**CYBER THREAT DETECTION IN OPEN-SOURCE DATA USING SELECTED MACHINE LEARNING ALGORITHMS**

**BY  
AJAYI MARY ODUNAYO  
(MATRIC. NO – 20PGCD000153)**

**A MASTER OF SCIENCE DISSERTATION SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE, COLLEGE OF PURE AND APPLIED SCIENCES, LANDMARK UNIVERSITY, OMU-ARAN, NIGERIA.**

**NOVEMBER 2022**

# DECLARATION

I, Mary, Odunayo AJAYI a **Master’s in Science** student in the **Computer Science Department**, Landmark University, Omu-Aran, with matriculation number **20PGCD000153** hereby declare that this dissertation entitled **Cyber Threat Detection in Open-Source Data Using Selected Machine Learning Algorithms** submitted by me is based on my original work. Any material(s) obtained from other sources or work done by any other persons or institutions have been duly acknowledged.

------------------------------------------------------------

Student’s Full Name and Matriculation Number

------------------------

Signature & Date

# CERTIFICATION

This is to certify that this dissertation has been read and approved as meeting the requirements of the Department of Computer Science, Landmark University, Omu-Aran, Nigeria, for the award of M.Sc. Degree.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Professor A. A. Adebiyi Date

(Supervisor)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. M. O. Adebiyi Date

(Co-Supervisor)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. M. O. Adebiyi Date

(H. O. D)

# DEDICATION

This project is dedicated to the Lord Almighty who made this program a success, also to my mother Grace Ajayi, Engr. S. O. Aniyi, Dr. J.A. Aniyi, and my siblings for their support and utmost care for me.

# ACKNOWLEDGEMENTS

My profound gratitude goes to the Almighty God the repository of wisdom, for divine guidance, provision and inspiration all through this program.

I appreciate my supervisor Professor A. A. Adebiyi and my co-supervisor Dr. M. O. Adebiyi for their time, tireless effort and guidance for the success of this project.

I want to acknowledge all the lecturers and staff of the Department of Computer Science and the School of Postgraduate Studies, Landmark University.

Lastly, I also acknowledge my parents and family members for the love and encouragement during this program. To my siblings and friends Fadekemi, Adeoluwa, Esther, Miracle, Toluwase, Precious, Paul, I thank them for their unwavering support. Also, to my course mates Udo, Suraj and Nehemiah, I thank them for the memories we shared.

# TABLE OF CONTENTS

Title Page………………………………………………………………………………i

Declaration...…...……………………………………………………………………...ii

Certification...………...………………………………………………………………iii

Dedication...…...……………………………………………………………………...iv

Acknowledgements……………………………………………………………………v

Table of content.……………………………………………………………………....vi

List of Tables…………………………………………………………………...……. ix

List of Figures…………………………………………………………………………x

Abstract.……………………………………………………………………...……….xi

**CHAPTER ONE**

**1.0 INTRODUCTION**

* 1. Background to the study…...………………………………………………………1
  2. Statement of the Problem………………………………………………………….2
  3. Justification for the Dissertation……….……….………………………………....2
  4. Aim and Objectives of the Study…………………………………………......…...3
  5. Research Methodology…………………………………………………………....3
  6. Scope of the Study……………………………………………………………...…4
  7. Significance of the Study………………………………………………………….4
  8. Organization of the Thesis……...…...…………………………………………….4

**CHAPTER TWO**

**2.0 LITERATURE REVIEW**

2.1 Conceptual Issues…………………………………………………………………6

2.2 Cyber Threat Intelligence…………………………………………………………7

2.2.1 Information versus Intelligence………………………………………….7

2.2.2 Sources of Cyber threat intelligence………………….………………...9

2.3 Open-Source Intelligence………………………………………………………....9

2.3.1 Origin of Open-Source Intelligence…………...……………………….10

2.3.2 Application of Open-Source Intelligence………………………………11

2.3.3 Open-Source Intelligence tools…………………………………….…..11

2.4 Machine Learning Approaches to Cyber Threat Intelligence…………………....12

2.4.1 Machine Learning Paradigms.…………………………………….…....12

2.5 Survey of Methodologies………………………………….………………….….13

2.6 Gaps Identified in the Literature…………………………………………………18

**CHAPTER THREE**

**3.0 METHODOLOGY**

3.1 Research Design………………………………………………………………….19

3.2 Research Design Layout………………………………………………………….19

3.2.1 Dataset………………………………………………………………….20

3.2.2 Data Preprocessing.…………………………………………………….21

3.2.3 Classification Models………………………………………….……….22

3.2.4 Performance Evaluation……………………………………….……….30

3.3 Research Instrument/Tools……………………………………………………….31

3.4 Data Collection………………………………………………………………..….32

3.5 Data Analysis Algorithm…………………………………………………………32

**CHAPTER FOUR**

**4.0 RESULTS AND DISCUSSION**

4.1 Machine Learning Models...………………………...……………………………34

4.1.1 Logistic Regression.……………………………………………………34

4.1.2 Random Forest…………………………………………………………35

4.1.3 Decision Tree.….…………………….…………………………...……35

4.1.4 Naive Bayes ……………………………………………………………36

4.2 Result Analysis…………………………………………………………………...36

4.3 Discussion………………………………………………….…………………….38

**CHAPTER FIVE**

**5.0 SUMMARY, CONCLUSION AND RECOMMENDATION**

5.1 Summary…………………………………………………………………………39

5.2 Conclusion…….….………………………………………………………………40

5.3 Recommendation....………………………………………………………………40

5.4 Future Works.………………………………………….…………………………40

References……………………………………………………………………………42

# LIST OF TABLES

Table 2.1: Sources of CTI…………………………………………………….….…...9

Table 4.1: Performance Score Comparison………………………………………….37

# LIST OF FIGURES

Figure 2.1 The Relationship between Data, Information and Intelligence……………8

Figure 3.1 The Conceptual Framework of the Thesis……………………………….19

Figure 3.2 The Dataset used for the Thesis.….………………………………………20

Figure 3.3 Preprocessed Dataset.…………………….………………………………22

Figure 3.4 Pseudo code for Naive Bayes…………………………………………….25

Figure 3.5 A Decision Tree…………….…………………………………………….27

Figure 4.1 Performance Scores for Logistic Regression….…………………….……34

Figure 4.2 Performance Scores for Random Forest ....………………………………35

Figure 4.3 Performance Scores for Decision Tree ....…………………….….………36

Figure 4.4 Performance Scores for Naive Bayes....……………………..….………..36

Figure 4.5 Chart Presenting the Performance Metrics for the four Algorithms……...38

# ABSTRACT

Threat actors are developing and evolving their tools to quickly sight loopholes and vulnerabilities in devices and the security of organizations. Open sources are frequently used by these malicious threat actors to exchange their Tactics, Techniques, and Procedures (TTP) to attack devices. There is a huge amount of threat data available on these open sources making it difficult for cybersecurity professionals to utilize and share. Humans can easily differentiate the useful and relevant information, but it is daunting when the data is large with limited time hence the need to automate the process.

This thesis presents a comparative analysis on the performance of four machine learning algorithms (Decision Tree, Logistic Regression, Random Forest and Naïve Bayes) to help cybersecurity professionals in making decision on the most suitable algorithm to analyze cyber threat intelligence dataset. The dataset used in this study is a Cyber Threat data generated by (Kim & Kim, 2019) and was automatically obtained from reports on freely available platforms and malware repository databases. The dataset is in an extensible markup language (XML) format which entails roughly 640,000 records gathered from various security reports produced between January 2008 and June 2019. 70% of the total of the dataset was used for training, with the rest 30% used as the testing dataset and to construct the machine learning model.

Experimental results show that Random Forest algorithm has the best performance with an accuracy score of 97.16% followed by Decision Tree with an accuracy of 97.08%, Naïve Bayes also has an accuracy of 93.92% while Logistic Regression classifier has the least score of all the four algorithms with the accuracy of 80.15%.

The Other evaluation metrics used for the comparative analysis in this study are F1 score, recall and precision of the algorithms. Precision for Logistic Regression is 72.51%, Naïve Bayes is 75.17%, Decision Tree is 95.49% and Random Forest is 95.6%. Also, for the recall, Logistic regression is 71.11%, Naïve Bayes is 83.13%, Decision tree is 95.06% and Random Forest is 95.07%. And lastly, the F1 score for Logistic Regression is 67.27%, Naïve Bayes is 78.04%, Decision Tree is 95.11% and Random Forest is 95.13%.

Logistic regression which had the least scores in all the four metrics compared to the other three algorithms. These means that the algorithm is not best suited for the dataset used in this thesis. Future work can investigate how to improve the performance of the algorithm. Prospective researchers can learn from the findings of this work in order to come up with newer and enhanced algorithms, which can be useful in decision making for cyber security experts.

# CHAPTER ONE

# INTRODUCTION

* 1. **Background to the study**

In the last few years, research has shown a substantial hike in the extent and heterogeneity of cyberattacks, as a result of this, forensic experts and security professionals struggle to identify, analyze and counteract cyber threats concurrently (Conti et al., 2018 & Lancaster, 2020). Cyber Threat Intelligence (CTI) was designed to help security analysts to spot indicators of compromise, extract threat vectors accurately and swiftly respond to the attacks (Conti et al., 2018).

CTI is a crucial part of cyber security that ensures both proactive and reactive security in order to limit the time frame between compromise and detection if previous security plans fail (Tounsi & Rais, 2018). Although the benefits of CTI are widely recognized in the industry, putting them into practice can be a tough and inefficient undertaking. Threat information is often overwhelming, and it has not always followed established forms, making it difficult to utilize and share, which can lead to organizations avoiding its use. Threat intelligence can become a realistic tool for cyber threat prevention by utilizing successful technology like machine learning (Baker, 2022).

Machine Learning (ML) has been evolving as a discipline and is now widely employed practically in almost every field of endeavor as it has substantial achievements in various subfields such as intrusion detection, predictions, sentiment analysis, image processing, spam filter, wide societal impact, signal processing, speech recognition, medical diagnosis based on image recognition, navigation for autonomous vehicles and various areas where digitally stored information is used to make choices and improve productivity (Fastovets, 2019).

* 1. **Statement of the Problem**

There are several online platforms from which cyber security experts can get data about security threats and attacks. This huge amount of threat data available on various platforms include relevant and useless data. It is easy for humans to distinguish between useful data from the irrelevant ones. However, this becomes more difficult because of the amount of data and the need for speed (Deliu et al., 2017).

Existing literature have used different sources of data as well as algorithms. These previous works have shown that there is need to get threat data from other sources, try other machine learning algorithm and improve on the traditional capabilities of the existing solutions. It is in this light that this study used four algorithms for machine learning (Decision Tree, Logic Regression, Random Forest and Naïve Bayes) to analyze threat data after performing dimensionality reduction through feature selection.

Naïve Bayes Classifier was chosen because of its performance flexibility, be easily trained on small dataset as well as large datasets and it predicts the class of text data quickly and easily. Decision Trees algorithm was chosen because it does not need so much data preprocessing and is not affected by missing values. Random Forest is widely embraced algorithms for feature selection. It has a built-in library that selects a smaller subset of features. This function can be called no need for extra coding.

* 1. **Justification for the study**

The increase in the use of smartphones and computing devices has led to an increase in internet users, several of the devices are vulnerable including organizations. Attackers communicate and share some vulnerabilities and new trends among themselves by using some online platforms and forums (Deliu et al., 2017).

Cyber security professionals have found ways to also get access into these forums in order to mitigate their threats but the data available to them is huge, the data contain both relevant and irrelevant data. There is need to automate the extraction of threat data from these platforms and forums in order to make use of them for quick and efficient decision making as well as getting actionable intelligence from these information sources.

In this thesis, four machine learning algorithms are used to get cyber threat intelligence from open-source data. The outcome of this study is intended to assist cyber security experts to identify, resolve and protect against cyber threat synchronously through cyber threat intelligence.

* 1. **Aim and Objectives of the Study**

The aim of this study is to detect cyber threat in open-source data. The specific objectives of the study are to:

1. identify the features in the existing CTI models;
2. design CTI model using machine learning algorithms (Decision Tree, Logic Regression, Random Forest and Naïve Bayes);
3. implement the models as proof of concept; and
4. evaluate the performances of the classifiers using performance metrics (accuracy, precision, recall and F1 score)
   1. **Research Methodology**

The dataset is used to train and create a model using Naive Bayes, Logistic Regression, Decision Tree and Random Forest classification algorithm.

The dataset used in this study is a Cyber Threat data generated by (Kim & Kim, 2019) and was automatically obtained from reports on freely available platforms and malware repository databases. The dataset is in an extensible markup language (XML) format which entails roughly 640,000 records gathered from various security reports produced between January 2008 and June 2019. The dataset includes attributes such as id, date, info, category, value type and id2. The dataset is in comma separated values (csv) format. 70% of the data was used for training and 30% for testing.

* 1. **Scope of the study**

This study is restricted to only four machine learning algorithms. (Decision Tree, Logistic Regression, Random Forest and Naïve Bayes).

* 1. **Significance of the Study**

Threat actors are getting smarter as they also leverage on machine learning, AI and new technologies as they advance and cyber security professionals must be at alert and updated in order to proactively counter adversaries. This research will help to reduce the time cyber security experts spend to properly classify cyber threat data.

Findings from this study are also important for future researchers that may want to leverage machine learning for cyber threat intelligence in making decision on their choice of algorithm.

* 1. **Organization of the Thesis**

This thesis is divided into five chapters. The remaining part of the dissertation is as follows: Chapter Two covers the review of literature consisting of existing research on Cyber Threat Intelligence, Open-Source data and Machine learning algorithms. Chapter Three covers the research methodology, the design of the algorithms, dataset obtained. Chapter Four presents the result. The fifth chapter which is the last, presents the summary, conclusion, and recommendations.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 Conceptual Issues**

Ponemon Institute (2016) estimated the cost of one data breach in an organization to be $4 million of which 48% occur because of insiders and hackers. These threat actors are also developing and evolving their tools to gather private data making them more proficient in quickly sighting loopholes as well as vulnerabilities in companies’ security posture (EY, 2014). Due to the Bring Your Own Device (BYOD) policy that companies are adopting, hackers exploit this policy to infect organizations through mobile malware. The statistics in 2015 shows 1.4 billion smartphones to be in use giving hackers a wider coverage, and in that same year 430 million malware constructions were found (Symantec, 2016).

Criminals are increasingly targeting mobile devices since important personal information is kept on them, and services like Android Pay are becoming more widespread. Cyber threat intelligence is being used by businesses to gain insight into malware and threat ecosystems in order to prevent threats. Current CTI methods, on the other hand, look at malware in cyberspace in a reactive manner, focused on analyzing former attacks. Bromiley (2016) stated that “CTI cannot be dated information that fails to help an organization protect itself or better understand their threats, such as their attackers and their related techniques.”

Online hackers’ forum among other open sources are frequently exploited by malicious threat actors to discuss their Tactics, Techniques, and Procedures (TTP) to attack devices. Hackers, for instance may send mobile malware updates as forum attachments. Identifying and researching such posts, as well as the threat actors that create them, can help to create a new and proactive kind of CTI. As a result, open sources can be a valuable resource for malware and threat actors (Grisham et al., 2017). There are several platforms and open sources.

Related work for this thesis falls into three main categories: Cyber threat intelligence, open-source intelligence (OSINT), and machine learning

**2.2 Cyber Threat Intelligence (CTI)**

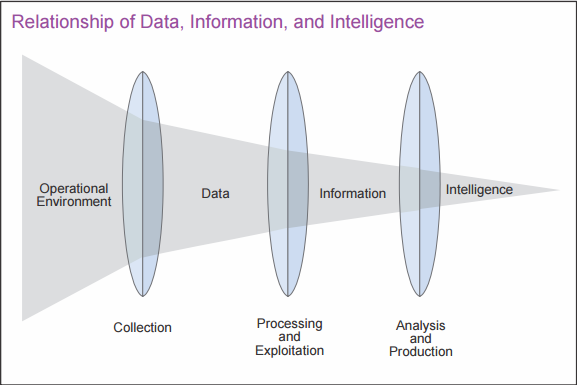
Cyber threat intelligence is a method that helps to comprehend the target of attackers and ultimately predict threat by gathering valid information (Kim and Kim, 2019). Cyber threat intelligence arrangements built on open-source data are one way to increase security operations center (SOC) performance. These algorithms sift through a vast number of posts from a variety of sources in order to select pertinent threat information. The data is then processing to provide threat intelligence in the form of threat information with operable perception. This cyber threat information can provide businesses with insights into the threat landscape, allowing them to take a more proactive overall security of their company (Shackleford, 2017).

Cyber Threat Intelligence is prominently identified to be an efficiently proactive and resilient procedures owing to its ability to systematically predict potential threats. CTI gathers and analyzes information from a range of sites, including amateurs, specialists, and then detects associated risks according to their prior findings (Hossein et al, 2021).

Asides cyber security experts and analysts, all consumers of technology products, security operations, risk assessment, vulnerability management, fraud prevention, and strategic implementation among others can also benefit from CTI (Webroot, 2022).

**2.2.1 Information versus Intelligence**

Figure 2.1 compares data, information and intelligence and how useful insights can be deduced from it eventually. The end goal of the life cycle of cyber security is intelligence which can be acted upon in improving the overall security posture of an organization. Hence organizations need to be proactive in developing and engaging tools that can provide actionable threat intelligence as it provides sufficient information for decision making (Abu et al. 2018).



**Figure 2.1 The Relationship between Data, Information and Intelligence**

**Source: (US Joint Chiefs of Staff, 2013)**

Data are made up of raw, unprocessed, and unfiltered information, which are commonly represented by symbols and signal readings. Words (whether vocal or written), numbers, graphs, and images or video are all examples of symbols. Data which have been analyzed, aggregated, and organized into a more human-friendly format that provides more breadth and can be used for evaluation are referred to as information (Liew, 2007).

From a professional standpoint, intelligence is data that was cleaned, analyzed, and refined, with the result being implementable, and significant. Those three outcomes can be achieved through human-driven strategic and critical procedures that provide contextual data and useful output (Dalziel, 2014). When it comes to information security, intelligence is defined as credible information or the outcome of the intelligence maturity model (Brown et al., 2015). Intelligence is mainly used for decision making and operationally to detect, prevent and respond to threat (Dalziel, 2014).

**2.2.2 Sources of Cyber threat intelligence**

The intelligence domains, acronyms, and definitions are briefly described in Table 2.1. For cyber threat intelligence, each of the intelligence disciplines listed in Table 2.1 can work with or be part of OSINT.

**Table 2.1: Sources of CTI (Noel, 2021)**

|  |  |  |
| --- | --- | --- |
| **Discipline** | **Abbreviation** | **Description** |
| Human | HUMINT | from an individual |
| Geospatial | GEOINT | satellite and aerial photography, as well as mapping and topography data |
| Measurement and Signature | MASINT | based on a variety of distinguishing traits It's possible to subdivide it into distinct disciplines. |
| Signals | SIGINT | from signal interception |
| Technical | TECHINT | from analysis of weapons and equipment |
| Open-Source | OSINT | derived from open sources |

**2.3 Open-source Intelligence (OSINT)**

Open-source intelligence is the investigation and collection of data, tools, and methodologies from publicly available channels like social media, government reports, geolocations, code sites, social networks, satellite images, scholarly journals, vulnerability databases, the internet, and other open media features. It is publicly available data that are gathered, evaluated, and released to the proper audience in a reasonable timeframe to fulfill a particular intelligence need (Richelson, 2016).

**2.3.1 Origin of OSINT**

Since 1941, when the Foreign Broadcast Monitoring Service (later known as the Open-Source Center) was established to watch German and Japanese radio transmissions, the procedure of data and information acquisition has been a point of controversy (Evangelista et al., 2020). Even though the Open-Source Center was founded in 1941, OSINT was officially defined in 2001 when the North Atlantic Treaty Organization (NATO) published "The Open-Source Intelligence Handbook". OSINT is defined as quasi-information that has been intentionally discovered, disaggregated, refined and released to a predetermined audience to solve a particular problem, according to the publication (NATO, 2001).

The US Department of the Army (2012) broke OSINT into two namely “open source” and “publicly available”. Any people, business, or system that provides information without concern for privacy is considered open source. These sources provide data that are not protected from public disclosure. Even though this information is publicly available, it is not necessarily information that should be made public. Data, facts, instructions, or publications given to the broad public for consumption are known as publicly available information and can be legally accessed by anybody.

**2.3.2 Application of OSINT**

Considering OSINT is an immediate and appropriate method of doing in-depth security evaluations, government organizations such as the Federal Bureau of Investigation (FBI), the Central Intelligence Agency (CIA), and Europol are widely using it for police prosecutions (Hayes & Cappa, 2018).

Since the invention of computational analysis, and auto-run systems, researchers in the computer science domain and information security have changed the direction of OSINT from the traditional use for military goals. Researchers are now considering applying data mining and artificial intelligence techniques to OSINT since it extracts intelligence from a significant volume of data. Natural language processing tools and algorithms are being employed for data structuring, automatic translation, information retrieval, and further examination of the findings (Yang & Lee, 2012).

Cyber security OSINT are gathered from social media and blog posts, threat reports, and identified threat objects like malware (Noel, 2021).

**2.3.3 OSINT tools**

There are certain tools researchers and cybersecurity professional use to gather intelligence about their target or current trends on the cyberspace. The quality and accuracy of the intelligence sometimes depend on the platform or tools used.

Most of these tools rely on artificial intelligence to extract information. These tools perform functions like locating publicly available information, collecting relevant details and extracting information. OSINT tools query various social media, websites, search engines and specific platforms or repositories tailored towards cyber security.

Information security experts carry out investigations using some automated tools. Examples of OSINT tools are Recon-ng, searchh diggity, creepy, the harvester and shodan (Chauhan & Panda, 2015).

**2.4 Machine Learning for Cyber Threat Intelligence**

Organizations can use machine learning in gathering, rectification, evaluation of the massive volumes of data required to safeguard them from the ever-changing threat landscape. Most of the non-automated processes that drag back today's cyber security teams can be automated with machine learning modeling. Cybersecurity professionals and machine learning experts can work together to create a safe and security environment for businesses. It is a synergistic connection that produces astonishing benefits as it can help detect threats and threat sources ahead of time (Webroot, 2022).

The benefits of using machine learning techniques on open-source CTI include provision of context, proactive mitigation of threat, connecting seemingly unconnected threats and profiling attackers based on similarity of their approaches (Noel, 2021 & Ghazi et al., 2018).

**2.4.1 Machine Learning Paradigms**

Machine Learning allows a machine to learn from experience and data samples without specific instructions from a user or programmer; the machine forms its own understanding. The paradigms as described by Neumann et al. (2019) are:

1. Supervised Learning – uses labelled data (data which entails information about the class it belongs) to learn. The method uses tagged samples to create a classifier which is a function that assigns labels to samples, including those that the algorithm is not familiar with e.g., classification problems, microarray data from cells for the labels to show whether the sample cells are malignant or healthy (Wittek, 2014).
2. Unsupervised Learning – learns with unlabeled data to find structure in the samples, data points must be assigned to a specific cluster of similar points, without prior information e.g., clustering problems, finding clusters of identical instances in a growing collection of text documents indicates thematic changes over time, emphasizing debate patterns, and identifying out-of-fashion themes.
3. Semi-supervised Learning – partially labelled data are available, and models are tested to see if they can enhance categorization by combining labelled and unlabeled data. Many of these models use generative and probabilistic methods.
4. Reinforcement Learning – Although there is no labelled data available, a mechanism is employed to estimate the machine's performance through rewards. Based on the feedback (rewards) it receives, the computer attempts a variety of choices and learns which acts are the best.

**2.5 Survey of Methodologies**

Recently, several studies have leveraged machine learning for Cyber Threat Intelligence with different data sources. This section gives a summary of existing works that used Twitter, hackers’ forum or other open sources and employed machine learning algorithms for Cyber Threat Intelligence.

Deliu et al. (2017) investigated the potential of Machine Learning approaches to rapidly sift through hacker forums for pertinent threat intelligence using Convolutional Neural Network (CNN). The authors compared the text classification performance of CNN to Support Vector Machine (SVM) and found that SVM achieved high levels of performance compared to CNN. The result showed an accuracy of 98.10%, precision of 98.24%, recall of 97.02% and 97.60% F1 score.

Vangore (2018) leveraged machine learning to develop a small degree of intelligence from the data gathered using Naïve Bayes, Decision Tree and Max Entropy for three tests. Naïve Bayes’ accuracy for test 1 100%, test 2 90% and test 3 100%. Decision Tree, 90%, 100% and 90%, Max Entropy 70%, 70% and 60%.

Ghazi et al. (2018) in their paper titled “A Supervised Machine Learning Based Approach for Automatically Extracting High-Level Threat Intelligence from Unstructured Sources” extracted CTI (high-level indicators) from unstructured cyber threat information sources using a combination of Natural Language Processing NLP-based learning of a Named Entity Recognition (NER) model for extracting CTI and machine learning and used them to perform correlation and quality analysis. The authors got a precision of 70% providing comprehensive threat reports in Structured Threat Intelligence Exchange (STIX) which is a widely established industry standard for CTI. There is need to improve the model’s precision and recall.

Farooq and Otaibi (2018) developed effective machine learning techniques based on analytical and empirical evaluations of obtained data, utilizing many algorithms for prediction, classification, and forecasting at the same time. To choose a standard reference for cyber threat cases prior to their machine learning analytics for cyber threat detection, the authors created multiple Mitre taxonomies such as Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK), Common Attack Pattern Enumeration and Classification (CAPEC), and Malware Attribute Enumeration and Characterization (MAEC). They employed Kmeans clustering, PCA (Principal Component Analysis) preprocessing for dimensionality reduction, and StandardScalar for data scaling without affecting variance. They employed Clustering Algorithms to do Data Rate Analytic, then Linear Regression to anticipate user behavior, OneClass SVM (OCSVM) Classifier to detect anomalies in process executions, and Random Forest to classify messages.

Kim et al. (2018) proposed a platform to systematically deal with cyberattack information, which will then classify the attack into a group, then create the information needed for prompt response. The information created previously by the platform is used for automated analysis to establish correlation to determine a value for the relationship between attacks. Also, to forecast and block threats based on the monitored values of the most similar past and present phase of attack before they cause damages.

Zhou et al. (2018) developed an end-to-end neural-based sequence labeling to identify indicators of compromise automatically from cybersecurity reports. Their experimental result showed that their model performed better than other sequence labeling models, it had an average precision of 90.4%, F1 score of 88% and recall of 87.2%.

Dionísio et al. (2019) used Deep Neural Networks structures to process a data stream of cybersecurity information identified from Twitter. The authors retrieved important security-related information and generated useful knowledge that could be used by a Security Operations Center. They utilized a binary classifier based on CNN and carried out named entity identification and classification tasks. The named entity identification test had an average F1-score of 92%, while the classification task had a true positive rate of 94% and a true negative rate of 91%.

Serketzis et al. (2019) utilized AlienVault Open Threat Exchange (OTX) data to build on and enhance a Digital Forensic Readiness (DFR) system that integrates actionable CTI to increase DFR maturity levels. The algorithm was able to pinpoint the root causes of information security issues with accuracy of 90.73%, 96.17% precision, and 93.61% recall, and reduced the amount of data that digital forensic investigators had to analyze. It is a contribution since it shows that CTI may be used in digital forensics activities. Second, it shows and evaluates an effective solution to improve operational DFR. The model limited the volume of content an analyst must examine, reduced the time it takes to conduct a forensic investigation, reduced the cost of forensic analysis, determined the causal factors of an incident quickly and precisely, and identified threats that may have harmed an organization's security posture.

Noor et al. (2019) utilized a novel machine learning-based methods to track cyber threats based on attack signatures they encountered. To establish a semantic network, the framework conceptually combines threats and TTPs from well-known threat sources with accompanying detection methods. By constructing probabilistic links between threats and TTPs, this network is then used to determine threat occurrences. A TTP taxonomy dataset is used to train the system, and threat objects retrieved from threat reports are used to assess its performance. Even when TTPs are missing or fraudulent, the framework predicts assaults with 92 percent accuracy and negligible false positives. A system trained with 133 TTPs from 45 threat types has an average detection time of 0.15 seconds for a data breach incident.

Gautam et al. (2020) combined machine learning and deep learning using neural network to automatically classify hacker forum data into predefined categories and develop interactive visualizations that enables CTI practitioners to probe collected data for proactive CTI. They developed two deep learning models using Long Short-Term Memory Recurrent Neural Network (LSTM RNN) and Gated Recurrent Unit (GRU) RNN. The results show that RNN GRU gives the best result with 99.025% accuracy and 96.56% precision.

Rodriguez & Okamura (2020) classified open source data into a cybersecurity related informational collection, and improved the link's quality using keyword filtering method by implementing word2vec, a unique Multi-Layer Keyword Filtering (MLKF) to produce an affiliated words list that aids in the classification of ambiguous postings to reduce false-positives while getting vast amounts of data They then employed Naive Bayes, stochastic gradient descent, and linear regression to compare it to Bidirectional Encoder Representations from Transformers (BERT) using Naive Bayes, stochastic gradient descent, and linear regression on appropriately categorized data to detect and exclude groups that are not related to threats. When compared to other machine learning and neural network-based models, they got an F1-Score of 99 percent.

Koloveas et al. (2021) used Random Forest and CNN on Stack Exchange Data Dump dataset so that security analysts may quickly locate, gather, analyze, obtain, incorporate, and disseminate cyber-threat intelligence from a range of internet sources using this cyber-threat intelligence procedure. Accuracy for Random Forest is 87.37%, CNN 85.42%; Precision for Random Forest is 89.14%, CNN 84.91%.

Hossein et al. (2021) developed a model using various supervised (classification) and unsupervised (topic modeling) learning models, select and investigate important CTI from hacker forums. Naive Bayes, Logistic regression, Random Forest, Decision Tree and k-nearest neighbors and shallow neural network and one deep neural network-based classifier for supervised (classification). The accuracies vary from 79% to 89% for the eight classes considered out of the twenty newsgroups dataset.

Noel (2021) developed RedAI to examine if OSINT can be properly included into an operational method to accurately classify CTI. RedAI demonstrated using the Structured Threat Information Expression (STIX) language to objectively acquire, correlate, and synchronize intelligence with the MITRE ATT&CK framework and accurately classified unknown test threat intelligence. Groups and courses of action had 100% precision, malware had 85% and the author found that organizations can use OSINT and advanced solutions for a better cyber defense. Advancement should utilize more diverse and higher volumes of data, try ATT&CK types from a diverse pool and new python libraries and strategies.

Preuveneers and Joosen (2021) introduced machine learning based threat detection to the distribution of attribute-based Indicators of compromise. The authors used their model to secure a network for a distributed denial of service and did a comparative analysis of a few machine learning models and some deep learning models and their perceived strength to counter adversaries that tried to evade being detected. Their solution simplified the sharing of machine learning models on threat intelligence sharing and incidence response platforms, also built a fresh taxonomy to link this threat intelligence with machine learning threats adversaries simultaneously reducing false positives and false negatives, thereby improving the accuracy of the machine learning models. Finally, they used their previous work to encrypt the threat intelligence so that unauthorized persons will not be able to access the sensitive machine learning models making the decryption possible for only designated persons.

**2.6 Gaps Identified in the Literature**

Existing literature have shown that machine learning has been leveraged to better enhance Cyber Threat Intelligence. They focused on automatic extraction of threat data from online platforms like twitter and hackers’ forum while some of the authors also developed novel models. The literature review revealed that different sources of open-source data need to be used and other algorithms should also be explored. This dissertation is concerned with the comparative analysis of Decision tree, Naive Bayes, Logistic Regression and Random Forest for cyber threat dataset.

**CHAPTER THREE**

**METHODOLOGY**

1. **1 Research Design**

This study implemented four algorithms Naive Bayes, Logistic Regression, Random Forest and Decision tree and compared their performance on getting Cyber Threat Intelligence from open-source data. Figure 3.1 depicts the framework used in the study. The framework shows the phases involved in the development of the models which includes the collection and preprocessing of data, training and testing the data with Decision Tree, Naive Bayes, Logistic Regression and Random Forest algorithms.



**Figure 3.1: The Conceptual Framework of the Thesis**

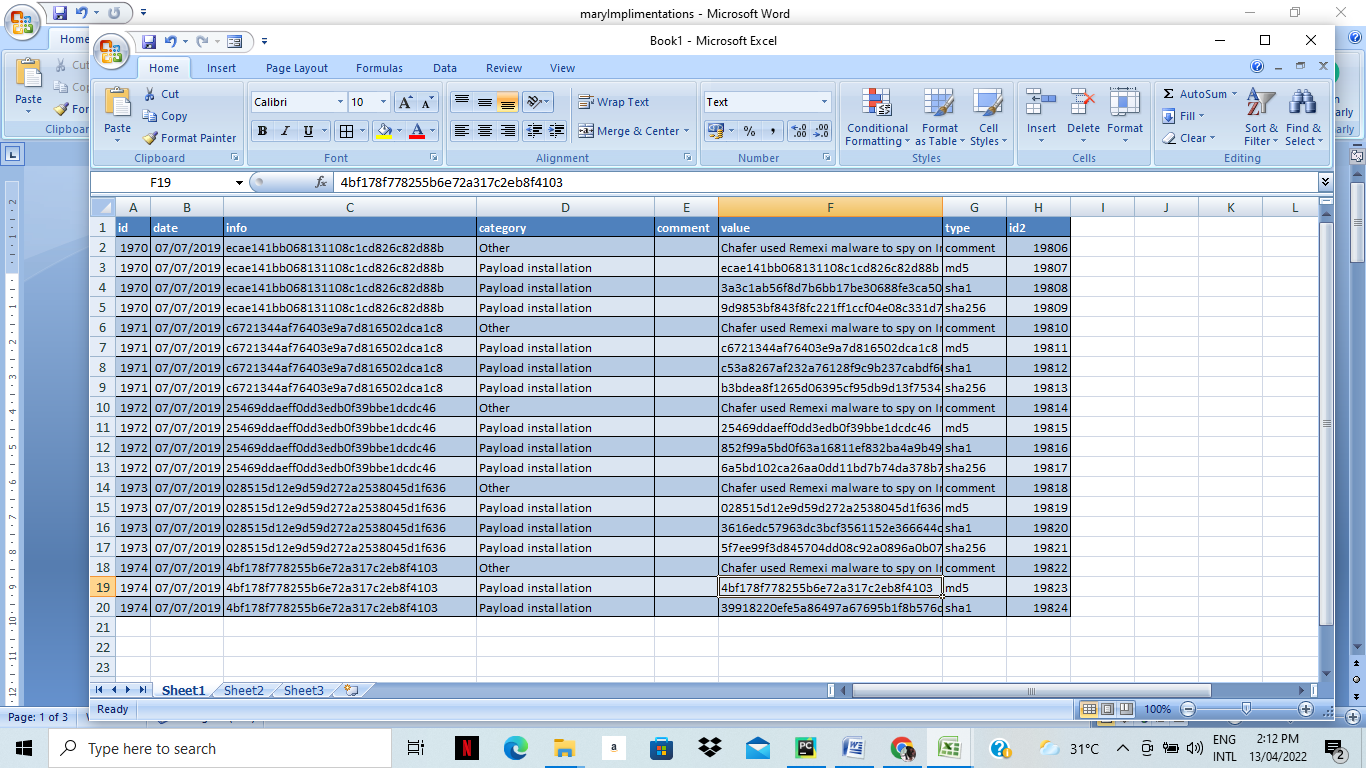
**3.2 Research Design Layout**

All the processes in the research design are explained below given the details of their descriptions and their implementation.

**3.2.1 Dataset**

The dataset employed in this comparative analysis is a Cyber Threat Intelligence (CTI) data that was automatically generated from public security report platforms and malware repository databases by Kim and Kim (2018). The dataset was in an extensible markup language (XML) format which entails roughly 640,000 records gathered from various security reports produced between January 2008 and June 2019. The URL, IP address, host, email account, hashes (such as MD5, SHA1 and SHA256), registry, common vulnerabilities and exposures (CVE), file names ending in specified extensions, and the program database (PDB) path are all included in the contents. 70% of the dataset was used for the training set to fit each model after which the performance was evaluated with the test set 30% for testing for an unbiased model evaluation.

The screenshot of the CTI dataset, in comma separated values (csv) format, used for the implementation of this project work is presented in Figure 3.2 for user to understand the level of abstraction represented in Figure 3.3.



**Figure 3.2: The Dataset used for the Thesis**

**3.2.2 Data Preprocessing**

Apparently, none of the selected machine learning models can work efficiently with the dataset shown in Figure 3.1 because some cells contain empty entries, some fields are not important, and most entries are categorical data.

In order to solve the stated problems associated with the dataset, some important operations were performed on the dataset. Firstly, the dataset was split into training and validation sets, the empty entries were eliminated, LabelEncoder was used to fix categorical features, and data normalization using the StandardScaler library part of data preprocessing tasks. The output of the preprocessing tasks is depicted visually in Figure 3.3.

The function “train\_test\_split” is imported from scikitlearn in order to split the dataset into the 70% train and 30% test subsets.

*from sklearn.model\_selection import train\_test\_split*

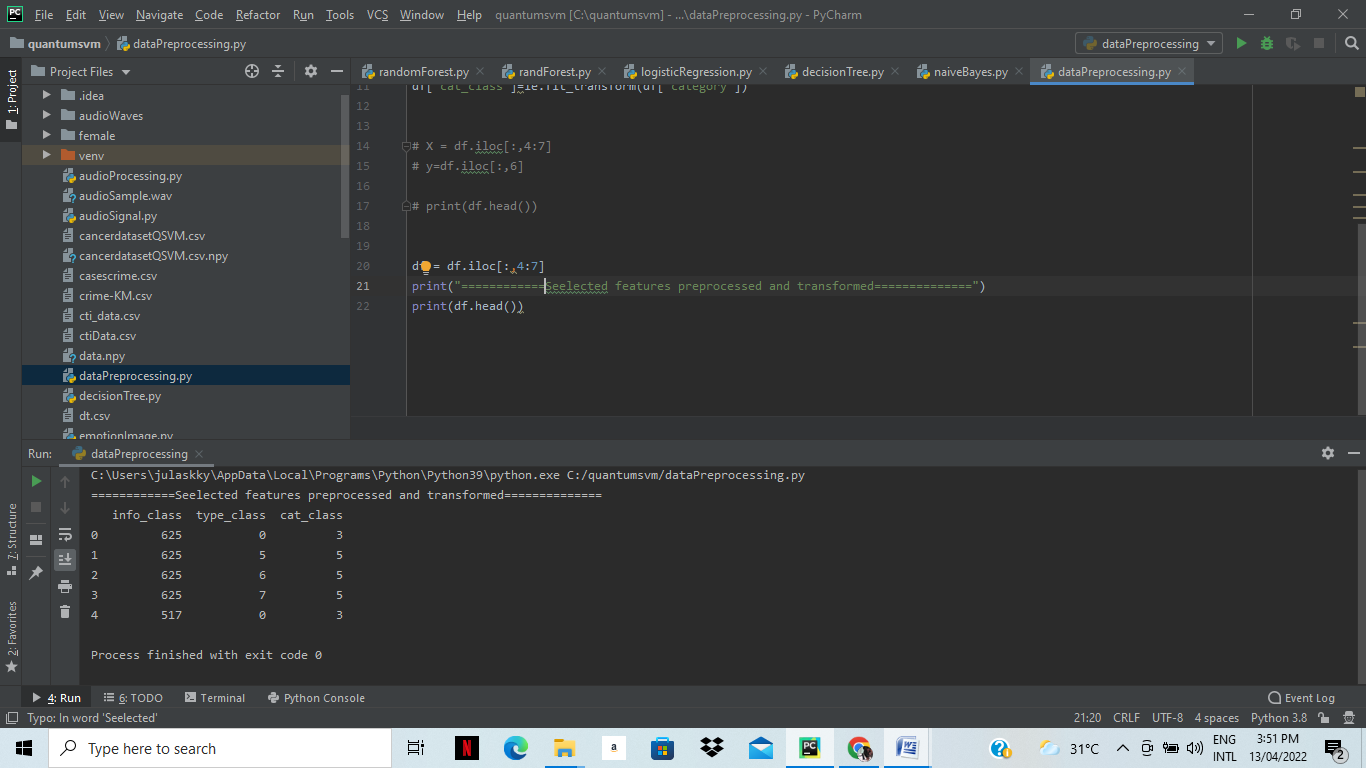
*X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=1)*

Elimination of empty entries provides clean and relevant dataset that can be further used in classification. This can be further broken down into smaller steps namely, stop word removal, tokenization and stemming. The inputs are scaled to a set interval ([-1, 0] or [0, 1]) to not get any bias towards specific inputs that happens to have large values. This scaling has a large effect on the accuracy of the models. The same scaling factor is also used on both the training and testing data. This was done using the “fit\_transform” function.

Feature selection is a method of reducing the dimensionality of features in a dataset. Dimensionality reduction involves diminishing the quantity of random features by only considering series of dominant crucial features. The final prediction is usually based on several features, the more the features the higher it is to visualize and work on the training dataset.

The feature selection process minimized the number of columns from 8 to 3 most important ones with the original ‘info’, ‘type’, and ‘category’ columns being renamed to ‘info\_class’, ‘type\_class’, and ‘cat\_class’ respectively in order to avoid confusion. Also, LabelEncoder object was used to convert the categorical data/values to their numerical representations for easy manipulation by the selected machine learning models.

Machine learning models perform better on normalized data. The “StandardScaler” class changes the values to a similar scale without disrupting the difference in their range. It sets the mean and standard deviation of the dataset to a Standard Gaussian distribution with the mean of 0 and the standard deviation of 1. The processed dataset is shown in Figure 3.3



**Figure 3.3: The Preprocessed Dataset**

**3.2.3 Classification models**

There are important steps to take in building a classifier, these activities are importing packages, libraries, functions and classes; getting the appropriate dataset and transforming it; creating the model and training with the dataset and finally, the performance evaluation of the model. Different classifiers have different performance on any given data set based on the diversity, or training time of the data. The classifiers in this study are described below with their pseudo codes.

1. **Naïve Bayes Classifier Algorithm**: is mainly used for classification. This classifier is built on Bayes' theorem defining its use of probability to forecast the category a combination of features belongs in such a way that all the values are independent. Naïve Bayes Classifier is known to be simple, and it is quite popular and often used as it performance surpasses some of the more modern classifiers.

Baye’s theorem in the equation (1) below.

Where,

P(A) is the prior probability, the probability of hypothesis being true.

P(B) is the probability of the evidence.

P(A\B) is the probability of the evidence given that the hypothesis is true.

P(B\A) is the probability of the hypothesis given that the evidence is true.

Prediction of membership probabilities is made for every class such as the probability of data points associated with a particular class. The class having maximum probability is appraised as the most suitable class. According to Dwivedi (2020) this is also referred to as Maximum A Posteriori (MAP).

The MAP for a hypothesis is:

MAP(H) = max((H|E) \* P(H) / P(E))

MAP(H) = max(P(E|H) \* P(H))

P(E) is evidence probability used to normalize the result. The result will not be affected if removed.

The Bayes’ theorem in (1) can be rewritten as (2) for easy understanding:

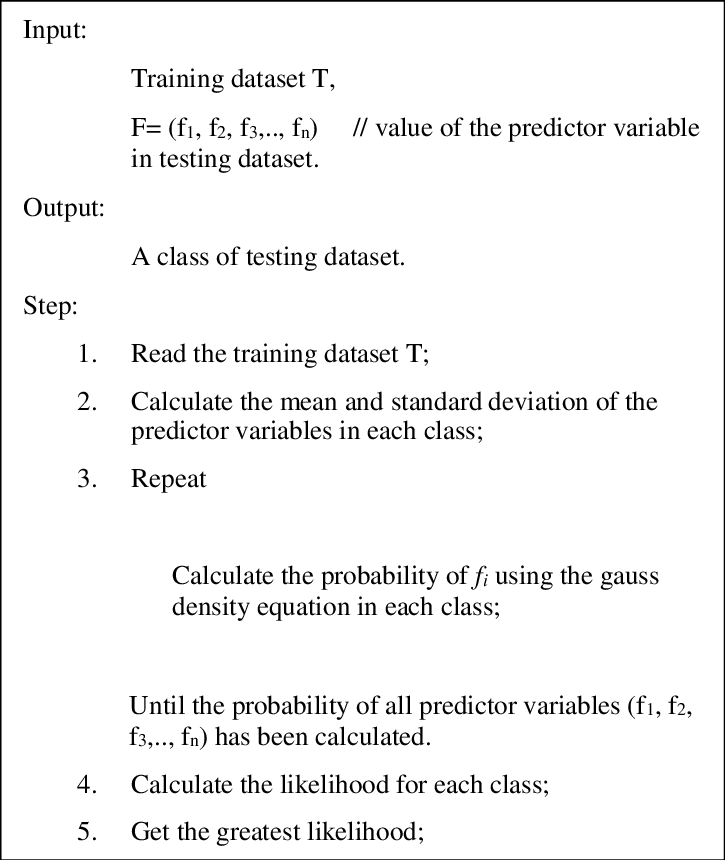
The equation will solve for the probability of y given input features X.

Naive Bayes classifiers assumes that variables are independent, the Existence or absence of a variable does not impact the existence or absence of any other variable (Dwivedi, 2020 & Shin, 2022). This is represented in (3)

Also, since we are solving for y, P(X) is a constant which means it can be removed from the equation and a new proportionality can be introduced. This leads to this (4):

The class “y” with the maximum probability is chosen. Argmax finds the argument with the maximum value from a target function, the maximum y value is in (5).

Naïve Bayes was chosen because it of its flexibility. It predicts class of text data quickly (Ray, 2017) and can also be easily trained on small dataset as well as large datasets.



**Figure 3.4 Pseudo code for Naive Bayes Source: (Saputra et al, 2018)**

1. **Logistic Regression**: is one of the classification methods used in supervised machine learning. It fundamentally represents a double target variable which can either be 0 or 1 and it practically works on previously supplied data to evaluate or forecast the possibility that an event will take place.

Predominantly, logistic regression takes advantage of a logistic function to model a binary dependent variable, while numerous highly complicated modifications are available. Logistic regression is an advanced type of regression analysis that approximately evaluates the variables of a logistic model (Tolles, & Meurer, 2016).

The goal of logistic regression is to estimate the likelihood of an event occurring based on past data. It's used to describe a binary dependent variable, which has only two possible outcomes: 0 and 1 (Chandrasekaran, 2021). Logistic regression was chosen because of the size of dataset available.

The formula for logistic regression is gotten from the standard linear equation for a straight line in (6) below:

The [logistic function or sigmoid function](https://en.wikipedia.org/wiki/Logistic_function) is used to calculate probability in logistic regression. It is a simple S-shaped curve used to convert data into a value between 0 and 1. The Sigmoid function in (7) below:

The standard linear formula was transformed to the logistic regression formula in (8).

There are some important assumptions of the logistic regression model (Wakefield, 2022) that were checked before deploying the model, these assumptions are:

1. The target variable is binary

ii. The predictive features are categorical or continuous

iii. Features are independent

iv. Sample size is adequate

After these assumptions have been checked, there is a need to check for missing values, the isnull() method is called, then the sum() method “Dataset.isnull().sum()” in Python.

1. **Decision Trees:** can be used for classification and regression analysis. A decision tree is like a graph and has a format of a tree which describes a node as test carried out on a variable while branches are all the achievable results of a choice. There are two phases involved in implementing decision tree algorithm:

Preprocessing

Prepare the data for analysis.

Using the Python sklearn module, split the dataset into train and test.

The classifier is then trained.

Analysis

Predict.

Determine the accuracy.

#### Importing the data: This study used pandas’ package in python to import the dataset

#### Splitting the data: the dataset was split into training and testing dataset using the train\_test\_splitModule in sklearn prior to training the model.

#### Pseudocode for decision tree

Choose the suitable parameters and insert it at the top of the tree's root node.

Divide the dataset's training set into subsets.

Ascertain that each subset of the training dataset has the same variable.

Repeat steps 1 and 2 upon every subset to find leaf nodes in all branches

Figure 3.5 describes a decision tree showing the root node to the leaf.

Root Node

A

B C

Decision Node

Terminal Node

Terminal Node

Terminal Node

Terminal Node

Terminal Node

Splitting Branch/Sub-Tree

Decision Node

Decision Node

**Fig. 3.5 A Decision Tree**

Where:

The root node is the whole population which is split into two or more similar sets.

Splitting is the process of dividing a node into two or more sub-nodes.

Decision node is the splitting of a sub-node into more sub-nodes.

A leaf / terminal node is a node that does not split, it has no child.

Pruning is the removal of the sub-nodes of a decision node, it is the reverse of splitting.

A branch / sub-tree is a subsection of the entire tree.

A parent node is a node that is divided into sub-nodes (child nodes).

A child node is a sub-node of a parent node.

Decision Trees algorithm was chosen because it does not need so much data preprocessing and is not affected by missing values. A decision tree is a tree layout that looks like a chart and employs a branching mechanism to show every possible outcome of a decision. Each node in the tree represents a test on a single variable, with each branch representing the result of that test (Wakefield, 2022). Scikit-Learn uses the Gini impurity to calculate the impurity criterion or node impurity, the equation below shows the GINI impurity formula (Bujokas, 2021) for a decision tree in (9) below:

Where is the frequency of a label at a node and is the number of unique labels.

Entropy in (14) is an alternative way to calculate impurity criterion

Information Gain in (11) is used to split the data after the entropy has been calculated. It is the reduction in entropy after splitting the dataset on an attribute (Ronaghan, 2021):

T = target variable

X = Feature to be split on

Entropy (T, X) = The entropy calculated after the data is split on feature X

1. **Random Forests**: is usually employed to perform either classification or regression analysis. It is an ensemble learning method in the sense that it combines many dependent algorithms that can altogether yield exceptional results to enhance its performance in order to give an outstanding result; a concept called bagging (Bootstrap and Aggregation).

Random forests are otherwise known as random decision forests because it begins with just one decision tree that has an entry at the apex. As the data progresses down the tree, it is separated into smaller and smaller sets based on specified variables. Every decision tree has a high variance, but when all of them are combined in parallel, the final variance is low since each decision tree is perfectly trained on that specific sample data, and thus the outcome is based on multiple decision trees rather than just one. The majority voting classifier is used to determine the final output in a classification challenge.

Random Forest is a machine learning algorithm that is widely embraced algorithms for feature selection. It has a built-in library that can be called so it does not need to be extra coding for it. It selects a smaller subset of features.

**Algorithm for random forest algorithm**

Install the libraries that are required.

The dataset should be imported and printed.

All rows and columns 1 are set to X, while all rows and columns 2 are set to Y.

Fit the dataset to the Randomforestregressor

Predict a new outcome.

Use scatter plots to visualize the outcome.

**3.2.4 Performance Evaluation Metrics**

It is paramount to know how accurate the model worked by obtaining the performance metrics to evaluate the classifiers before comparative analysis. The derived results were subjected to analysis to evaluate their performance. The performance evaluation metrics of the classifiers in this study were evaluated with respect to their accuracy, Precision, F-score, Recall. The terms are defined below:

Where:

TP (True Positives) = cases that were appropriately classed as positives,

TN (True Negative) = negative cases were appropriately categorized,

FP (False Positives) = positive cases that were wrongly labeled

FN (False Negative) =negative cases were mistakenly categorized.

Cross validation using training dataset and testing dataset

There are many ways in which this can be done but this study used cross-validation because of its efficiency in multiple train-test split which gives more accurate estimate of outliers and detects overfitting. This divides the training dataset into a portion for training which is usually with 70% of the training dataset and the same number of testing portion which is the remaining 30% of the dataset. After training the classifier with the training data, it is evaluated against the testing data to determine performance of each of the algorithms. This process is repeated countless times in which each time an average is calculated for each of the metrics. It is obvious that the more the data, the more the training or testing data available to be used.

**3.3 Research instruments/Tools**

The system was implemented on a Windows Operating System using Python programming language. All Development, Testing and Design were implemented on a Windows 8.1 has an Intel Core i5 processor with a 3.40 GHz clock speed and 8 GB of RAM.

**A Hardware Requirements**

1. Personal computer with Intel core 15 or higher processor recommended.
2. 8 GB of RAM recommended.
3. The processing speed of 3.6 GHz or higher recommended.
4. Hard disk memory of 500GB or higher recommended.

**B Software Requirements**

1. Operating System: Windows 7 or macOS E1 Capitan or higher.
2. Python Environment.

**3.4 Data Collection**

The dataset used in this study is a Cyber Threat data that was automatically obtained from reports on freely available platforms and malware repository databases. The dataset is in a formed in an extensible markup language (XML) format which entails roughly 640,000 records gathered from various security reports produced between January 2008 and June 2019. 70% of the data which is the training dataset was loaded as a file in the classifier object to allow it to predict the newer data.

According to Deliu et al. (2017), common security keywords that indicate a cyber threat and used to gather threat report data are adware, antivirus (kaspersky, avast, avira, etc.), backdoor, botnet, chargeware, crack, crimeware, crypter, cve, cyberweapon, ddos, downloader, dropper, exploit, firewall , hijack, infect, keylogger, logic bomb, malware, monetizer, password, payload, ransomware, reverse shell, riskware, rootkit, scanner, security, shell code, spam, spoof, spyware, stealware, trojan, virus, vulnerability, worm, zero-day, zeus, username, passwords, login, email, ddos, backdoor, crack. Indicators of compromise such as hashes, internet protocol addresses, domain names, uniform resource locator, email addresses, dynamic link libraries, registry keys etc.

**3.5 Data Analysis Algorithm**

This dissertation was developed in python. It is a programming language that has many libraries for building machine learning model, the libraries specific to this study are scikitlearn, matplotlib and pandas. Some of the libraries are described below:

**Scikit learn** is a popular python package that includes several important libraries for machine learning. It has the capability to perform several functions in building a machine learning model such as data preprocessing, reducing dimensionality, model validation, choosing the best model and solving classification problems.

**NumPy** is a library of codes, a file containing functions that can be called in the python application that is numeric, it provides mathematical functions for calculations, and it also reads data in NumPy arrays as well as performs manipulations.

**Pandas**: is a python library used to read and write files. It can also use data frames to manipulate data.

**Matplotlib** is the library that was used to visualize the classification problem. It is comprehensive and it provides plots with very high quality.

**CHAPTER 4**

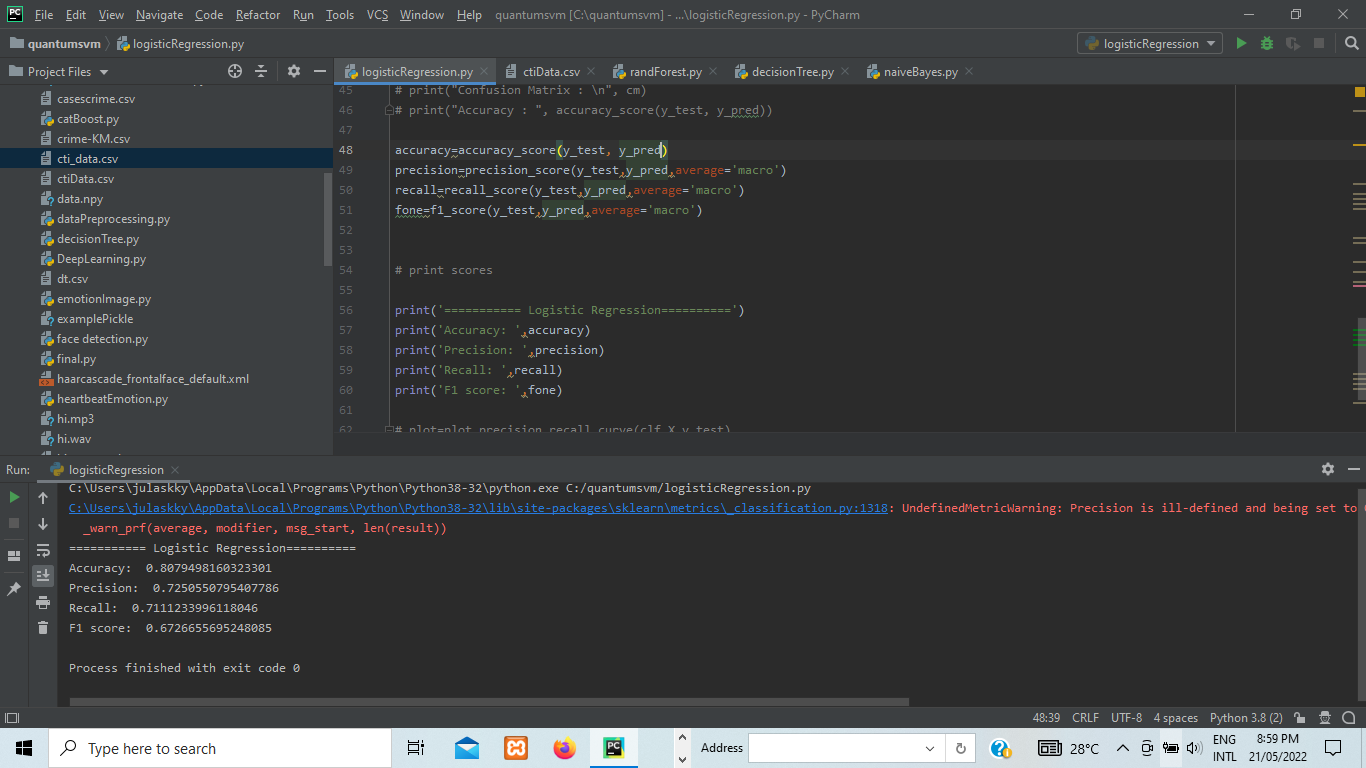
**RESULTS AND DISCUSSION**

**4.1** **MACHINE LEARNING MODELS**

Four machine learning classifiers were utilized to model the stated cyber threat security problem under study in order to fulfill the main aim of this dissertation. However, as implemented in all the four classifiers, the dataset is divided into two in a ratio of 0.7 and 0.3 for training and testing, respectively. Decision Tree, Logistic Regression, Random Forest and Nave Bayes classifiers are presented in this section along with their performance scores.

* + 1. **LOGISTIC REGRESSION**

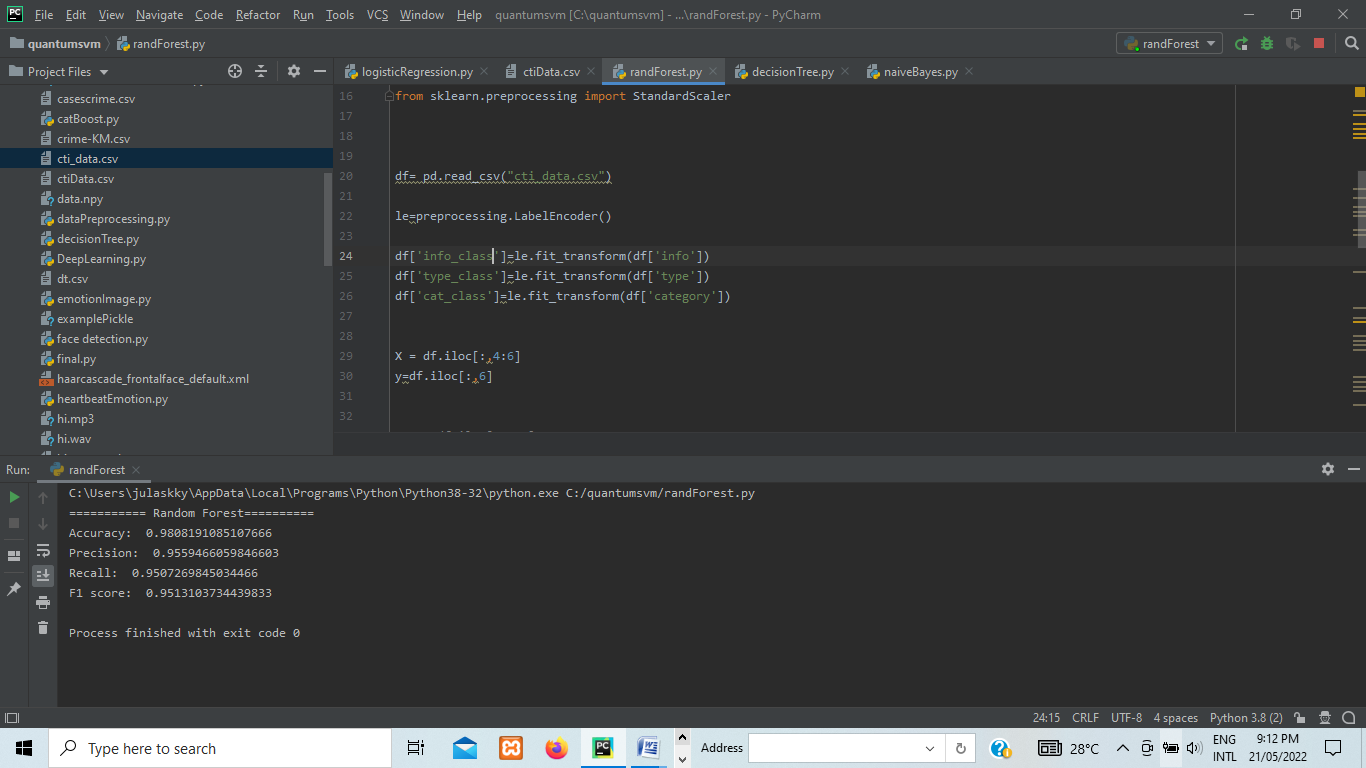
This model had an accuracy of 80.79%. It is one of the machine learning classifiers used for modeling the cyber security threat observed in this study and when assessed after training, it delivered a performance score of 72.5% Precision, 71.11% Recall, and F1 score of 67.27% as depicted visually in Figure 4.1.



**Figure 4.1: Performance Scores for Logistic Regression**

**4.1.2** **RANDOM FOREST**

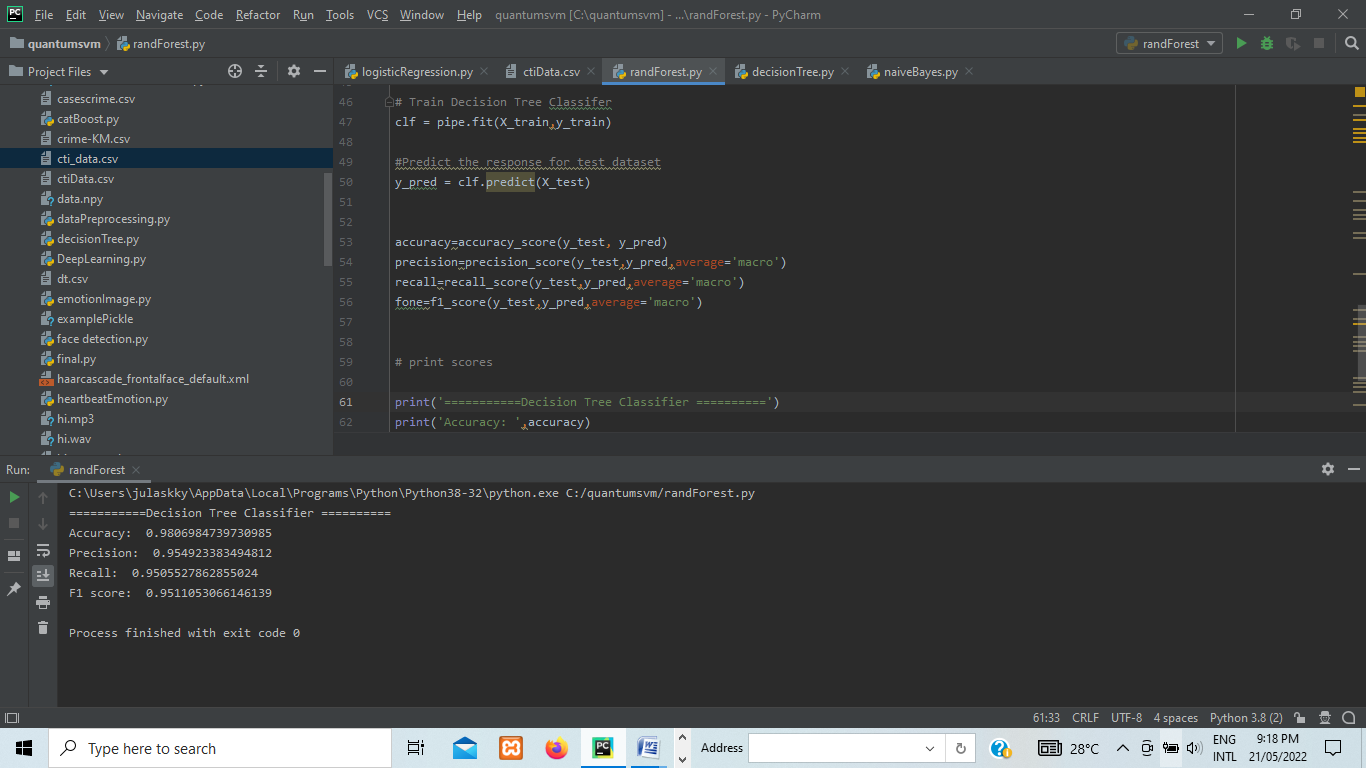
Random Forest classification analysis presents the mode of the classes as the output (classification), while its regression analysis presents the mean/average prediction of the each of the many decision trees (Athey et al., 2019). The performance of this model was the best of all the algorithms; nonetheless, the screenshot is shown below. After Random Forest machine learning model was used to examine the problem, it produced a performance score of 98.08% for Accuracy, 95.59% Precision, 95.07% Recall, and 95.13% F1 score, after training. This is depicted visually in in Figure 4.2.



**Figure 4.2: Performance Scores for Random Forest**

**4.1.3 DECISION TREE**

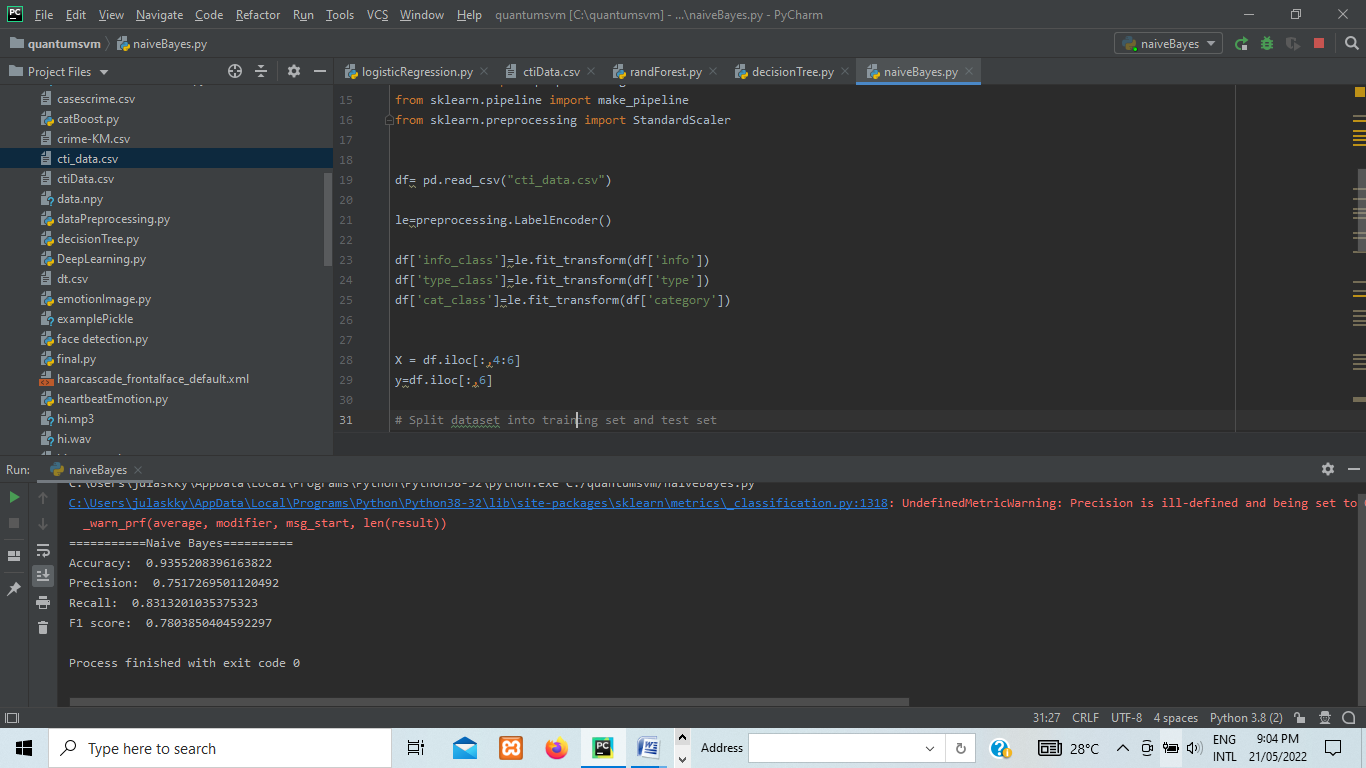
After being fully trained, Decision Tree produced a performance score of 98.07 percent for Accuracy, 95.49 percent Precision, 95.05 percent Recall, and 95.11 percent F1 score, which was one of the machine learning algorithms used. This is depicted visually in Figure 4.3.



**Figure 4.3: Performance Scores for Decision Tree**

**4.1.4 NAÏVE BAYES**

After being fully trained, Naive Bayes had a performance score of 93.55% for Accuracy, 75.17% Precision, 83.13% Recall, and 78.04% F1 score. Figure 4.4 illustrates this clearly.



**Figure 4.4: Performance Scores for Naïve Bayes**

**4.2 RESULT ANALYSIS**

With respect to the considered problem domain, Table 4.1 is the tabular representation of the summary of the results of accuracy, precision, recall and F1 score of all the four machine learning algorithms involved in the comparison in this dissertation.

Random Forest demonstrated excellent performance considering its accuracy with an approximate score of 0.9808 followed by Decision Tree with a score of 0.9806, then Naïve Bayes with an approximate score of 0.9355 while Logistic Regression classifier had the least score of approximately 0.8079 relatively.

Also, in terms of precision, Random Forest has the highest performance with an approximate score of 0.9559 followed by Decision Tree with a score of 0.9549, then Naïve Bayes with an approximate score of 0.7517 while Logistic Regression classifier had the least score of approximately 0.7250.

The recall for the four algorithms similarly proved Random Forest to be the best with an approximate score of 0.9507 followed by Decision Tree with a score of 0.9506, then Naïve Bayes classifier with an approximate score of 0.8313 while Logistic Regression had the least score of approximately 0.7111.

In terms of F1 score, Random Forest has the highest score with an approximate score of 0.9513 followed by Decision Tree with a score of 0.9511, then Naïve Bayes with an approximate score of 0.7804 while Logistic Regression classifier had the least score of approximately 0.6727 relatively.

**Table 4.1: Performance Scores Comparison**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **Logistic Regression** | **Random Forest** | **Decision Tree** | **Naïve Bayes** |
| **Accuracy** | **0.807949816** | **0.980819109** | **0.980698474** | **0.935520841** |
| **Precision** | **0.725055080** | **0.955946606** | **0.954923383** | **0.751726950** |
| **Recall** | **0.711123400** | **0.950726984** | **0.950552786** | **0.831320104** |
| **F1 score** | **0.672665570** | **0.951310373** | **0.951100530** | **0.780385040** |

For the purposes of clarity and visualization, the summary of the performance analysis of the four classifiers considered in this study are presented pictorially in Figure 4.5.

**Figure 4.5: Chart Presenting the Performance Metrics for the four Algorithms**

**4.3** **DISCUSSION**

In studying and forecasting future assaults and attack trends, the machine learning approach to cyber security has been successful. To construct threat intelligence tools that can forecast, machine learning techniques and classification algorithms namely Decision Tree, Logistic Regression, Random Forest and Naive Bayes were used. As seen, most of the predictions demonstrate that the model correctly categorizes most of the predictions. However, as can be seen throughout, certain classifications are incorrect. All the algorithms' accuracy was presented, machine learning can be used to combine algorithms to see which one gave the most accuracy performance for the performance metrics in this model. Nonetheless, it does not discover the threat actor's motivations and objectives.

**CHAPTER FIVE**

**SUMMARY, CONCLUSION AND RECOMMENDATION**

**5.1** **SUMMARY**

In the sectors of transportation, energy, healthcare, manufacturing, and communication, the incorporation of elaborate cyber frameworks and an application in an information technology environment has had economic, business, and societal implications for indigenous and intercontinental contexts. Cyber security, on the other hand, remains an issue because any vulnerability in the system might put the entire supply chain at risk. The aim of this dissertation is to enhance cyber security by combining cyber threat intelligence with machine learning for threat analysis and prediction. To analyze and predict the threat, concepts from computer science and cyber threat intelligence as well as a methodical process were used. In this research, a machine learning model was built that can assist cyber security professionals in timely detection of threats with maximum accuracy. Four different machine learning algorithms were considered (Decision Tree, Logistic Regression, Random Forest and Naïve Bayes classifiers). The results of the experiments demonstrated that the Decision Tree, Logistic Regression, Random Forest and Naïve Bayes algorithms were accurate in Majority Voting and identified a list of predicted risks.

Machine learning has proven to be potent in addressing diverse challenging and complex problems in several industries and areas that require human analysis.

Security organizations can leverage machine learning by using machine learning classifiers for threat prediction. This enables organizations to evaluate their existing systems in order to figure out a replacement, augmentation or advancement to improve their cyber security posture.

**5.2 CONCLUSION**

In this research, four different machine learning algorithms were considered for use inclusive of Decision Tree, Logistic Regression, Random Forest, and Naïve Bayes classifiers. For each of the algorithms, a machine learning model was built that can assist cyber security professionals in timely detection of cyber-threats with maximum accuracy. The models were implemented using a CTI dataset, and the accuracy precision, recall and F1 scores of each model was assessed using python programming languages and its libraries.

As discussed in the previous section, experimental results of the comparative analysis showed that Random Forest with well above 98.08% performance accuracy is the most suitable machine learning model for the dataset. Three of the selected machine learning algorithms performed as expected asides Logistic regression which had the least scores in all the four-evaluation metrics. This means that the algorithm is not best suited for the dataset used in this thesis.

**5.3 RECOMMENDATION**

Future work can investigate how to improve the performance of the algorithm and research into why logistic regression had the least scores. This research can be expanded to automatically generate the data to be used in the model and consider working with data in other languages.

**5.4 FUTURE WORK**

Future works can explore other algorithms then train and test the system with different datasets through data quality controls because some real-life situations demand more precision than the one obtained in this research. Further research can be done on how to enhance the overall performance of logistic regression. Researchers can also explore the benefits of CTI in relations to the cyber kill chain and pyramid of pain.

**REFERENCES**

Abu, M. S., Selemat, S.R., Ariffin, A., & Yusof., R. (2018). Cyber threat intelligence - issues and challenges. Indonesia Journal of Electrical Engineering and Computer Science vol. 10, no. 1, pp. 371-379, https//doi: 10.11591/ijeecs.v10.i1.pp371-379

Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. *The Annals of Statistics*, *vol. 47, no. 2*, pp. 1148-1178.

Baker, K. (2022). What is Cyber Threat Intelligence? Retrieved May 9, 2022, from https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/

Bromiley, M. (2016). Threat intelligence: What it is, and how to use it effectively. *SANS Institute InfoSec Reading Room, vol. 15, pp. 172*.

Bujokas, E. (2021). Decision tree algorithm in python from scratch. Retrieved Feb 15, 2022, from <https://towardsdatascience.com/decision-tree-algorithm-in-python-from-scratch-8c43f0e40173>

Chandrasekaran, M. (2021). *Logistic Regression for machine learning*. Retrieved July 28, 2022 from https://www.capitalone.com/tech/machine-learning/what-is-logistic-regression/

Chauhan, S., & Panda, N. K. (2015). Hacking Web intelligence : open source intelligence and Web reconnaissance concepts and techniques. *Computers & Security vol. 55, p. 113.* Syngress.

Conti, M., Dargahi, T., and Dehghantanha, A. (2018). Cyber Threat Intelligence: Challenges and Opportunities. *Advances in Information Security, 70,* pp 1-6*.* https://doi.org/10.1007/978-3-319-73951-9\_1

Dalziel H. (2014). How to Define and Build an Effective Cyber Threat Intelligence Capability. *Elsevier Science & Technology Books 2014, 1st Ed*.

Deliu, I., Leichter, C., & Franke, K. (2017). Extracting Cyber Threat Intelligence from Hacker Forums: Support Vector Machines versus Convolutional Neural Networks. *IEEE International Conference on Big Data,* pp. 3648-3656, https//doi: 10.1109/BigData.2017.8258359.

Dionísio, N., Alves, F., Ferreira, P. M., & Bessani, A. (2019). Cyber Threat Detection from Twitter using Deep Neural Networks. In *2019 International Joint Conference on Neural Networks (IJCNN)* pp. 1-8. IEEE.

Dwivedi. R. (2020). What is Naïve Bayes algorithm in machine learning? Retrieved June 14, 2022 from <https://www.analyticssteps.com/blogs/what-naive-bayes-algorithm-machine-learning>

Evangelista, J. R., Sassi, R. J., Romero, M., & Napolitano, D. (2020). Systematic Literature Review to Investigate the Application of Open Source Intelligence (OSINT) with Artificial Intelligence. *Journal of Applied Security Research,* <https://doi.org/10.1080/19361610.2020.1761737>

Farooq, H. M., & Otaibi, N. M. (2018). Optimal Machine Learning Algorithms for Cyber Threat Detection. *UKSim-AMSS 20th International Conference on Modelling & Simulation,* https://DOI 10.1109/UKSim.2018.00018.

Fastovets, D. V., Bogdanov, Y. I., Bantysh, B. I., & Lukichev, V. F. (2019). Machine learning methods in quantum computing theory. 85. <https://doi.org/10.1117/12.2522427>, 2019.

Gautam, A. S., Gahlot, Y., & Kamat, P. (2019). Hackers’ forum exploit and classification for proactive cyber threat intelligence. *International Conference on Inventive Computation Technologies*, pp. 279–285.

Ghazi, Y., Anwar, Z., Mumtaz, R., Saleem, S. & Tahir, A. (2018). A Supervised Machine Learning Based Approach for Automatically Extracting High-Level Threat Intelligence from Unstructured Sources. *2018 International Conference on Frontiers of Information Technology (FIT),* pp. 129-134, https://doi: 10.1109/FIT.2018.00030.

Grisham, J., Samtani, S., Patton, M., & Chen, H. (2017). Identifying Mobile Malware and Key Threat Actors in Online Hacker Forums for Proactive Cyber Threat Intelligence. *2017 IEEE International Conference on Intelligence and Security Informatics: Security and Big Data, ISI 2017,* pp. 13-18.

Hayes, D. R., & Cappa, F. (2018). Open-source intelligence for risk assessment. *Business Horizons, vol.* 61, no. 5, pp 689–697. <https://doi.org/10.1016/j.bushor.2018.02.001>

Headquarters Department of the Army. (10 July 2012). Open-Source Intelligence. Washington, DC *Army Techniques Publication No. 2-22.9 (FMI 2-22.9).*

Hossein, M. I., Islam, A., Anowar, F., Ahmed, E., Rahman M. M. (2021). Generating Cyber Threat Intelligence to Discover Potential Security Threats Using Classification and Topic Modeling. *arXiv preprint arXiv:2108.06862*. [https://doi,org/1048550/*arXiv:2108.06862.*](https://doi,org/1048550/arXiv:2108.06862.)

Kim, B. I., Kim, N., Lee, S., Cho, H., & Park, J. (2018).A Study on a Cyber Threat Intelligence Analysis (CTI) Platform for the Proactive Detection of Cyber Attacks Based on Automated Analysis.2018 *International Conference on Platform Technology and Service (PlatCon), 3(18) 31.*

# Kim, D., & Kim, H. (2019). Automated Dataset Generation System for Collaborative Research of Cyber Threat Analysis. *Security and Communication Networks, vol 2019, Article ID 6268476, 10 pages.* <https://doi.org/10.1155/2019/6268476>.

Koloveas, P., Chantzios, T., Alevizopoulou, S., Skiadopoulos, S., & Tryfonopoulos, C. (2021). INTIME: A Machine Learning-Based Framework for Gathering and Leveraging Web Data to Cyber-Threat Intelligence. *Electronics vol. 10, no. 7,* p 818. https:// doi.org/10.3390/electronics10070818

Liew, A. (2007). Understanding Data, Information, Knowledge and their Inter- Relationships. *J Knowl Manag Pract*. 2007; vol. 8, no. 2, pp 1–7.

Neumann, N., Phillipson, F., & Versluis, R. (2019). Machine learning in the quantum era. *Digitale Welt, 3,* pp. 24-29. https://doi.org/10.1007/s42354-019-0164-0.

Noel, L. (2021). RedAI: A Machine Learning Approach to Cyber Threat Intelligence. *Masters Theses, 2020-current. 81.* https://commons.lib.jmu.edu/masters202029/81.

Noor, U., Anwar, Z., Malik, A. W., Khan, S., & Saleem, S. (2019). A machine learning framework for investigating data breaches based on semantic analysis of adversary’s attack patterns in threat intelligence repositories. *Future Generation Computer Systems, vol. 95*, issue C, pp. 467-487. <https://doi.org/10.1016/j.future.2019.01.022.>

North Atlantic Treaty Organization (2001). NATO Open Source Intelligence handbook. Brussels: NATO Office of Information and Press.

Ponemon Institute. (2016, July). 2016 Cost of Data Breach Study: Global analysis. IBM Security. https://www.bankinfosecurity.com/whitepapers/2016-cost-data-breach-study-globalanalysis-w-2647

Preuveneers, D., & Joosen, W. (2021). Sharing Machine Learning Models as Indicators of Compromise for Cyber Threat Intelligence. *Journal of cybersecurity and privacy*, *1*, pp. 140–163. https://doi:10.3390/jcp1010008

Ray, S. (2017). 6 easy steps to learn Naïve Bayes Algorithm with codes in python and R. Retrieved June 13, 2022 from https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

Richelson, J. T. (2016). The US Intelligence Community. *Routledge*, *1,* 7. <https://doi.org/10.4324/9780429494321>.

Rodriguez, A., & Okamura, K. (2020). Enhancing data quality in real time threat intelligence systems using machine learning.[*Social Network Analysis and Mining*](https://kyushu-u.pure.elsevier.com/en/publications/enhancing-data-quality-in-real-time-threat-intelligence-systems-u)*, vol. 10, no .1.* <https://doi.org/10.1007/s13278-020-00707-x>.

Ronaghan, S. (2021). The mathematics of decision trees, random forest and feature importance in Scikit-Learn and Spark. Retrieved June 13, 2022 from https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3

Saputra, M. F., Widiyaningty, T., & Wibawa, A. (2018). Illiteracy Classification Using K Means-Naïve Bayes Algorithm. *International Journal of Informatics Visualization, vol. 2, no. 3*.  <http://dx.doi.org/10.30630/joiv.2.3.129>

Serketzis, N., Katos, V., Ilioudis, C., Baltatzis, D., & Pangalos, G. (2019). Improving Forensic Triage Efficiency through Cyber Threat Intelligence. *Future Internet 2019, vol. 11, no. 7,* 162*.* https://doi.org/10.3390/fi11070162.

Shackleford, D. (2017). Cyber threat intelligence uses, successes and failures: The sans 2017 CTI survey. *SANS, Tech. Rep*.

Shin, T. (2022) A mathematical explanation of Naïve Bayes in 5 minutes. Retrieved August 28, 2022 from https://towardsdatascience.com/a-mathematical-explanation-of-naive-bayes-in-5-minutes-44adebcdb5f

Symantec (2016). Internet Security Threat Report. Network Security.

Tolles, J. & Meurer, W.J. (2016). Logistic regression: relating patient characteristics to outcomes. *Jama, 316(5), 533-534*, 2016.

Tounsi, W., & Rais, H. (2018). A survey on technical threat intelligence in the age of sophisticated cyber-attacks. *Computer Security* 72, pp, 212–233.

US Joint Chiefs of Staff. (2013). Joint Publication 2-0 Joint Intelligence. *Jt Publ. 2013; (October)*:144. https://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/jp2\_0.pdf

Vangore, J. (2018). Machine Learning and Threat Intelligence. Student Scholarship - *Computer Science. 3*. <https://digitalcommons.olivet.edu/csis_stsc/3>

Wakefield, K. (2022). *A guide to the types of machine learning algorithms and their applications*. Retrieved Feb 9, 2022 from https://www.sas.com/en\_gb/insights/articles/analytics/machine-learning-algorithms.html

Wakefield, K. (2022). A guide to the types of machine learning algorithms and their applications. Sas. Retrieved Feb 9, 2022 from https://www.sas.com/en\_gb/insights/articles/analytics/machine-learning-algorithms.html

Webroot. (2022). *What is cyber threat intelligence?* Retrieved Feb 10, 2022 from <https://www.webroot.com/us/en/resources/glossary/what-is-cyber-threat-intelligence>

Wittek, P. (2014). Quantum Machine Learning What Quantum Computing Means to Data Mining. *Elsevier*, pp.153-163. https://doi.org/10.1016/C2013-0-19170-2.

Yang, H. C., Lee, C. H. (2012). Mining open-source text documents for intelligence gathering [Paper presentation]. *2012 International Symposium on Information Technologies in Medicine and Education, pp. 969–973.*

Zhou, S., Long, Z., Tan, L., & Guo, H. (2018). Automatic Identification of Indicators of Compromise using Neural-Based Sequence Labeling. arXiv:1810.10156v1 [cs.AI]