling and metrics for model evaluation.

```
import pickle

import operator

import re

import string

import pandas as pd

import numpy as np

from tensorflow import keras
```

```
import matplotlib.pyplot as plt

from plot_keras_history import plot_history

from sklearn.model_selection import train_test_split

from sklearn.metrics import multilabel_confusion_matrix,confusion_matrix

from keras_contrib.utils import save_load_utils

from sklearn_crfsuite.metrics import flat_classification_report,flat_f1_score

from keras import layers

import seaborn as sns

from keras import optimizers

from keras.models import Model

from keras.models import Input

from keras_contrib.layers import CRF

from keras_contrib import losses

from keras_contrib import metrics

import tensorflow as tf

from mlxtend.plotting import plot_confusion_matrix

from sklearn.preprocessing import OneHotEncoder
```

Next, we read and take a peek at the annotated dataset

```
data_df = pd.read_csv("textLastRecoverd_kifle.csv", encoding="iso-8859-1", header=0)

data_df.head()
```

The above code is loading the data from the labeled dataset as shown below.

	Sentence #	News	Pos	Tag
0	Sentence 1	shadow	NN	B-act
1	NaN	brocker	NN	I-act
2	NaN	target	NN	O
3	NaN	united	JJ	B-gpe
4	NaN	states	NNS	I-gpe

The meaning of the attributes is as follows:

- ✓ Sentence # - the Number of sentences that load from .csv format
- ✓ news – all news that contains all words that form individual sentences
- ✓ POS - Part of Speech tag for each word (Noun, verb, adjectives, etc.).
- ✓ Tag - IOB tag for each word(B-Act and I-Act represents the threat actor, B-Gpe and I-Gpe represents Country affected

Bi-LSTM-CRF model set the following model hyper parameters:

- ✓ DENSE_EMBEDDING - Dimension of the dense embedding
- ✓ LSTM_UNITS - Dimensionality of the LSTM output space
- ✓ LSTM_DROPOUT - Fraction of the LSTM units to drop for the linear transformation of the recurrent state
- ✓ DENSE_UNITS - Number of fully connected units for each temporal slice
- ✓ BATCH_SIZE - Number of samples in a training batch
- ✓ MAX_EPOCHS - Maximum number of training epochs

We add an input layer, an embedding layer (to transform the indexes into dense vectors, a bidirectional LSTM layer, and a time-distributed layer (to apply the dense output layer to each

temporal slice). We then pipe this to a CRF layer, and finally construct the model by defining its input as the input layer and its output as the output of the CRF layer.

We also set a loss function (for linear chain Conditional Random Fields this is simply the negative log-likelihood) and specify "accuracy" as the metric that we'll be monitoring.

```
#BiLSTM-CRF modelling

input_layer = layers.Input(shape=(MAX_SENTENCE,))


model = layers.Embedding(WORD_COUNT, DENSE_EMBEDDING,
embeddings_initializer="uniform",
input_length=MAX_SENTENCE)(input_layer)


model = layers.Bidirectional(layers.LSTM(LSTM_UNITS,
recurrent_dropout=LSTM_DROPOUT, return_sequences=True))(model)


model = layers.TimeDistributed(layers.Dense(DENSE_UNITS,
activation="relu"))(model)


crf_layer = CRF(units=TAG_COUNT)

output_layer = crf_layer(model)


ner_model = Model(input_layer, output_layer)


loss = losses.crf_loss

acc_metric = metrics.crf_accuracy

opt = optimizers.Adam(lr=0.001)
```

```
ner_model.compile(optimizer=opt, loss=loss,
metrics=[acc_metric])



ner_model.summary()
```

5.7. Model testing and evaluation

One of the important aspects of machine learning is evaluating the performance of learning algorithms.

We can inspect the loss and accuracy plots from the model training. They both look acceptable, and it doesn't appear that the model is overfitting. The result of loss and accuracy plots from the model training graph, as shown in Figure 6.6 (loss value of the CRF model), Figure 6.7 (training and testing CRF accuracy), and Table 63, can be seen as the predicted and actual values of the confusion matrix.

```
#plot CRF model accuracy and los
plot_history(history.history)
```

```
#plot confusion matrix
matrix =confusion_matrix(y_test.flatten(), y_pred.flatten())
fig, ax = plt.subplots(figsize=(8,6))
sns.heatmap(matrix, annot=True, fmt='d',
            xticklabels=data_df.Tag.unique(),
yticklabels=data_df.Tag.unique())
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.show()
```

Additionally in this study, we used flat_classification_report (), confusion matrix () methods, which are used to evaluate the performance of the built models using classify the value for the entire dataset. These methods give a result of performance evaluation metrics such as precision, recall, and f1-score.

CHAPTER 6 : SIMULATION RESULTS AND DISCUSSION

This section discusses and analysis the implementation results and provides guidance how to test results achieved on the implementation of the proposed scheme. In order to show the result, in this research python is used as a tool to analyze cyber threat detection from online sources based on features using Bi-directional long short time memory with Conditional random field. The simulation process is the most important part in the designing procedure and the most time consuming. Detection of cyber threat news of training and testing includes the python coding for each block in a synthesizable way and the test benches are to be written for each module and simulated using Tensor flow, and Keras simulation environment.

6.1. Experiments evaluation

In this section, the pre-processing and identifying cyber-attack features are done and the results obtained using Bi-directional long short time memory with conditional random field. Duplicate news article avoids using latent semantic analysis.

I. In this experiment, from the building dataset, different types of cyber-attack are shown below statistically. ransomware attacks are becoming increasingly sophisticated, more damaging to victims and more challenging to prevent.

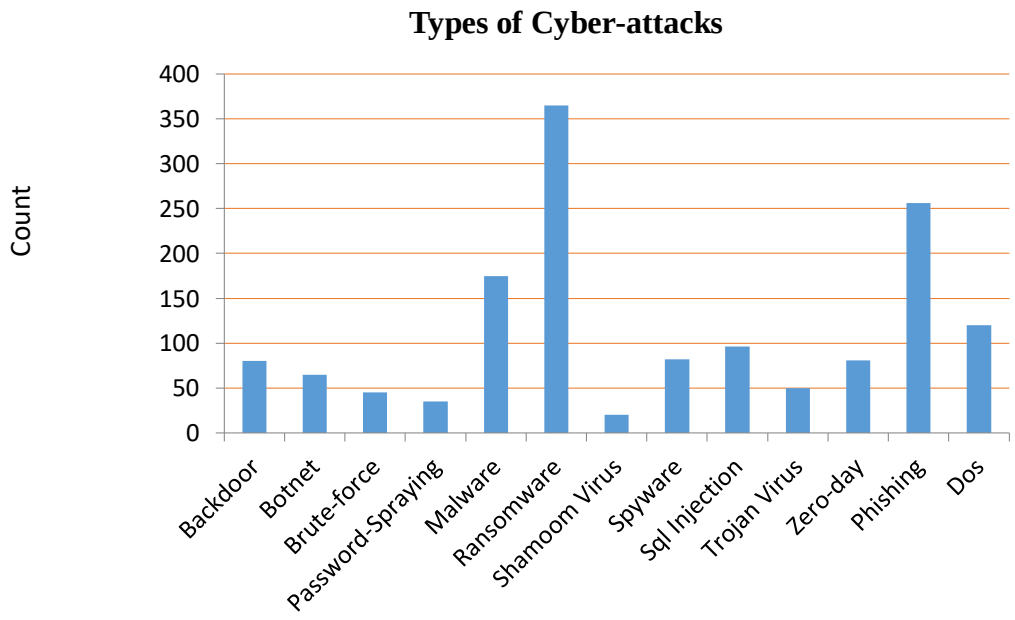


Figure 6-1 : Types of cyber-attack

I. The cyber threats occurrences and activities are increasing for every year.

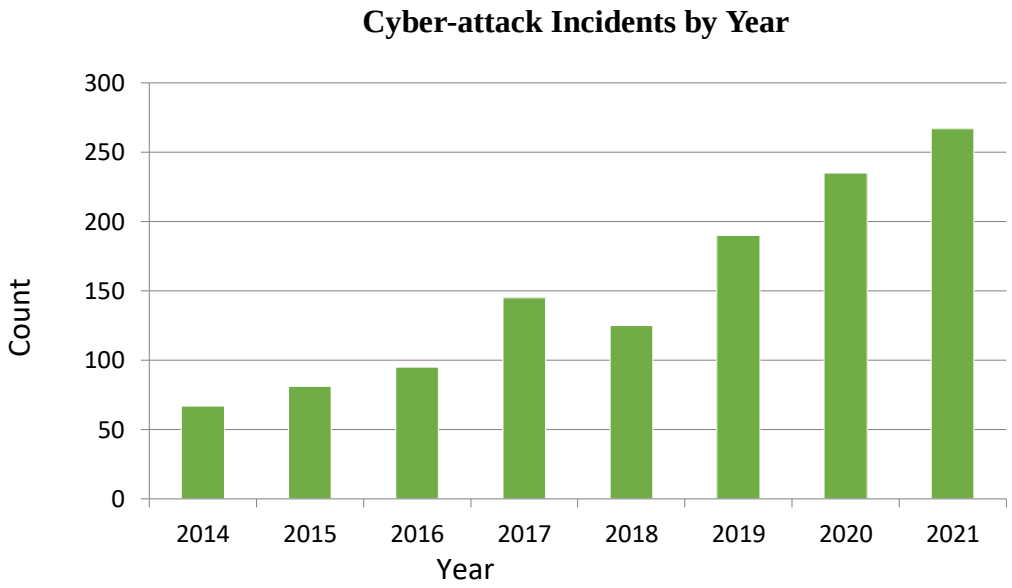


Figure 6-2 : Cyber-attack incidents by year

II. Tagged Number of words

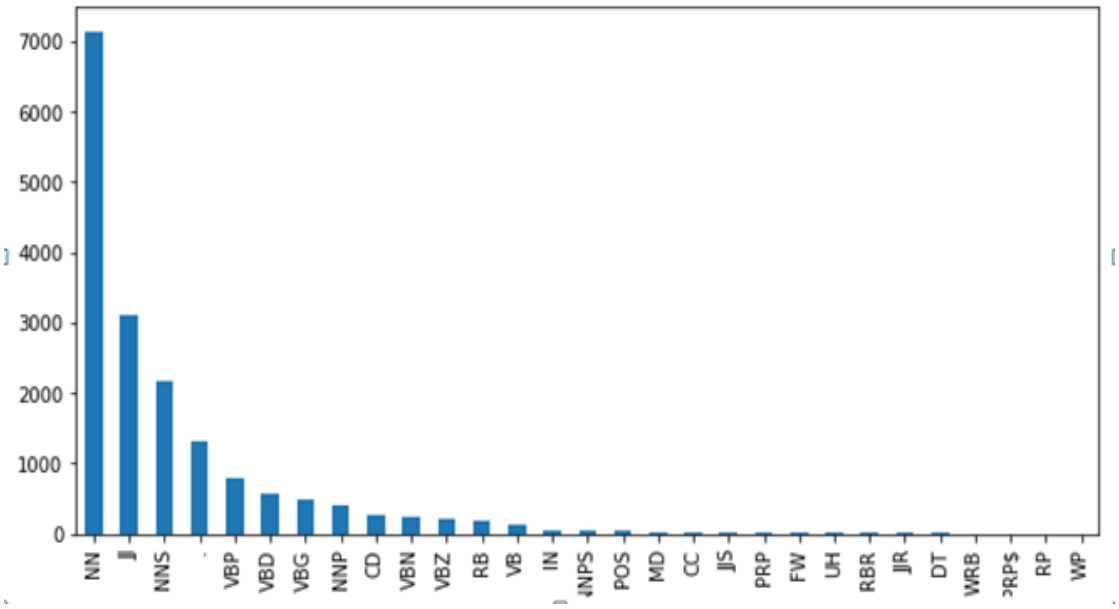


Figure 6-3 : Number of words part of speech tagging

III. Cyber-attack features from prepared dataset

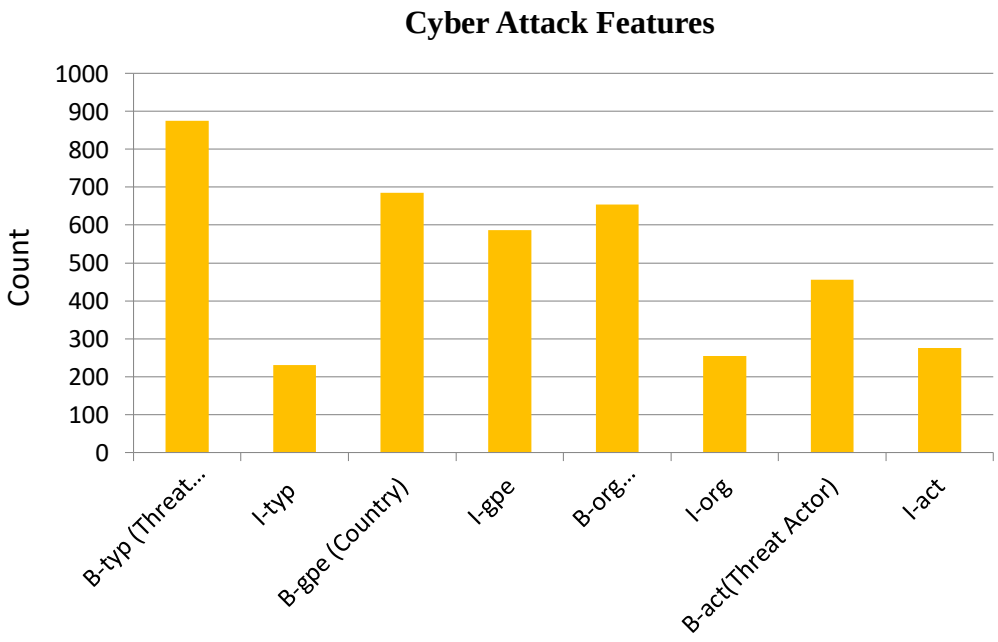


Figure 6-4 : tagged Cyber-attack features

IV. The length of words with in prepared dataset

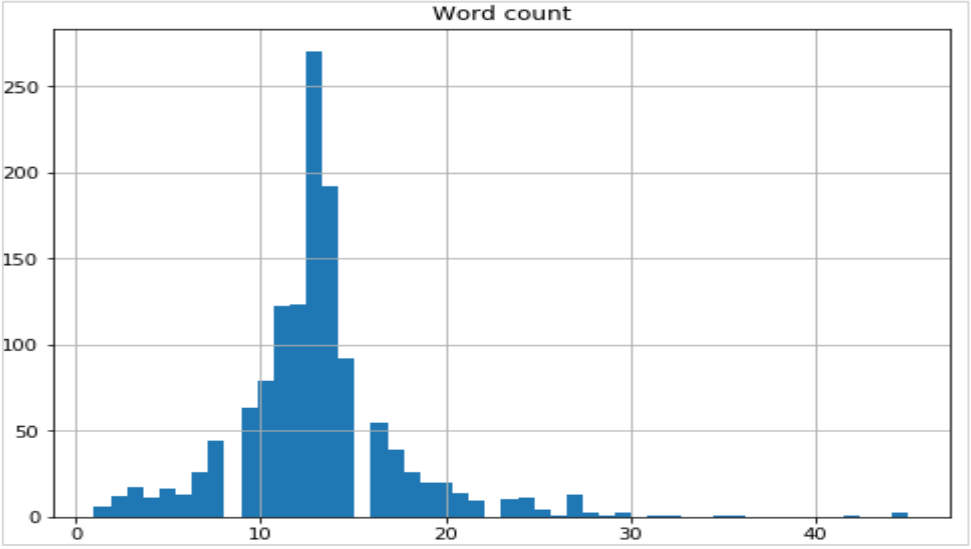


Figure 6-5 : Number of words in a sentence

6.2. Model building Results

The overall performance of Bi-LSTM-CRF classifier on Cyber threat news detection is examined. The classifier Bi-LSTM-CRF CRF is applied on the entire feature space. The experimental results are shown below with a number of iterations (epoch). The Bi-LSTM-CRF model los decreases for every iterations but the accuracy increase and reach at optimum level of 99%.

1. Bi-LSTM-CRF **model accuracy results**

Table 6-1: Bi-LSTM-CRF model training and testing results

Epoch 1/85
- 5s - loss: 2.0442 - crf_accuracy: 0.0624 - val_loss: 1.9447 - val_crf_accuracy: 0.0617
Epoch 2/85
- 2s - loss: 1.8834 - crf_accuracy: 0.3983 - val_loss: 1.7228 - val_crf_accuracy: 0.9021
Epoch 3/85
- 2s - loss: 1.5986 - crf_accuracy: 0.8778 - val_loss: 1.2665 - val_crf_accuracy: 0.8387
Epoch 4/85
Epoch 7/85
- 2s - loss: 0.3761 - crf_accuracy: 0.8356 - val_loss: 0.3202 - val_crf_accuracy: 0.8413

• • •

• • •

• • •

Epoch 80/85
- 2s - loss: 0.0343 - crf_accuracy: 0.9931 - val_loss: 0.0468 - val_crf_accuracy: 0.9914

Epoch 83/85
- 2s - loss: 0.0296 - crf_accuracy: 0.9943 - val_loss: 0.0405 - val_crf_accuracy: 0.9932
Epoch 84/85
- 2s - loss: 0.0284 - crf_accuracy: 0.9948 - val_loss: 0.0384 - val_crf_accuracy: 0.9936
Epoch 85/85
- 2s - loss: 0.0275 - crf_accuracy: 0.9948 - val_loss: 0.0365 - val_crf_accuracy: 0.9934

2. Training and testing los value

We can inspect the loss and accuracy plots from the model training. They both look acceptable and it doesn't appear that the model is over fitting. The iterations (epoch) increases, training and test value of loss decreases.

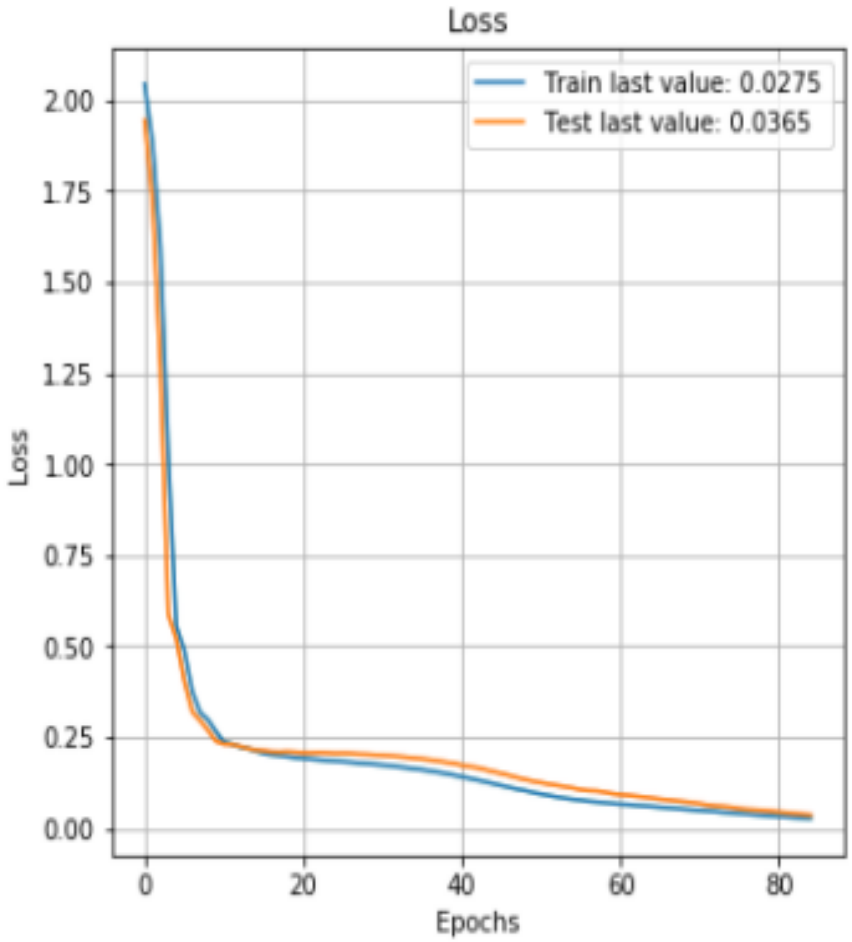


Figure 6-6 : Lose value of training and test CRF model

3. Bi-LSTM-CRF **model Accuracy**

Bi-LSTM-CRF model training and test value of accuracy increases and reaches maximum 99% at epoch 75.

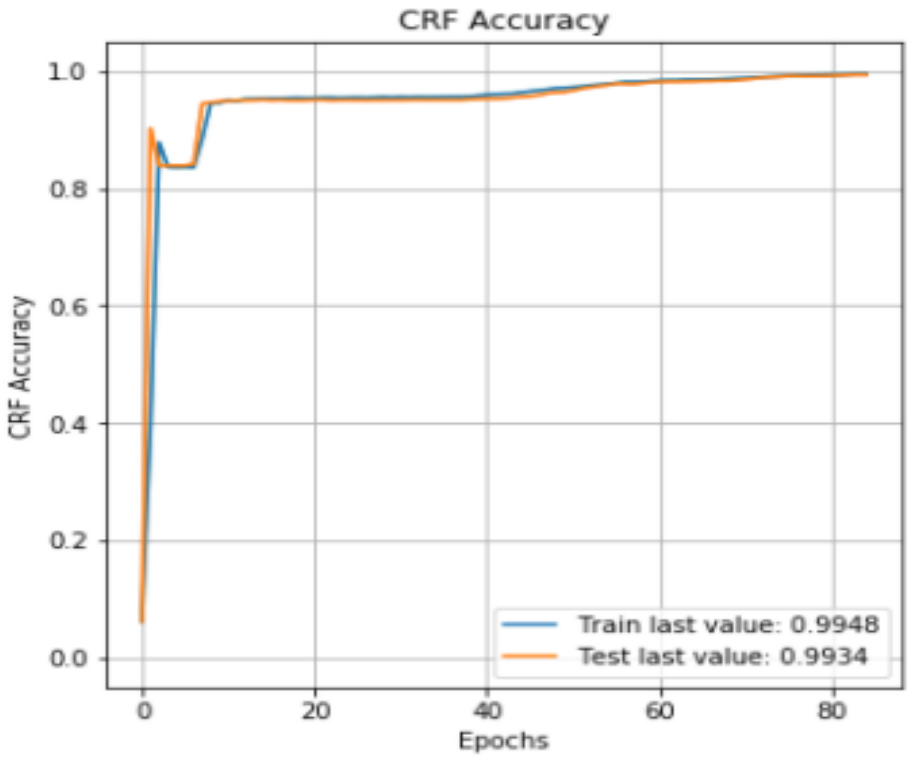


Figure 6-7 : Training and testing CRF accuracy

4. Result of Bi-LSTM-CRF **model performance based on metrics**

a) Precision score is 99% that used to measure the model performance on measuring the count of true positives in a correct manner out of all positive predictions made.

b) Recall score is 99% used to measure the model performance in terms of measuring the count of true positives in a correct manner out of all the actual positive values.

c) Accuracy score is 99% that used to measure the model performance in terms of measuring the ratio of sum of true positive and true negatives out of all the predictions made.

d) F1-score is harmonic **mean of precision and recall score** and is used as a metrics in the scenarios where choosing either of precision or recall score can result in compromise in terms of model giving high false positives and false negatives respectively.F1-score scores 98%.

Table 6-2 : CRF model performance measurement

	precision	recall	f1-score	support
B-act	1.00	1.00	1.00	17354
I-act	0.86	0.93	0.89	179
O	0.87	0.91	0.89	182
B-gpe	0.82	0.81	0.81	94
I-gpe	0.92	0.72	0.81	76
B-typ	0.99	0.96	0.97	2495
B-org	0.94	0.93	0.93	131
I-org	0.89	0.93	0.91	127
I-typ	0.89	0.80	0.85	41
accuracy			0.99	20736
macro avg	0.92	0.88	0.90	20736
weighted avg	0.99	0.99	0.99	20736

Confusion matrix:- presents a table layout of different outcomes of prediction and results of classification problem and helps visualize its outcomes. The information about the actual and predicted samples can be represented with the help of a confusion matrix. The actual values of our data set are along the column and the value predicted by our classifier are along the rows. As usual, the diagonal elements are the correctly predicted data set. Thus, the overall accuracy is 99.12%.

Table 6-3 : predicted and actual value of confusion matrix

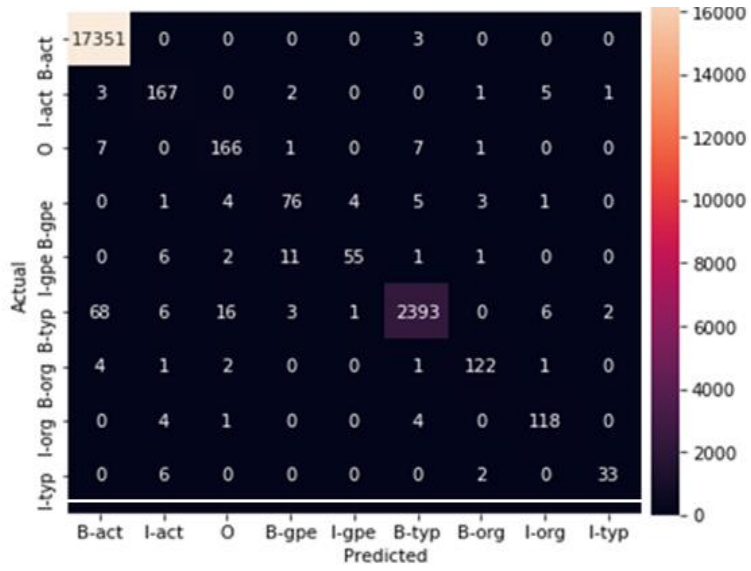


Table 6-4 : True Positive values

S.No	Tagged features	True Positive(TP)	False Positive(FP)	True Negative(TN)	False Negative(FN)
0	B-act	17351	82	3300	3
1	I-act	167	28	20529	12
2	O	166	25	20529	16
3	B-gpe	76	17	20625	18
4	I-gpe	44	5	20655	21
5	B-typ	2393	24	18217	102
6	I-typ	33	4	20691	8
7	B-org	122	8	20597	9
8	I-org	118	14	20595	9

Input text:

Text news*: "Shadow broker target health centre sustained by ransomware attack in Canada."*

Output Prediction:

shadow	**B-Act**
brocker	**I-Act**
target	**O**
health	**B-Org**
center	**I-Org**
sustained	**O**
by	**O**
ransomware	**B-Typ**
attack	**O**
in	**O**
canada	**B-Gpe**
.	**O**

Figure 6-8 : Cyber threat news detection

Finally, by constructing a sample sentence and getting predictions for the detected entities. We toke nize, and convert all words to indices. Then we call the model and print the predicted tags. From the

above sample text news, the output is “Cyber threat news”. Ransomware cyber-attacks affects in the organization of health centre as well as in the country of Canada. Next, the important idea is detectin g cyber-attack terms and keywords using CRF model. We can use this model for another research ar eas related to cyber threat name entity recognition.

Generally the system is detected cyber threat news based on cyber-attack features such us the types of attacks, hackers who influence organization in a country. As a result, the system help the people show the actual occurrences of cyber-attack activities.

5. Similarity of news

The system of cyber threat detection, duplicate articles will occur with the same meaning, so that the system also identify similar news article using Latent semantic index(LSI) by calculate degree of similarity of existing text. Sample 100 news article are calculated based on index with similarity as shown below. Index number 28 and index number 97 have similarity articles 0. *0.99979615.*

Table 6-5 : calculated similarity of news article

[(28, 0.99979615), (97, 0.99979615), (23, 0.99979293), (93, 0.99979293), (81, 0.9994453), (88, 0.9986372), (86, 0.9986069), (42, 0.9982173), (60, 0.99816966), (16, 0.99787706), (78, 0.99787706), (2, 0.9976277), (33, 0.9974921), (6, 0.9974302), (41, 0.997319), (50, 0. 99601924), (32, 0.9958291), (89, 0.9964068), (14, 0.9961767), (77, 0.9961767), (50, 0.99 601924), (32, 0.9958291), (38, 0.99533695), (20, 0.9946731), (3, 0.9919007), (31, 0.9906 614), (1, 0.98086363), (37, 0.9718709), (79, 0.965561), (92, 0.9655041), (51, 0.9593831), (49, 0.95395684), (61, 0.9500506), (11, 0.9446228), (100, 0.94357646), (52, 0.9435498), (85, 0.93325526), (36, 0.9221887), (87, 0.9204581), (73, 0.91146517), (58, 0.8962345), (18, 0.90488243), (80, 0.90488243), (95, 0.8941472), (19, 0.88976026), (90, 0.88296926), (82, 0.85733473), (29, 0.85730135), (53, 0.84619415), (10, 0.83313346), (71, 0.83313346), (63, 0.82396865), (62, 0.8177059), (64, 0.7975943), (72, 0.7919852), (83, 0.7698263), (8, 0.66799664), (69, 0.66799664), (35, 0.57929707), (7, 0.56316507),), (22, -0.1106365),

6. Bi-LSTM-CRF **model comparison with other machine learning model**

Comparative test results for different Machine learning for classification of news article, according to classifications shown below Bi-LSTM-CRF model is better results based on metrics.

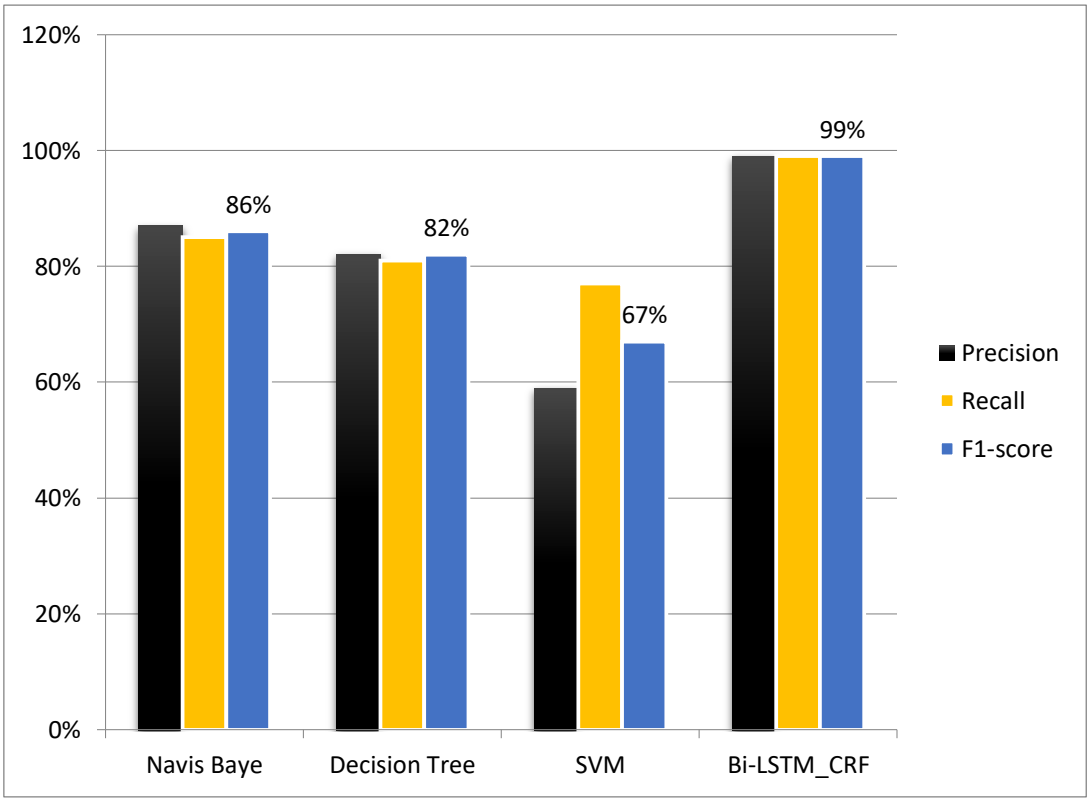


Figure 6-9 : Comparison between machine learning models for text news classification

Comparison results:- the performance of detecting cyber threat information using machine learning model such as Navies bays, Decision tree SVM and the proposed model Bi-LSTM-CRF have different results.

- **Navy's bays: -** Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. The performance of detecting cyber threat information using navy's bays model based on performance parameter is 86%.
- **Decision Tree: -** Decision Trees are great for their simplicity and interpretation, but they are more limited in their power to learn complicated rules and to scale to large data sets. The performance of detecting cyber threat information using this model is 82%.

- **Support Vector Machines:-** are often more powerful and can scale to larger data sets, but they are also more complicated and it can be difficult to know what patterns they rely on. The performance of detecting cyber threat information using this model is 67%.
- **Bi-LSTM-CRF: -** shows good performance when dealing with entity recognition (any types of entities, including named entities, time expressions, events, etc.). It can use both linguistic (characters, words) and non-linguistic information (upper/lower case, punctuation marks, spaces etc.). The achievable quality of entity recognition is about 0.80-0.99 (F1-measure), which is high. There is an interesting idea to make a hybrid algorithm by combining LSTM and CRF for entity recognition. Experimentally shows that the algorithm could increase overall quality of entity recognition by 5-10% and reach 0.92-0.99.

CHAPTER 7 : CONCLUSIONS, RECOMMENDATION, AND FUTURE WORKS

7.1 Conclusion

In this work, we presented an experimental evaluation of the performance of proposed Machine learning using Conditional Random Field (CRF) with Bidirectional Long short time memory for classification and recognition of cyber-attack name entities .The research has been able to identify the cyber-attack features from cyber threat news .The cyber-attack features are used in building the news detection scheme. The scheme is implemented with hybrid of machine learning models which are Bi-LSTM-CRF classifier and LSA for identifying ambiguity issues of news article. Bi-LSTM-CRF classifier and latent semantic analysis with addition of new cyber-attack features should able to increase the accuracy of detection of the cyber-attack news. Bi-LSTM-CRF classifier achieves the best results so far. Experimental results demonstrate that using Bi-LSTM-CRF classifier is an effective way of classification performance. The model achieves an overall F-measure of 98.48% for Cyber threat news detection with accuracy 99.12%. The detection scheme is helping a better detection of cyber-attacks news and categories the threats accordingly. The output will eventually provide the full picture of cyber-attacks that take place in our cyber environment. As a result, it can assist cyber security organizations in producing reports about cyber-attack activities or offering insight into any decision-making process. Finally, this study aims to raise public awareness by displaying the numbers and figures of recent cyber-threat incidents, so that people will be more aware and able to avoid getting victims. Build cyber threat name(keyword) entity recognition model in order to help researchers for extracting cyber threat keywords.

7.2. Recommendation

There are large volumes of data from online-generated cyber threat news that are freely available and might contain valuable information. Cyber threat information is highly increasing and can be analysed to gather informative insights into the current situation. This study identifies cyber-threat information from online news using machine learning. The system starts with identifying cyber-attack features that are used to classify cyber threat information using the Bi-directional Long Short

Time Memory with Conditional Random Field model, as well as categorising similar news articles using Latent Semantic Analysis to eliminate ambiguous cyber threat news.

The study made several recommendations based on the conclusions drawn from its findings. We have discussed and concluded that we cannot recommend a particular learning technique for every cyber threat detection. Different learning models are being used for specific cyber threats. On the other hand, there are a vast number of authors who have worked to highlight the constraints faced by machine learning techniques. We have observed and suggested that there is a dire need for the latest benchmark dataset to test the latest advancements in the field of machine learning for cyber threat detection. Available datasets lack diversity and sophistication and contain missing values. There is a need for specific and customised learning models specifically designed for security purposes.

7.3. Future Work

For future study, we can employ more diverse categories and cyber-attack features that would be difficult for a machine to categorize, as well as compare more advanced classifiers. The algorithm to be applied is determined by the target application.

REFERENCE

Abdullah, M. S., Zainal, A., Maarof, M. A., & Kassim, M. N. (2018). *Cyber-attack features for detecting cyber threat incidents from online news.* Paper presented at the 2018 Cyber Resilience Conference (CRC).

Aggarwal, N., Asooja, K., & Buitelaar, P. (2012). *DERI&UPM: Pushing corpus based relatedness to similarity: Shared task system description.* Paper presented at the * SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012).

Arpitha, B., Sharan, R., Brunda, B., Indrakumar, D., & Ramesh, B. (2021). Cyber Attack Detection and notifying system using ML Techniques. *International Journal of Engineering Science and Computing (IJESC)*.

Arulanandam, R., Savarimuthu, B. T. R., & Purvis, M. A. (2014). *Extracting crime information from online newspaper articles.* Paper presented at the Proceedings of the second australasian web conference-volume 155.

Bissell, K., Lasalle, R. M., & Cin, P. J. P. I. D., Ireland. (2019). Ninth annual cost of cybercrime study. *6*.

Boling, C., & Das, K. (2014). Semantic similarity of documents using latent semantic analysis. *2014 NCUR*.

Bonaccorso, G. (2017). *Machine learning algorithms*: Packt Publishing Ltd.

Brown, D. E., & Liu, X. (2016). *Extracting Addresses from News Reports Using Conditional Random Fields.* Paper presented at the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA).

Carey, H. J., & Manic, M. (2016). *HTML web content extraction using paragraph tags.* Paper presented at the 2016 IEEE 25th International Symposium on Industrial Electronics (ISIE).

Fang, Y., Gao, J., Liu, Z., & Huang, C. (2020). Detecting cyber threat event from twitter using IDCNN and BILSTM. *Applied Sciences, 10*(17), 5922.

Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. *international journal of Computer Applications, 68*(13), 13-18.

Iorga, D., Corlătescu, D., Grigorescu, O., Săndescu, C., Dascălu, M., & Rughiniş, R. (2020). *Early detection of vulnerabilities from news websites using machine learning models.* Paper presented at the 2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet).

Kao, A., Poteet, S., Wu, J., Ferng, W., Tjoelker, R., & Quach, L. (2012). Latent semantic analysis for text mining and beyond. In *Intelligent multimedia databases and information retrieval: Advancing applications and technologies* (pp. 253-280): IGI Global.

Kharaz, A., Arshad, S., Mulliner, C., Robertson, W., & Kirda, E. (2016). *{UNVEIL}: A large-scale, automated approach to detecting ransomware.* Paper presented at the 25th {USENIX} Security Symposium ({USENIX} Security 16).

Landauer, T. K., McNamara, D. S., Dennis, S., & Kintsch, W. (2013). *Handbook of latent semantic analysis*: Psychology Press.

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes, 30*(1), 3-26.

Park, J.-H., & Kwon, H.-Y. (2022). Cyberattack detection model using community detection and text analysis on social media. *ICT Express, 8*(4), 499-506.

Ponemon, L. J. P. I. (2017). Cost of data breach study.

Rollo, F., Po, L., & Bonisoli, G. (2022). *Online news event extraction for crime analysis.* Paper presented at the Proceedings of the 30th Italian Symposium on Advanced Database Systems, SEBD.

Rozeva, A., & Zerkova, S. (2017). *Assessing semantic similarity of texts–methods and algorithms.* Paper presented at the AIP Conference Proceedings.

Schiappa, M., Chantry, G., & Garibay, I. (2019). *Cyber security in a complex community: A social media analysis on common vulnerabilities and exposures.* Paper presented at the 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS).

Shabat, H., Omar, N., & Rahem, K. (2014). *Named entity recognition in crime using machine learning approach.* Paper presented at the Asia Information Retrieval Symposium.

Tsai, F. S., & Chan, K. L. (2007). *Detecting cyber security threats in weblogs using probabilistic models.* Paper presented at the Pacific-Asia Workshop on Intelligence and Security Informatics.

Veldkamp, Q. H. B. P. (2012). Classifying unstructured textual data using the Product Score Model: An alternative text mining algorithm. *Psychometrics in Practice at RCEC*, 47.

Vijaymeena, M., & Kavitha, K. (2016). A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal, 3*(2), 19-28.

Ye, N., Harish, B., & Farley, T. (2005). Attack profiles to derive data observations, features, and characteristics of cyber attacks. *Information Knowledge Systems Management, 5*(1), 23-47.

Appendixes

Appendix A: Text Pre-processing

```
# Load packages
import pandas as pd
import numpy as np
import nltk

# load Dataset
data = pd.read_csv('Data_with_Feature_kifle.csv', encoding = "ISO-8859-1")
data.head()

# Types of attack occurences visualizations
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
plt.style.use('dark_background')
plt.figure(figsize=(20,12))
edgecolor=(0,0,0),
sns.countplot(df['Threat_Type'].sort_values(), palette = "Dark2", edgecolor=(0,0,0))
plt.title("Types of attack occurences",fontsize=20)
plt.xlabel('Threat_Type')
plt.ylabel('Number of Threats')
plt.xticks(fontsize=16,rotation=80)
plt.show()

# Cyber threat incidents group by year
plt.figure(figsize=(20,12))
edgecolor=(0,0,0),
sns.countplot(df['Incident_year'].sort_values(), palette = "Dark2", edgecolor=(0,0,0))
plt.title("Cyber threat incidents group by year",fontsize=20)
plt.xlabel('Year')
plt.ylabel('Number of Threats')
plt.xticks(fontsize=12,rotation=75)
plt.show()

#Remove stopwords
#install NetText
#Load Text Cleaning PKgs
import neattext as nt
import neattext.functions as nfx
#noise scan
df['Text'].apply(lambda x: nt.TextFrame(x).noise_scan()['text_noise'])
#remove custom Pattern
```

```
df['Text'].apply(lambda x: nfx.remove_custom_pattern(x,term_pattern=r'&#\$+'))
df['clean_txt']=df['clean_txt'].apply(nfx.remove_multiple_spaces)

#extract stopwords
df['clean_txt'].apply(lambda x: nt.TextExtractor(x).extract_stopwords())

#remove stopwords using TextFrame
df['clean_txt'].apply(lambda x: nt.TextFrame(x).remove_stopwords())
df['msg_lower']= df['clean_txt'].apply(lambda x: x.lower())
df.head()
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
def porter_stemmer(text):
        # word tokenization
        tokens = word_tokenize(text)

        for index in range(len(tokens)):
                # stem word to each word
                stem_word = stemmer.stem(tokens[index])
                # update tokens list with stem word
                tokens[index] = stem_word

        # join list with space separator as string
        return ' '.join(tokens)

# initialize porter stemmer object
stemmer = PorterStemmer()
# example text for stemming technique
```

Appendix B: Bi-LSTM-CRF model implementation code

```
import pickle

import operator
import re
import string
import pandas as pd
import numpy as np
from tensorflow import keras
import matplotlib.pyplot as plt
from plot_keras_history import plot_history
from sklearn.model_selection import train_test_split
from sklearn.metrics import multilabel_confusion_matrix,confusion_matrix
from keras_contrib.utils import save_load_utils
from sklearn_crfsuite.metrics import flat_classification_report,flat_f1_score
from keras import layers
import seaborn as sns
from keras import optimizers
from keras.models import Model
from keras.models import Input
from keras_contrib.layers import CRF
from keras_contrib import losses
from keras_contrib import metrics
import tensorflow as tf
from mlxtend.plotting import plot_confusion_matrix
from sklearn.preprocessing import OneHotEncoder
```

```
# load Annotated Dataset
data_df = pd.read_csv("textLastRecoverd.csv", encoding="iso-8859-1", header=0)
data_df.head()
```

```
#training and testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1234)
print("Number of sentences in the training dataset: {}".format(len(X_train)))
print("Number of sentences in the test dataset    : {}".format(len(X_test)))
```

```
#this makes feeding the data to the model simpler.
X_train = np.array(X_train)
X_test = np.array(X_test)
y_train = np.array(y_train)
y_test = np.array(y_test)
```

```
#
from keras import layers
input_layer = layers.Input(shape=(MAX_SENTENCE,))
```

```
model = layers.Embedding(WORD_COUNT, DENSE_EMBEDDING,
        embeddings_initializer="uniform", input_length=MAX_SENTENCE)(input_layer)
model = layers.Bidirectional(layers.LSTM(LSTM_UNITS,
        recurrent_dropout=LSTM_DROPOUT, return_sequences=True))(model)
model = layers.TimeDistributed(layers.Dense(DENSE_UNITS, activation="relu"))(model)
crf_layer = CRF(units=TAG_COUNT)
output_layer = crf_layer(model)
ner_model = Model(input_layer, output_layer)
loss = losses.crf_loss
acc_metric = metrics.crf_accuracy
opt = optimizers.Adam(lr=0.001)
ner_model.compile(optimizer=opt, loss=loss, metrics=[acc_metric])
ner_model.summary()
#plot the model
history = ner_model.fit(X_train, y_train, batch_size=BATCH_SIZE,
        epochs=MAX_EPOCHS, validation_split=0.1, verbose=2)
plot_history(history.history)
```

Appendix C: LSA implementation code

```
from gensim import corpora, models, similarities

print(documents)
texts = [[word for word in document.lower().split() if word not in stoplist] for document in
        documents]
tfidf = models.TfidfModel(corpus) # step 1 -- initialize a model
print(tfidf)
lsi = models.LsiModel.load('model.lsi') # try to load above saved model
print(lsi)
index = similarities.MatrixSimilarity(lsi[corpus]) # transform corpus to LSI space and
        indexize it
print(index)
sims = index[vec_lsi] # calculate degree of similarity of the query to existing corpus
print(sims)
sims = sorted(enumerate(sims), key=lambda item: -item[1])  # sort output object as per
        similarity ( largest similarity document comes first )
print(sims)
```