5-5-2021

RedAI: A machine learning approach to cyber threat intelligence

Luke Noel
James Madison University

RedAI: A Machine Learning Approach to Cyber Threat Intelligence

Luke Noel

A Thesis submitted to the Graduate Faculty of

JAMES MADISON UNIVERSITY

In

Partial Fulfillment of the Requirements

for the degree of

Master of Science

Department of Computer Science

May 2021

FACULTY COMMITTEE:

Committee Chair: Brett Tjaden, Ph.D.

Committee Members/Readers:

M. Hossain Heydari, Ph.D.

Xunhua Wang, Ph.D.

ACKNOWLEDGEMENTS

I would like to thank my Committee Chair Professor Brett Tjaden as well as my Committee Members Professor M. Hossain Heydari and Professor Xunhua Wang.

I would also like to extend my gratitude to the entirety of the JMU Information Security faculty and staff.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The world is continually demanding more effective and intelligent solutions and strategies to combat adversary groups across the cyber defense landscape. Cyber Threat Intelligence (CTI) is a field within the domain of cyber security that allows for organizations to utilize threat intelligence and serves as a tool for organizations to proactively harden their defense posture. However, there is a large volume of CTI and it is often a daunting task for organizations to effectively consume, utilize, and apply it to their defense strategies.

In this thesis we develop a machine learning solution, named RedAI, to investigate whether open-source intelligence (OSINT) can be effectively integrated into a working approach that accurately classifies cyber threat intelligence. By focusing on open-source and easily available resources, RedAI demonstrates how to use the Structured Threat Information Expression (STIX) (OASIS, 2017) language to objectify, collect, and integrate intelligence and align it to the MITRE ATT&CK framework (MITRE ATT&CK Enterprise, 2021). To test the accuracy of this solution, machine learning models were built using training data and then further tested with test data to determine the model's effectiveness at classifying unknown threat intelligence. The results showed that RedAI could, with high accuracy, use OSINT cyber threat intelligence data to build a machine learning model and then classifying unknown test threat intelligence. Based off these findings, it is apparent that organizations have the ability to leverage OSINT and advanced solutions to augment their cyber defense posture.

Part I

Introduction

1. INTRODUCTION

In today's cyber world, it is evident how difficult it is to prevent attacks. Advanced Persistent Threats (APTs) have grown their capabilities to factors that make it hard to detect and investigate security breaches. The cyber security space is an increasingly complex industry, which reflects the growing threat landscape. Today, adversaries have developed advanced tactics, techniques, and procedures (TTPs) that allow them to dynamically change their posture while evading commonplace defense mechanisms (Recorded Future, 2019).

In the threat environment today traditional cyber defense mechanisms no longer suffice. Adversary behaviors have become industry-grade and require matching advanced mechanisms by organizations to effectively mitigate risk across their threat landscape (Bromander, 2017). By strategically incorporating cyber threat intelligence (CTI) practices, an organization can enable itself to make proactive, informed, data-backed decisions to combat APTs. While understanding the benefits of CTI is often accepted across the industry, applying it in practice can be a difficult and inefficient task. Threat intelligence is often overwhelming and historically does not follow standardized formats. Because of this, consuming and sharing threat intelligence can be difficult, which in return can cause organizations to forego its use. By applying effective technological methods like machine learning and ontologies, utilization of threat intelligence can become a viable strategy for cyber threat mitigation (Crowdstrike, Kurt Baker, 2021).

An ontology is a set of concepts and categories in a subject area or domain that shows their properties and the relations between them (Gronberg, 2019). Implementing a cyber threat intelligence ontology has the following challenges:

- Lack of methods to define relationships between layers of intelligence to leverage a wholistic and useable viewpoint of attack scenarios. Analysis of siloed information without use case relationships is not enough for organizations to proactively prepare for wide and evolving threats.
- Non-standardized objectivation of intelligence information and data constructs leading to hardships in consumption and sharing of information. Since organizations have not agreed on a standardized representation of CTI, it makes it difficult to consume their intelligence with one another, which has proven to be essential in fighting APTs.
- Data overload due to the volume of information being captured. Adversary TTPs create a vast amount of information making it difficult for security analysts to manually analyze, demanding a more automated and efficient solution.

This thesis introduces a machine learning approach to cyber threat intelligence that highlights the benefits CTI can provide when machine learning practices are used to analyze and classify open-source intelligence (OSINT). This work evaluates different strategies and use-cases to find efficient and real-world focused methods to proactively mitigate threats. While doing so, explicit focus has been placed on incorporating and utilizing industry standards like MITRE ATT&CK and STIX to call attention to the importance of effective and structured strategies for intelligence consumption and sharing.

1.1 RESEARCH QUESTIONS

This thesis aims to answer the following research questions:

1. How can the MITRE ATT&CK framework serve organizations as a cyber threat intelligence (CTI) ontology?
2. Can open-source intelligence (OSINT) be a viable method for collecting and consuming cyber threat intelligence (CTI)?
3. Can machine learning prove to be an accurate and efficient method for harnessing the benefits of a cyber threat intelligence ontology?

To answer these questions, this thesis demonstrates a working solution developed with a cyber threat intelligence focus by processing open-source intelligence data using a machine learning pipeline.

1.2 METHODOLOGY

For this thesis, we developed a working solution called RedAI. RedAI is built with a microservice focused design that can silo each of its components into programmatic services. RedAI was developed using Python 3 with accompaniment from the Scikit Learn (Scikit-Learn) and Pandas (Pandas) Python libraries for machine learning and data representation functionality. The cyber threat intelligence ontology was developed in parallel from the works and findings of the MITRE ATT&CK framework (The MITRE Corporation, 2021). Open-source intelligence data from the MITRE ATT&CK repository was utilized using the Structured Information Expression (STIX) serialization and language format (OASIS, 2017).

Part II

Related Work

2. RELATED WORK

Related work for this thesis falls into three main categories: Cyber threat intelligence, open-source intelligence (OSINT), and machine learning. In the following sections, we will review each of these topics briefly. In Chapter 3 we will demonstrate how all three are combined in RedAI to automatically classify CTI data.

2.1 CYBER THREAT INTELLIGENCE (CTI)

Currently, for organizations to provide proper cyber security risk mitigation, they must develop many different strategies and approaches to combat the increasingly complex adversary threat field (The MITRE Corporation, 2018). Part of the solution is utilizing Cyber Threat Intelligence.

FireEye defines cyber threat intelligence as the following:

"Evidence-based knowledge about adversaries – their motives, intents, capabilities, enabling environments and operations – focused on an event, series of events or trends, and providing a decision advantage to the defender."

Through this definition, CTI has a strong focus on adversaries and their behaviors, due to attacks and defense actions being executed because of human interaction. By taking this approach to defining CTI, it enables a high-level focus on the attackers themselves rather than the low-level techniques that they use. By doing so, this enables organizations to implement proactive defenses based on their adversary's intents and capabilities.

For CTI, FireEye defines the difference between information (which often is not actionable) and intelligence (which often includes content which makes it actionable). Information, for example might include a malicious IP address or malware hash

signature. These alone may help an organization in its defense, but that information is too low-level to form an intelligence-drive defense posture. Intelligence includes information but has added analysis, evaluation, relation, and context. This can include how malware is used from a strategic standpoint, an APT's preferred set of malware tools, the types of industries an APT target, and so on (FireEye, 2021).

Below is a table comparison provided by FireEye that shows the different properties of information and intelligence (FireEye, 2021).

TABLE 1 INFORMATION VS. INTELLIGENCE

Information	**Intelligence**
Raw, unfiltered	Processed, sorted
Unevaluated	Evaluated and interpreted by an analyst
Aggregated from virtually every source	Aggregated from tailored, reliable sources
May be true, false, misleading, relevant or irrelevant	Accurate, timely, complete as possible, assessed for relevancy
Not actionable	Actionable

CTI requires a contextual perspective and has a strong focus on analyzing the results and effectiveness of the information that sits within that intelligence. This can include past, present, and future tactics, techniques, and procedures (TTPs). Further, the relationships between each defined piece of information may be linked together and mapped to create a visual representation.

There have been efforts to develop cyber threat intelligence frameworks to better classify attacks and assess risks across the global threat landscape. One of these

frameworks is the MITRE ATT&CK framework, which provides a globally accessible knowledge base of adversary tactics and techniques based on real-world observations (The MITRE Corporation, 2018). Through ATT&CK, specific threat models and methodologies can be developed to be used across the entire industry and community. The purpose of ATT&CK is to allow sharing of CTI in an effective and open manner that allows for better security across the industry.

ATT&CK, considered to be one of the leading industry standards for CTI frameworks, was originally developed to improve detection of adversaries by utilizing data and analytics. Throughout these efforts, there were four main issues that ATT&CK sought to highlight:

- Adversary behaviors. Focusing on adversary tactics and techniques allowed us to develop analytics to detect possible adversary behaviors. Typical indicators such as domains, IP addresses, file hashes, registry keys, etc. were easily changed by adversaries and were only useful for point in time detection — they did not represent how adversaries interact with systems, only that they likely interacted at some time.
- Existing lifecycle models that did not fit. Existing adversary lifecycle and Cyber Kill Chain concepts were too high-level to relate behaviors to defenses — the level of abstraction was not useful to map TTPs to new types of sensors.
- Applicability to real environments. TTPs need to be based on observed incidents to show the work is applicable to real environments.
- Common taxonomy. TTPs need to be comparable across different types of adversary groups using the same terminology.

It is apparent how the objectives of ATT&CK align with how FireEye defines CTI. By focusing on adversaries, rather than low-level tactics, common and standardized models can be built to be tailored to real-world use cases and aligned with data and analytics. By focusing on these key themes, MITRE has found that intelligence focused solutions can be developed that provide more effective strategies to defense postures across the globe.

2.2 OPEN-SOURCE INTELLIGENCE (OSINT)

There are many different forms of data and intelligence throughout the world's technological ecosystem. Because of this vast array of data, there are many different methods that can be used to gather the data and develop intelligence. The following are the industry defined intelligence gathering disciplines:

TABLE 2 INTELLIGENCE GATHERING DISCIPLINES

Discipline	**Abbreviation**	**Description**
Human Intelligence	HUMINT	Gathered from a person.
Geospatial Intelligence	GEOINT	Gathering from satellite and aerial photography or mapping/terrain data.
Measurement and Signature Intelligence	MASINT	Gathered from an array of distinctive characteristics. Can be further split into different disciplines.
Open-Source Intelligence	OSINT	Gathered from open sources.
Signals Intelligence	SIGINT	Gathered from interception of signals.
Technical Intelligence	TECHINT	Gathered from analysis of weapons and equipment.

Open-source intelligence (OSINT) is the most broadly defined, and widely used, form of intelligence (Air Force Institute of Technology, 2007). It is often misunderstood and improperly used throughout technological solutions. According to U.S. public law, open-source intelligence has the following characteristics (Richelson, 2016):

- Produced from publicly available information.
- Collected, analyzed, and disseminated in a timely manner to an appropriate audience.
- Addresses a specific intelligence requirement.

The main characteristic is that it is publicly available, meaning that anyone can consume the information. It further allows for collecting, analyzing, and making decisions based on the OSINT in an intelligence context. OSINT sources can be divided into different categories based on the source of information (Richelson, 2016):

TABLE 3 OSINT CATEGORIES

Category	**Examples**
Media	Newspapers, magazines, radio
Internet	Online publications, blogs, social media
Public Government Data	Reports, budgets, websites, directories
Professional and Academic Publications	Journals, conferences, academic papers
Commercial Data	Commercial imagery, assessments, and databases
Grey Literature	Technical reports, working papers, newsletters

Within the realm of cyber security, OSINT can be gathered from many different sources: social media and blog posts, threat reports, and identified threat objects like malware.

OSINT collection can be achieved in two ways, passive or active. Passive collection is performed by using threat intelligence platforms to combine threat feeds into a single location for consumption. Active collection is performed by using techniques to query specific targets of information. Active collection is best utilized when the use-case is very specifically defined.

Cyber threat intelligence sources are the origin that the intelligence you have collected came from. A feed is a collection of intelligence from many sources typically of the same type of intelligence. These open-source feeds can be queried and consumed, which allows for them to be directly built into solutions and systems (Recorded Future, 2017).

Further, an intelligence platform is the system that organizes feeds into a stream of intelligence. These platforms pull intelligence from OSINT feeds making it more efficient for consumption across many different areas. These platforms can deliver feeds via serialized methods like STIX or TAXII (OASIS, 2017). Lastly, there are intelligence providers which are the organizations or entities that produce intelligence. An example of a provider, applicable to this thesis's work, is the MTIRE ATT&CK repository.

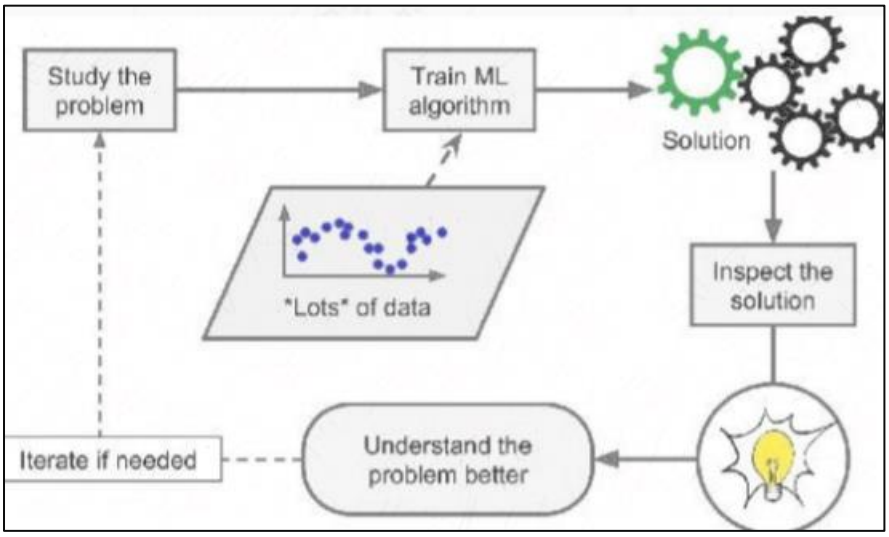
Open-sourced platforms are relatively new within the cyber security field but provide great benefits to the community. These benefits allow thousands of feeds to be combined into a single consumable stream in standardized formats. By building solutions

around open-sourced platforms, consumers can have access to large amounts of data that have already been organized and serialized, which allows the heavy lifting of OSINT data collection to be mitigated and streamlined (Recorded Future, 2017).

2.3 MACHINE LEARNING

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. ML is classified as a subset of artificial intelligence (AI). ML algorithms work by building models on training data to make predictions without being explicitly programmed to do so. ML algorithms can be applied to an extremely wide variety of use cases but are best suited for scenarios where it is difficult or inefficient to develop conventional algorithms to perform the desired tasks (Géron).

FIGURE 1 MACHINE LEARNING SOLUTION DEVELOPMENT (GÉRON)



The benefits of machine learning are typically best realized when the problems at hand require a lot of hard code or manual tuning like with long rule lists. Further,

machine learning algorithms are efficient at solving complex problems that may contain a lot of data or need to be able to adapt to new data (Mitchell, Machine Learning, 1997).

Machine learning solutions can be broken down into two types: supervised or unsupervised. Supervised machine learning relies on labeling data which is fed into the algorithm that includes the desired solution. It maps an input to an output based on supplied examples. It does this by analyzing a set of training data and produces an inferred function that can then be used for mapping new examples. From here, the algorithm can generalize from the supplied training data to unseen situations, which can be measured statistically by evaluating the generalization error (Mitchell, Artificial Intelligence: A Modern Approach, Third Edition, 1997).

Unsupervised machine learning is when the data is not tagged or labeled. The idea is that the algorithm will self-organize itself and capture patterns within the data and perform clustering strategies to make predictions. The focus of this thesis is on supervised learning, as OSINT data suits this strategy best (Géron).

For supervised learning, there are a wide variety of algorithms that can be chosen, each with their own strengths and weaknesses. No algorithm works best for all tasks and it is vital that different algorithms are considered and have their effectiveness evaluated to the problem at hand.

The most widely used supervised learning algorithms are:

- Support-vector machines
- Linear regression
- Logistic regression

- Naïve Bayes
- Linear Discriminant analysis
- Decision trees
- K-nearest neighbor algorithm
- Neural networks
- Similarity learning

Along with determining the best algorithm for a problem, there are other factors to consider. Bias-variance tradeoff is the process for determining the threshold based on training data sets to combat bias and variance. The prediction error is related to the bias summed with the variance. If an algorithm exhibits low bias, it may be more flexible, and it can fit each training set of data differently which causes high variance. Further, it is important to determine the amount of training data used which is to be considered in comparison with how complex the true function is. In other words, if the true function is simple, an inflexible (high bias) algorithm can learn from a small amount of training data. If the data is very complex and has many different behaviors or interactions, a large amount of data is required with a flexible algorithm (low bias) (Géron).

Noise also needs to be considered when developing a machine learning solution. If the output has low accuracy (often incorrect) then the algorithm should not find a function that exactly matches the training examples. If the data is fit too closely it causes overfitting. This can be caused if there are no errors, noise, when building your learning algorithm.

A key part of a supervised approach is feature engineering. This is the process of selecting the features within your data to train on. For example, if your data is animals, a feature would be "type of animal" which could equal "cat". Your features should be relevant to the task at hand. Features can be extracted which is the process of combining features to produce more useful complex features to train on (Stanford University, 2019).

Finally, machine learning solutions need to be tested and validated. To build a learning model, it is best to split your data into two sets, a training set and test set. A typical ratio is 80% training data and 20% test data; however, this can be tailored to fit the problem and algorithm. If there is a low training error and high generalization, overfitting is typically occurring (Géron).

Implementing a working machine learning algorithm can be challenging. Below is a typical framework to follow when designing a solution:

1. Frame the problem at hand
2. Collect data
3. Explore the data to gain insights
4. Prepare the data for use with ML algorithms
5. Explore and evaluate different learning models
6. Fine-tune the model
7. Evaluate the solution

Because of the nature of open-source intelligence (OSINT) for this thesis, natural language processing (NLP) must also be considered. NLP is used to understand structure and meaning of human language by analyzing large amounts of data, thus allowing the

computer to under the contents of a natural language data source. Through the analysis of syntax, semantic, pragmatics, and morphology, machine learning algorithms can be built to solve problems related to human language (Bird, 2009).

Natural language processing allows for many benefits to be realized through a machine learning solution. Including:

- Performing large scale analysis such as unstructured text data
- Automate processes in real-time
- Tailoring NLP tools to the task like criteria and industry-specific language

Natural language processing is performed by transforming text into associates that the model can understand. This process is called text vectorization. Statistical analysis is then used to build a dictionary to determine which features best represent the identified text. The more language data fed into a NLP algorithm, the better the analysis will be (Yse, 2019).

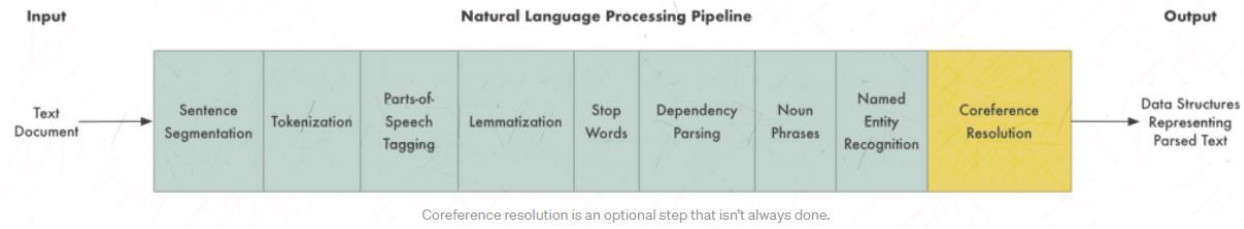
One of the key steps in NLP is tokenization. This process is conducted by breaking unstructured text into tokens that can then be used for model development. Through these tokens, text can be used in an objectified programmatic sense and fed into the solution. Another way of thinking of this is "building meaning" from the text.

Lastly, when dealing with large amounts of unstructured text it is important to perform certain procedures to make sure that the text being fed into the machine learning model is as clean as possible. In other words, we do not want to feed in unnecessary data into our model, which can skew the model's fit given the problem. Below are outlines of

steps to be performed in an NLP pipeline (Geitgey, 2018). The result are data structures representing text that are then fed into the machine learning algorithm.

- Sentence segmentation: break apart unstructured text into separate sentences.
- Tokenization: break sentences into separate words.
- PoS Tagging: guessing the part of speech (noun, verb, adjective).
- Lemmatization: identifying different word forms.
- Stop Word Removal: removing filler words (ex: "and", "the", "a").
- Dependency Parsing: how words relate to each other.
- Noun Phrases: grouping words that represent a single idea.
- Named Entity Recognition: label nouns or real-world objects.
- Coreference Resolution: (optional) identifying related words based on context.

FIGURE 2 NATURAL LANGUAGE PROCESSING PIPELINE (GEITGEY, 2018)



Part III

RedAI

3.0 REDAI

For this thesis we developed a working, real-world focused, solution that can test, evaluate, and answer the research questions that have been posed. This working solution has been named RedAI and was developed using the Python 3 programing language along with accompaniment from numerous Python libraries. By building a working and demonstrable product, this thesis is able to clearly articulate its findings in a dynamic and testable manner. Further, this method of research allows others to further develop the work and evaluate the findings themselves. Lastly, this solution allows these findings to be dynamic and not statically locked in time. The data used throughout the solution can be modified for different use-cases or time-related evolution.

Code for this work has been released under an open-source license and can be accessed at the GitHub repository:

https://github.com/lukeanoel/redAI/tree/main/redAI

3.1 CYBER THREAT INTELLIGENCE ARCHITECTURE

3.1.1 MITRE ATT&CK OVERVIEW

To realize CTI into RedAI's design, the MITRE ATT&CK framework, specifically MITRE ATT&CK Enterprise, was utilized as a foundation. MITRE ATT&CK can be leveraged as a taxonomy as well as a useful data repository. ATT&CK for Enterprise is an adversary model and framework for describing the actions an adversary may take to compromise and operate within an enterprise network. By utilizing the model, an organization can characterize and describe post-compromise behavior that has been observed. Information on adversaries has been gathered through MITRE

research as well as penetration testing and red teaming. The result is a knowledgebase that characterizes activities adversaries use against enterprise networks (The MITRE Corporation, 2021).

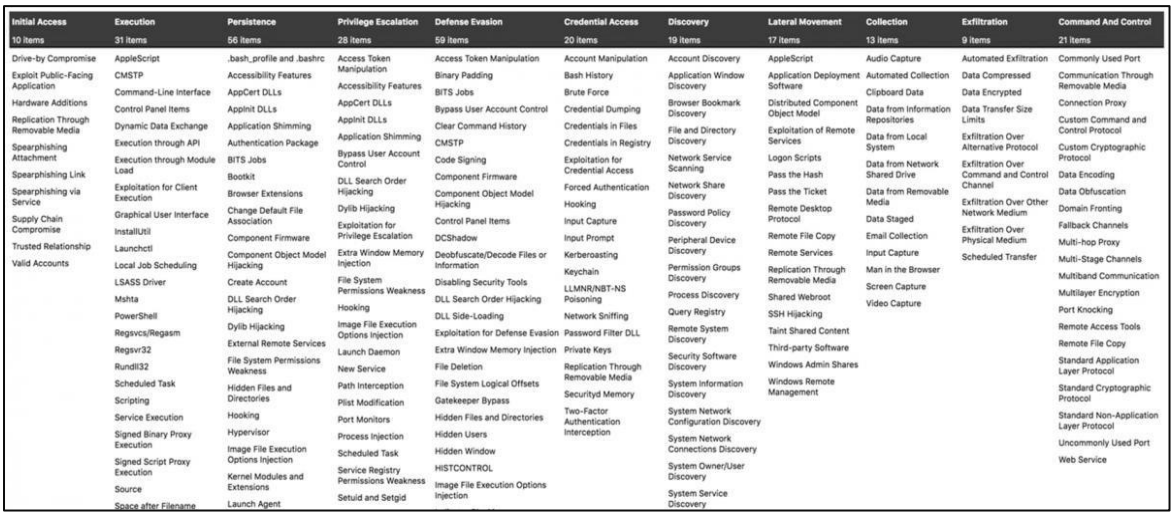
MITRE ATT&CK represents a large knowledge base of adversarial techniques which are offensively oriented actions performed by attackers. As outlined in previous sections of this thesis, the focus is on adversary behavior, not on low-level details like malware and tools themselves, but on how they are used within context. Techniques are organized into a set of tactics that explain each technique. Techniques provide details on how the technique works which provides context.

Tactics explain the "why" of a technique. This is the objective or goal the attacker has in-mind when performing an action. Tactics are high-level contextual categories. While techniques represent "how" an attacker achieves an objective (The MITRE Corporation, 2018).

One of the key features of ATT&CK that is used in this thesis is the relationship between tactics and techniques. This relationship is visually demonstrated via the ATT&CK Matrix.

Example: The tactic Persistence contains techniques including Account Manipulation, BITS Jobs, and External Remote Services. Each one of these techniques can be used to achieve Persistence.

FIGURE 3 MITRE ATT&CK MATRIX



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
10 items	31 items	56 items	28 items	59 items	20 items	19 items	17 items	13 items	9 items	21 items
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Password Policy Discovery	Pass the Ticket	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Permission Groups Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	InstallUtil	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Input Capture	Process Discovery	Remote Services	Input Capture		Multi-hop Proxy
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Prompt	Query Registry	Replication Through Removable Media	Man in the Browser		Multi-Stage Channels
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files or Information	Kerberoasting	Remote System Discovery	Shared Webroot	Screen Capture		Multiband Communication
	LSASS Driver	Create Account	Image File Execution Options Injection	Disabling Security Tools	Keychain	Security Software Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning	System Information Discovery	Taint Shared Content			Port Knocking
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing	System Network Configuration Discovery	Third-party Software			Remote Access Tools
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL	System Network Connections Discovery	Windows Admin Shares			Remote File Copy
	Regsvr32	File System Permissions Weakness	Plist Modification	Extra Window Memory Injection	Private Keys	System Owner/User Discovery	Windows Remote Management			Standard Application Layer Protocol
	Rundll32	Hidden Files and Directories	Port Monitors	File Deletion	Replication Through Removable Media	System Service Discovery				Standard Cryptographic Protocol
	Scheduled Task	Hooking	Process Injection	File System Logical Offsets	Securityd Memory					Standard Non-Application Layer Protocol
	Scripting	Hypervisor	Scheduled Task	Gatekeeper Bypass	Two-Factor Authentication Interception					Uncommonly Used Port
	Service Execution	Image File Execution Options Injection	Service Registry Permissions Weakness	Hidden Files and Directories						Web Service
	Signed Binary Proxy Execution	Kernel Modules and Extensions	Setuid and Setgid	Hidden Users						
	Signed Script Proxy Execution	Launch Agent		Hidden Window						
	Source			HISTCONTROL						
	Space after Filename			Image File Execution Options Injection						

Further, techniques can be broken down into sub-techniques which describe how, in more detail, the behaviors are performed. For the sake of this thesis, there were no distinctions made between techniques and sub-techniques, all techniques were treated equally.

3.1.2 ATT&CK & CTI INTEGRATION

One of the key benefits of using ATT&CK as a framework is its ability to integrate with cyber threat intelligence (CTI). ATT&CK's repository contains documented information on adversary group behavior profiles which are developed from open-source intelligence. Groups are then further documented by how they have used techniques in real-world scenarios to achieve their goals. These instances are classified as procedures. Analyzing procedures allows proactive defense by seeing how adversaries performed in the real-world, which allows organizations to trace their steps and implement risk management (Ghaith Huasri, 2017).

By integrating a solution with ATT&CK, we can make use of its data repository of cyber threat intelligence. Within this repository, is a large collection of information following the ATT&CK Matrix. MITRE makes all this information openly accessible via their website or by accessing it through their maintained GitHub repository. This information has been collected by MITRE from many different sources and centralized into one location. This information has been analyzed, vetted, and tied together under the wholistic intelligence approach (The MITRE Corporation, 2021).

This allows for anyone to be able to query the ATT&CK repository for a specific adversary group (or all groups) and find the techniques they have used throughout their observed actions, as well as the names of tools and malware they have used. Further, each one of these breakdowns provides a detailed description of how the group specifically used the technique or tool within that identified scenario. These descriptions have great value and serve as the foundation for this thesis's work.

For example, MITRE has defined Group APT1 as having a description of: **"APT1 is a Chinese threat group that has been attributed to the 2nd Bureau of the People's Liberation Army (PLA) General Staff Department's (GSD) 3rd Department, commonly known by its Military Unit Cover Designator (MUCD) as Unit 61398."**

Further, ATT&CK then contains the 20 different types of techniques that APT1 has been observed using. These techniques then give a detailed description tailored to APT1's use case. An example of how APT1 has used the technique System Service Discovery in the real-world: **"APT1 used the commands net start and tasklist to get a listing of the services on the system."**

ATT&CK also contains the known types of software that groups have been found to be using. Software consists of tools as well as malware. Each piece of software within ATT&CK also has its own description so that the consumer can get a better description of the software. More importantly, ATT&CK maps the techniques that each piece of software realized tailored to the specific group. That is, software can realize many different techniques, but for cyber threat intelligence it is important to know how each group specifically uses the software.

RedAI utilizes the following ATT&CK concepts: Techniques, Mitigations, Groups, and Software. Tactics are not considered as they are too high level for any real analysis.

ATT&CK consists of several different concepts to objectify its data repository. Below is a table outlining those concepts. Each concept has a respective ID variation that is used to identify concepts. This is ID is unique to each object within a concept (The MITRE Corporation, 2021).

TABLE 4 MITRE ATT&CK CONCEPTS

ATT&CK Concept	Description	ID Format
Technique	Represent "how" an adversary achieves a tactical objective by performing an action	Txxxx
Mitigation	Represent security concepts and classes of technologies that can be used to prevent a technique or sub-technique from being successfully executed	Mxxxx
Group	Known adversaries that are tracked by public and private organizations and reported on in threat intelligences reports	Gxxxx
Software	Represent an instantiation of a technique or sub-technique. Software is broken out into two high-level categories: tools and malware.	Sxxxx

3.1.3 Structured Threat Information Expression (STIX)

Structured Threat Information Expression (STIX) is a method in which to share cyber threat information. Historically, sharing information has proven to be inefficient and unstandardized. Information exchanged between providers and consumers is often inconsistent and expresses little detail and flexibility. Because of these factors, it is often hard to consume the information and integrate it into existing systems or solutions. Because of this, the industry has relied heavily on unstructured text exchanges for intelligence sharing, which has obvious difficulties for machines to understand. STIX aims to enable widespread consumption of information by providing common mechanisms for addressing threat information (OASIS, 2017).

STIX, by design, is an openly developed language for the characterization and communication of standardized cyber threat information. Through this language, cyber threats can be analyzed, specified, and shared in a unified fashion. STIX's design allows a large set of threat information to be collected, including:

- Cyber observables
- Indicators
- Incidents
- Adversary TTPs
- Exploit targets
- Courses of Action (remedies or mitigations)
- Cyber Attack Campaigns
- Cyber Threat Actors (adversaries or groups)

STIX Domain Objects (SDOs) allow for the actual relationship building and sharing of cyber threat intelligence. Each SDO contains property and relationship information. Property information includes common properties (type, ID, labels, etc.) and relationship information includes embedded relationships to other SDOs (OASIS, 2017).

STIX version 2.1, which was used for the work in this thesis, defines 18 STIX Domain Objects (SDOs).

TABLE 5 STIX DOMAIN OBJECTS

Name	Description
Attack Pattern	A type of TTP that describe ways that adversaries attempt to compromise targets.
Campaign	A grouping of adversarial behaviors that describes a set of malicious activities or attacks (sometimes called waves) that occur over a period of time against a specific set of targets.
Course of Action	A recommendation from a producer of intelligence to a consumer on the actions that they might take in response to that intelligence.
Grouping	Explicitly asserts that the referenced STIX Objects have a shared context, unlike a STIX Bundle (which explicitly conveys no context).
Identity	Actual individuals, organizations, or groups (e.g., ACME, Inc.) as well as classes of individuals, organizations, systems or groups (e.g., the finance sector).
Indicator	Contains a pattern that can be used to detect suspicious or malicious cyber activity.
Infrastructure	Represents a type of TTP and describes any systems, software services and any associated physical or virtual resources intended to support some purpose (e.g., C2 servers used as part of an attack, device or server that are part of defense, database servers targeted by an attack, etc.).
Intrusion Set	A grouped set of adversarial behaviors and resources with common properties that is believed to be orchestrated by a single organization.
Location	Represents a geographic location.
Malware	A type of TTP that represents malicious code.
Malware Analysis	The metadata and results of a particular static or dynamic analysis performed on a malware instance or family.
Note	Conveys informative text to provide further context and/or to provide additional analysis not contained in the STIX Objects, Marking Definition objects, or Language Content objects which the Note relates to.
Observed Data	Conveys information about cyber security related entities such as files, systems, and networks using the STIX Cyber-observable Objects (SCOs).
Opinion	An assessment of the correctness of the information in a STIX Object produced by a different entity.
Report	Collections of threat intelligence focused on one or more topics, such as a description of a threat actor, malware, or attack technique, including context and related details.

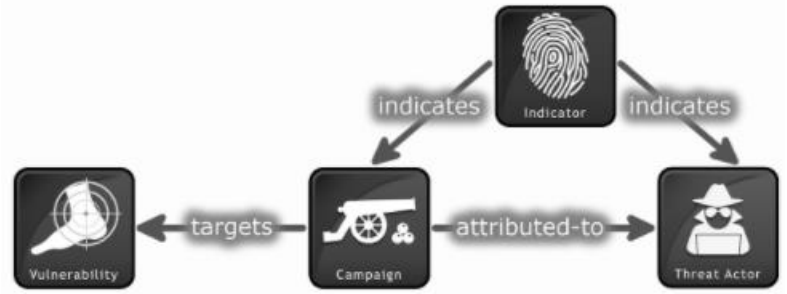
Threat Actor	Actual individuals, groups, or organizations believed to be operating with malicious intent.
Tool	Legitimate software that can be used by threat actors to perform attacks.
Vulnerability	A mistake in software that can be directly used by a hacker to gain access to a system or network.

TABLE 6 STIX DOMAIN OBJECT RELATIONSHIPS

Source	Type	Target	Source	Type	Target
attack-pattern	targets	vulnerability	intrusion-set	attributed-to	threat-actor
attack-pattern	targets	identity	intrusion-set	targets	identity
attack-pattern	uses	malware	intrusion-set	targets	vulnerability
attack-pattern	uses	tool	intrusion-set	uses	attack-pattern
campaign	attributed-to	intrusion-set	intrusion-set	uses	malware
campaign	attributed-to	threat-actor	intrusion-set	uses	tool
campaign	targets	identity	malware	targets	identity
campaign	targets	vulnerability	malware	targets	vulnerability
campaign	uses	attack-pattern	malware	uses	tool
campaign	uses	malware	malware	variant-of	malware
campaign	uses	tool	threat-actor	attributed-to	identity
course-of-action	mitigates	attack-pattern	threat-actor	impersonates	identity
course-of-action	mitigates	malware	threat-actor	targets	identity
course-of-action	mitigates	tool	threat-actor	targets	vulnerability
course-of-action	mitigates	vulnerability	threat-actor	uses	attack-pattern
indicator	indicates	attack-pattern	threat-actor	uses	malware
indicator	indicates	campaign	threat-actor	uses	tool
indicator	indicates	intrusion-set	tool	targets	identity
indicator	indicates	malware	tool	targets	vulnerability
indicator	indicates	threat-actor			
indicator	indicates	tool			

Below is an example visualization of a relationship between multiple SDOs. In this example, a Campaign is being attributed to a Threat Actor, both of which were indicated by an Indicator. That campaign then targets a Vulnerability.

FIGURE 4 IDENTIFYING A THREAT ACTOR STIX EXAMPLE (OASIS-OPEN, N.D.)



Below is an example of a specific attack-pattern form of spear phishing that references CAPEC. As shown, there are three SDOs, one for the attack-pattern, one for the intrusion-set (adversary), and one for the relationship connecting to the two (OASIS, 2017).

FIGURE 5 STIX ATTACK-PATTERN EXAMPLE


```
[
{
"type": "attack-pattern",
"id": "attack-pattern--7e33a43e-e34b-40ec-89da-36c9bb2cacd5",
"created": "2016-05-12T08:17:27.000Z",
"modified": "2016-05-12T08:17:27.000Z",
"name": "Spear Phishing as Practiced by Adversary X",
"description": "A particular form of spear phishing where the attacker claims that the target had won a contest, including personal details, to get them to click on a link.",
"external_references": [
{
"source_name": "capec",
"id": "CAPEC-163"
}
]
},
{
"type": "relationship",
"id": "relationship--57b56a43-b8b0-4cba-9deb-34e3e1faed9e",
"created": "2016-05-12T08:17:27.000Z",
"modified": "2016-05-12T08:17:27.000Z",
"relationship_type": "uses",
"source_ref": "intrusion-set--0c7e22ad-b099-4dc3-b0df-2ea3f49ae2e6",
"target_ref": "attack-pattern--7e33a43e-e34b-40ec-89da-36c9bb2cacd5"
},
{
"type": "intrusion-set",
"id": "intrusion-set--0c7e22ad-b099-4dc3-b0df-2ea3f49ae2e6",
"created": "2016-05-12T08:17:27.000Z",
"modified": "2016-05-12T08:17:27.000Z",
"name": "Adversary X"
}
]
```


3.1.4 STIX & MITRE ATT&CK

This section describes how this thesis programmatically uses MITRE ATT&CK's data repository by implementing and extending the STIX format. ATT&CK can be enabled by STIX by using some predefined STIX objects as well as creating some custom STIX objects. Below is a table mapping ATT&CK concepts to the STIX Object type used throughout this thesis's work. Note, that the STIX language version 2.1 contains more object types than mapped below. However, these are the only object types used throughout this thesis as they are the only ones utilized by ATT&CK's repository.

TABLE 7 ATT&CK TO STIX MAPPING

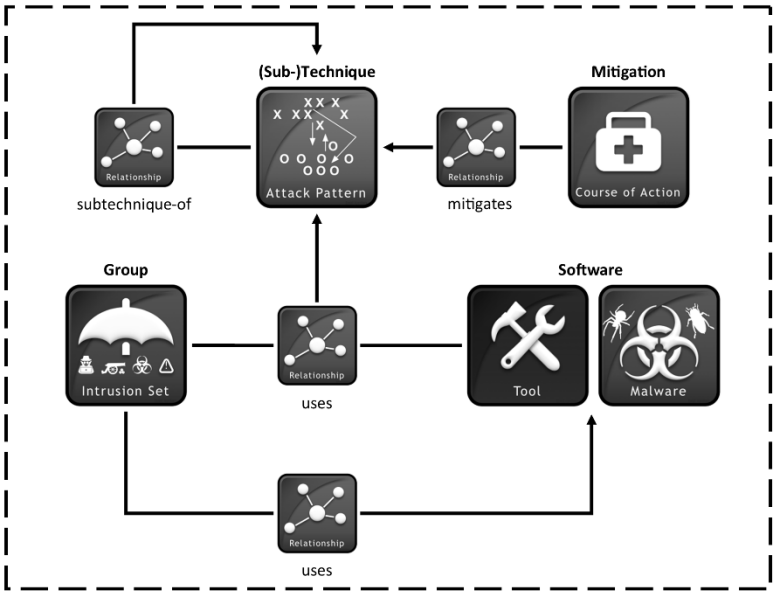
ATT&CK Concept	STIX Object Type
Matrix	X-mitre-matrix
Tactic	X-mitre-tactic
Technique	Attack-pattern
Sub-technique	Attack-pattern where x_mitre_is_subtechnique = true
Procedure	Relationship where relationship_type = "uses" and target_ref is attack-pattern
Mitigation	Course-of-action
Group	Intrusion-set
Software	Malware or Tool

While ATT&CK maintains unique IDs for each object in its repository, each STIX objects also possess a unique ID. These IDs are unique and allow for effective retrieval and reference to STIX objects programmatically. These IDs are different and need to be treated as such (The MITRE Corporation, 2021).

Relationships are the key component to relating ATT&CK objects to one another. This is performed by utilizing the STIX Relationship object. Relationships convey a wholistic picture of events and allow for a more informed intelligence posture. Below is an example of a relationship scenario. In this example, the relationships between a specific group, it is used tools/malware, the techniques performed, and the corresponding mitigations are shown (The MITRE Corporation, 2018). Within the box images are the name of the corresponding STIX object. Relationships cannot be revoked; however, they can be considered revoked if one of the objects they reference is revoked. This factor is important in the event groups, mitigations, techniques, or tools become deprecated and no longer form good intelligence by referencing them. Further, STIX Relationships allow for

a description field, which contains unstructured text information that provides context to the relationship between the objects (OASIS, 2017).

FIGURE 6 ATT&CK & STIX RELATIONSHIP VISUAL (THE MITRE CORPORATION, 2017)



3.2 OPEN-SOURCE INTELLIGENCE

For the work in this thesis, OSINT is focused on the collected material within MITRE ATT&CK's data repository. This repository is globally accessible to anyone and can be viewed via their website. This intelligence has been gathered for use by the private sector, government, and cybersecurity product and service community (FireEye, 2021). This information is routinely maintained and kept up to date. Further, MITRE strives to make sure the information and resulting intelligence is of high quality and centralized to the cyber security field, which is typically is not common with other OSINT sources (Ang, 2020).

MITRE's intelligence within ATT&CK is a result of much of their own efforts and research as well as dozens of different well-known entities within the cyber security industry. With these factors in mind, MTIRE and ATT&CK are highly beneficial OSINT resources to take advantage of for this work.

3.2.1 Collection

MITRE makes the ATT&CK data repository globally accessible by anyone. This information can be viewed manually via their website which also allows for the possibility for web scraping. MITRE also makes all their data accessible via a maintained GitHub repository. Through this repository they have all the data categorized by ATT&CK concept in STIX format which can be easily downloaded via JSON files. These JSON files contain the bundle of STIX SDOs which contain all needed information for the work in this thesis (descriptions, names, ATT&CK IDs).

For the work of this thesis, the JSON files were downloaded and integrated into the working RedAI solution. The following table breaks down the figures for data obtained from the data repository (MITRE ATT&CK Enterprise, 2021).

Table 8 ATT&CK Data Collected

ATT&CK Category	Data Objects Collected
Groups	112
Mitigations	208
Techniques	392
Software	428
Relationships	1349

The ease it which it is to collect this amount of valuable intelligence highlights the benefits of working with ATT&CK. This large amount of collection was performed seamlessly while following industry standardized formats that can be flexibly integrated into the solution.

3.2.2 CONSUMPTION

With the cyber threat intelligence collected in STIX format, programmatic methods were developed to convert this format into more usable data constructs to integrate with machine learning libraries. To do this, each ATT&CK object's corresponding STIX SDO was processed, with desired fields extracted. After this process, fields were extracted from the STIX SDOs that were required for this work. These fields include ATT&CK type, ATT&CK name, and description.

After extraction, all the data is stored in CSV format since most machine learning libraries take CSVs as input.

3.3 A MACHINE LEARNING SOLUTION

There were numerous key libraries utilized throughout this work to provide machine learning and natural language processing functionality. These key libraries are highlighted in Table 9.

TABLE 9 PYTHON LIBRARIES UTILIZED (SCIKIT-LEARN) (PANDAS) (NLTK PROJECT, 2021)

Library	Description
SciKit Learn	Scikit-learn is an open source machine learning library that supports supervised and unsupervised learning. It also provides various tools for

	model fitting, data preprocessing, model selection and evaluation, and many other utilities.
Pandas	Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.
NLTK	NLTK (Natural Language Toolkit) is a leading platform for building Python programs to work with human language data.

3.3.1 Approach

With the intelligence now collected, and the cyber threat intelligence architecture understood, the next step is developing a solution to demonstrate the research. The issue at hand is being able to demonstrate how machine learning can be utilized to effectively classify cyber threat intelligence. To get to this result, certain qualities of the intelligence need to be analyzed.

Each ATT&CK object obtained contains a type (Group, Technique, etc.) along with an unstructured text form description. This description is going to be the key feature used to build the machine learning models. These descriptions define each ATT&CK object in detail and link them to a real-world observation (MITRE ATT&CK Enterprise, 2021).

Figure 7 Example Group Description

[APT19] is a Chinese-based threat group that has targeted a variety of industries, including defense, finance, energy, pharmaceutical, telecommunications, high tech, education, manufacturing, and legal services. In 2017, a phishing campaign was used to target seven law and investment firms. Some analysts track and [Deep Panda] as the same group, but it is unclear from open source information if the groups are the same.

Figure 8 Example Malware Description

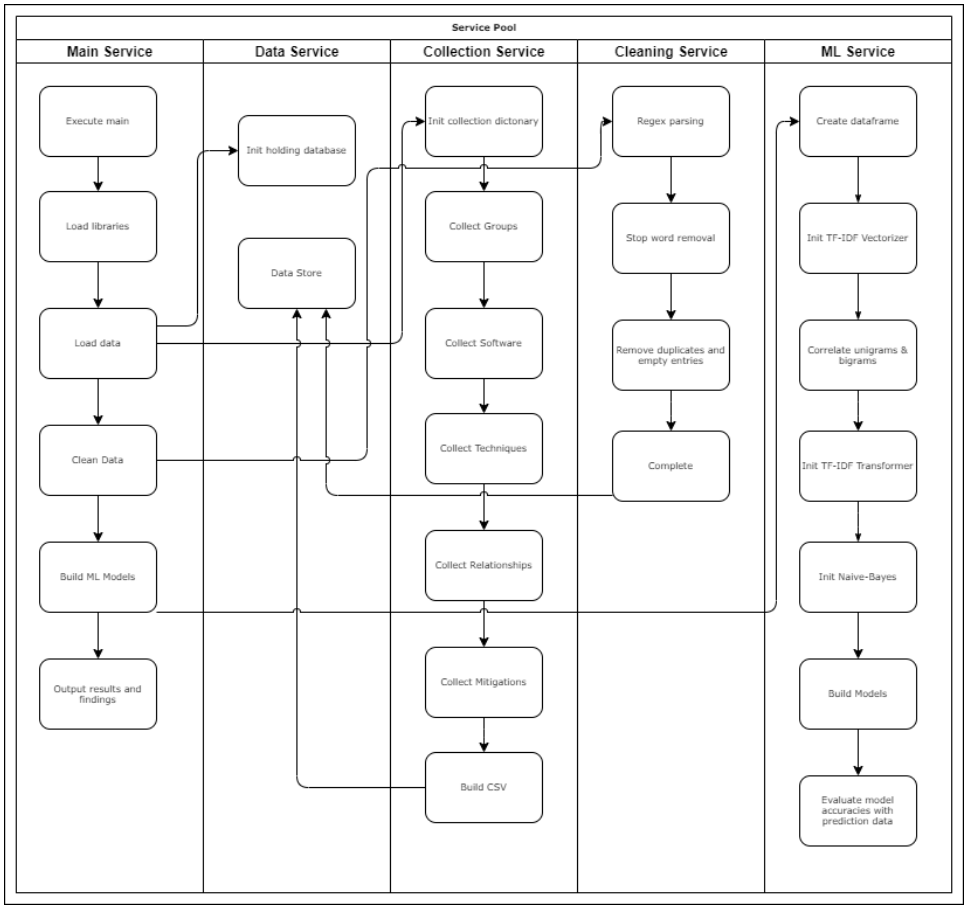
[CARROTBAT] is a customized dropper that has been in use since at least 2017. has been used to install and has infrastructure overlap with.

The goal of the machine learning solution is to be able to take in cyber threat intelligence data, and by analyzing the unstructured text of their descriptions, be able to accurately classify what type of ATT&CK object the intelligence falls under.

3.3.2 System Design Overview

Figure 9 shows the system design overview highlighting the entire pipeline of RedAI. This view shows how RedAI utilizes a microservice architecture approach to achieve its execution. Further, this overview shows the roles of each service as well as their placement in the process chain.

FIGURE 9 REDAI PIPELINE



3.3.3 Data Evaluation

A great deal of unstructured text data has been gathered from the MITRE ATT&CK repository. Evaluating this data is a key step and being able to prepare it efficiently to be used to build the machine learning models. Further, it is important for any machine learning solution to fully understand what your data represents and contains. Data is the foundation of any machine learning solution, and without a good grasp of the data the solution is set up for failure.

Table 10 ATT&CK Features of Focus

ATT&CK Name	ATT&CK Type	Description
The name of the ATT&CK object	The classification of the ATT&CK object (Group, Software, etc.)	The unstructured text field explaining contextual information about the ATT&CK object. This field can be just a sentence to multiple paragraphs.

This data evaluation process is a key enabler of the data cleaning process to be performed. By analyzing the data obtained, we can see that there are many unwanted pieces of text data. Figure 8 is an example of a Group's (Axiom) description extracted from the ATT&CK repository in raw form of text with spaces removed.

Figure 10 Group Description Example

[Axiom](https://attack.mitre.org/groups/G0001)isacyberespionagegroupsuspectedtobeassociatedwiththeChinesegovernment.ItisresponsiblefortheOperationSMNcampaign.(Citation:Novetta-Axiom)Thoughbothisgroupand[WinntiGroup](https://attack.mitre.org/groups/G0044)usethemalware[WinntiforWindows](https://attack.mitre.org/software/S0141),thetwogroupsappeartobedistinctbasedondifferencesinreportingonthegroups'TTPsandtargeting.(Citation:KasperskyWinntiApril2013)(Citation:KasperskyWinntiJune2015)(Citation:NovettaWinntiApril2015)

This example is a good representation for all the raw unstructured text that was collected by ATT&CK. As can be seen, there are many characteristics of this data that are not of value for building the ML models. For starters, URL reference links to either MTIRE's website pages or to external references do not provide contextual value. While important to know the sources of data, it does not provide any machine learning tokenization value. Next, any references to citations are not needed for the same reasons. Lastly, it was noticed that same malware descriptions have code samples or references. The unstructured text denotes this code using HTML tags like <code>. These tags are not valuable, however the text inside the tags is. Lastly, throughout this work it was deemed necessary to convert all text to Unicode UTF-8, as some of the data pulled in was in an improper format.

Another important realization was that some ATT&CK objects collected had empty descriptions. These provide no value and are a hindrance to the solution, as the remaining solution pipeline is expecting for there to be unstructured text information. Other key evaluation notices are that there can sometimes be unstructured text references to the group itself, other groups, other aliases, or possible malware used. These are acceptable to leave in the unstructured text as they provide contextual information that will help build better models for that specific ATT&CK type.

3.3.4 Data Cleaning & Preparation

After the data collected from MITRE ATT&CK has been evaluated appropriately, the data can now be cleaned and prepared to build models with. For this approach, the entire MITRE repository will be used to develop training and test sets of data. The training data set will be used to train the model. The test data set will be the means to test

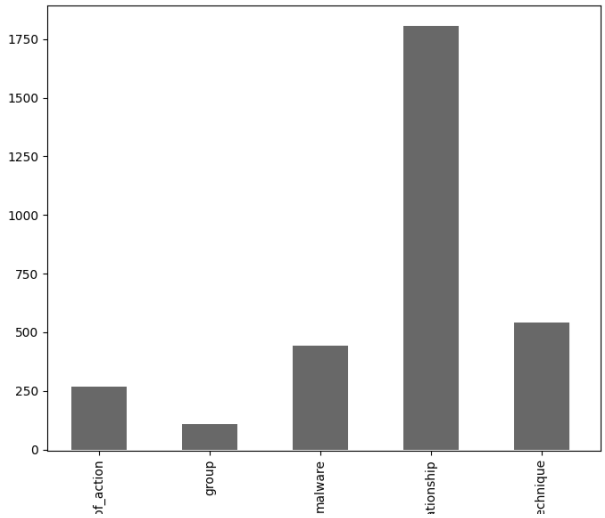
the accuracy of the model. Currently, it is not important to know the training and test ratios, but it is important to know where those sets are coming from. Further, it is important to know that when creating those data sets, the sets will be created using a random approach, so that all data is treated equally.

To begin cleaning the data, RedAI iterates through the stored data checking for empty description fields as well as duplicates. If any of these are found, they are removed. The next step is to utilize regular expressions (regex) to remove all URLs, citation references, and ATT&CK references. Also using regular expressions, all possible <code> HTML tags are removed as well as new line and indentation tabs like '\n' and '\t'. After all this is complete, the text is converted to Unicode UTF-8 format. After this process is complete, RedAI is left with clean looking data that is intelligence focused.

After this initial cleaning, more general machine learning cleaning methods are applied. To begin, all words are converted to lowercase, all punctuation is removed, and stop words are removed. Stop word removal is important as stop words provide no added value to the sentences themselves but are only needed to formulate English language continuity. These are common words such as "a", "an", "the", and "in". The entire list of stop words used for this work was referenced by the NLTK's list of English stop words.

After all this cleaning has been performed across the data, the data is stored for later use to build the machine learning models. Below is a figure plot showing the totals of each ATT&CK category.

FIGURE 11 ATT&CK DATA OBJECTS COLLECTED



3.3.5 MODEL DEVELOPMENT

To be able to answer the research questions effectively, this thesis places a strong focus on evaluating different strategies throughout the development of the machine learning solution. These strategies include the train/test ratio, NLP procedures, and the type of algorithm deemed the best fit.

Model development begins by following a bag of words approach towards our data. Bag of words is taking in each document of unstructured text and ignoring the placement of the words and placing an emphasis on frequency. For this work, Term Frequency Inverse Document Frequency (TF-IDF) is used. TF-IDF is a statistic to reflect on the importance of a word within a collection. TF-IDF value increases as a word appears more often in a document but is offset by the number of documents that contains the word. This helps adjust for words that generally appear more frequently across the whole spectrum (Rajaraman, 2011).

FIGURE 12 TERM FREQUENCY FORMULA

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

A TF-IDF vector is calculated using the Scikit-Learn library with the following properties:

- Sublinear_df set to True to use logarithmic frequency
- Min_df is the minimum amount of documents a word must be present in to be kept, which is set to 3
- Norm is set to 12, ensuring all feature vectors have a euclidian norm of 1.
- Ngram_range is set to (1,2), this allows for consideration of both unigrams and bigrams

FIGURE 13 TF-IDF VECTORIZER

```
TfidfVectorizer(sublinear_tf=True,min_df=3,norm='l2',encoding='latin-1',ngram_range=(1,2))
```

The next step in the process is to transform the Description feature of each ATT&CK object into a vector of numbers. This assists with training supervised classifiers. This gives a vector representation of the text, which allows us to train unseen ATT&CK objects to make predictions on which ATT&CK type they fall under. After this, everything has been prepared to develop models and evaluate their effectiveness.

3.3.6 Model Evaluation

The next step in model development is to benchmark different types of models and evaluate their accuracy. The four models that will be benchmarked are:

- Multinominal Naïve Bayes
- Logistic Regression
- Linear Support Vector Classification
- Random Forest

This evaluation process included splitting the original ATT&CK dataset into the training and test sets, developing the model using the SciKit-Learn Python library, and evaluating the accuracy between all the models. Further, a combined boxplot and strip plot approach was used for each ATT&CK category to determine the accuracy focuses on the category per model. The accuracy is determined by feeding in the description features of the test data set into each model, knowing where they belong, and seeing how accurate the model classifies them (Géron).

For high-level analysis, it is sometimes important to analyze the most common unigrams and bigrams for each type of classification within the training set. This gives an overview of some common features, but also would indicate issues in data collection or cleaning if there are noticeable features that are not of value.

Table 11 Most Common Unigrams & Bigrams

ATT&CK Type	Most Correlated Unigrams	Most Correlated Bigrams
Group	Threat Group	Group targeted Threat group

Software	Trojan Backdoor	Access tool Backdoor used
Relationship	Adversaries Legitimate	Threat group Backdoor used
Technique	Adversary Adversaries	Adversaries use Adversaries abuse
Mitigation	Applocker Privilege	Whitelisting software Potentially malicious

FIGURE 14 GROUP BOX PLOT

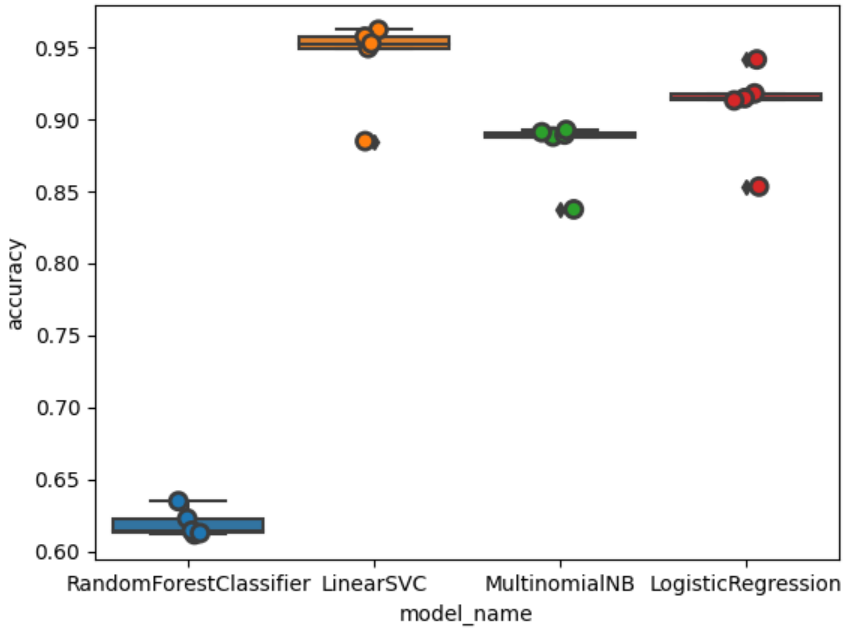


FIGURE 15 MITIGATIONS BOX PLOT

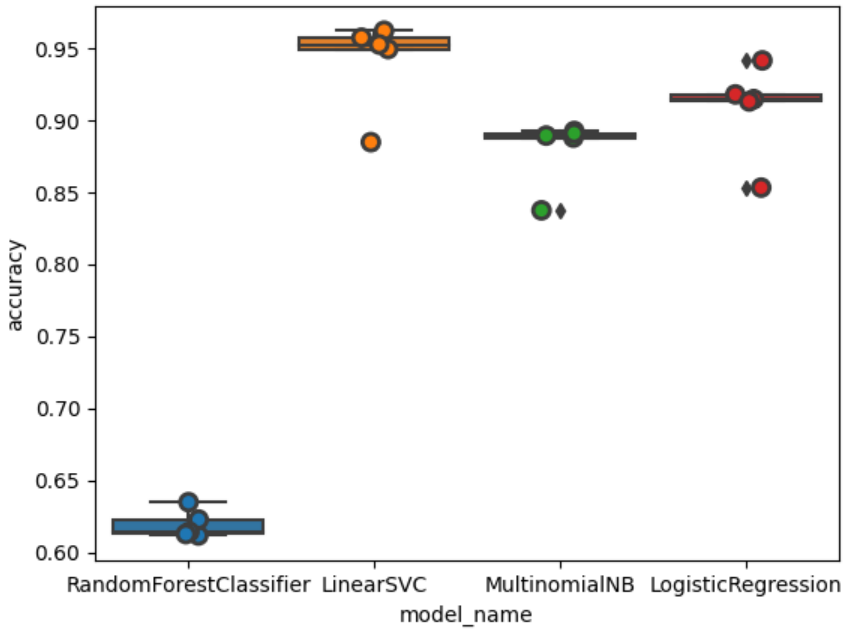


FIGURE 16 RELATIONSHIPS BOX PLOT

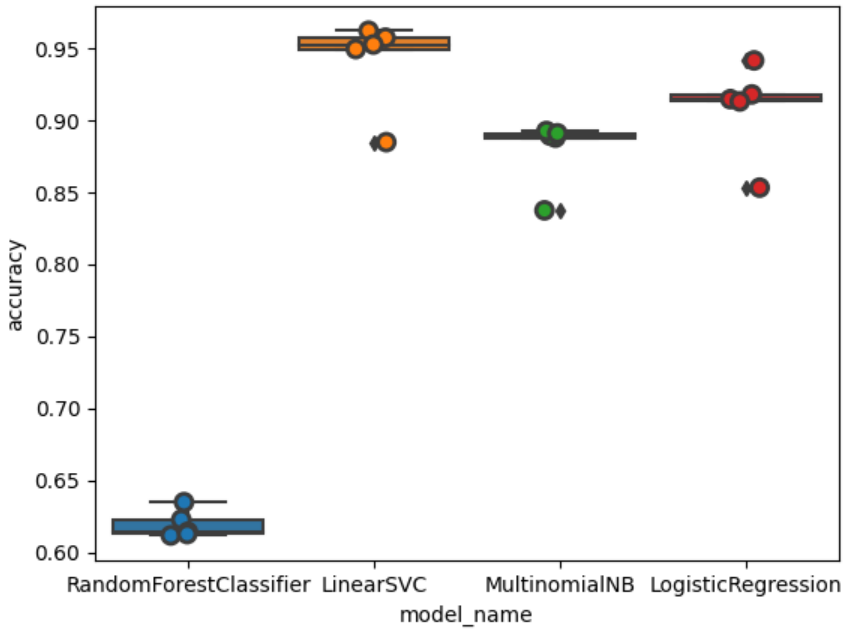


FIGURE 17 SOFTWARE/MALWARE BOX PLOT

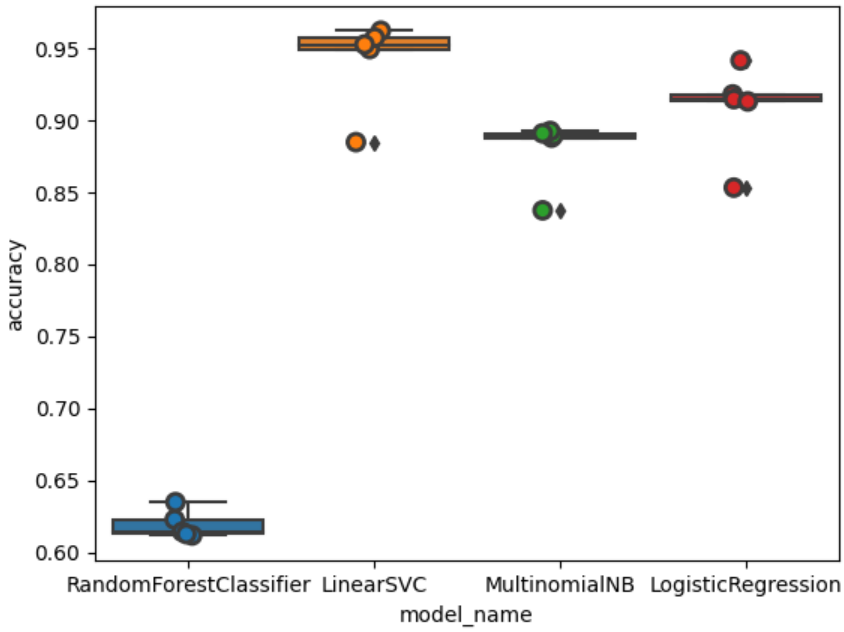
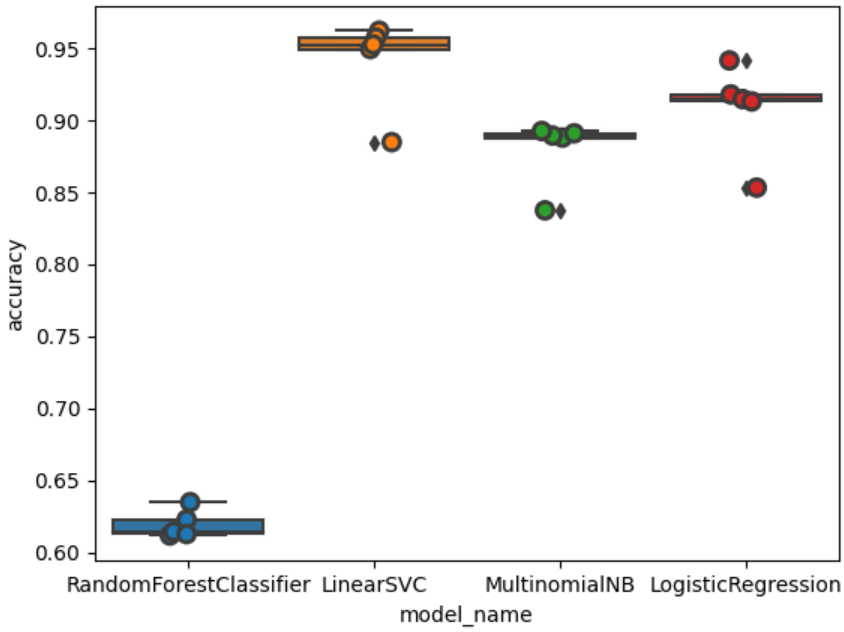


FIGURE 18 TECHNIQUES BOX PLOT



As can be identified from the box plots shown above, the variation in accuracy between each ATT&CK category is very limited, meaning that the mean accuracy for each model for each ATT&CK category show no indication of certain categories creating outlying results within certain models. This is as expected as the structure of data is unstructured text, and each category followed the same text cleaning procedures.

To begin, the Naïve Bayes model was evaluated. Naïve Bayes is a family of probabilistic classifiers based on Bayes' theorem. This classifier places a significance on the independence between features (McCallum, 2019). For this work, the Multinominal Naïve Bayes approach was used. As can be seen, Naïve Bayes performed similarly to Logistic Regression except at a slightly lower mean accuracy of .831703.

Next, Random Forest algorithm was evaluated. Random Forest operates by constructing multiple decision trees and then classifying the data and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees (TK, 1998). This model performed the weakest of all the algorithms; however, it did have the less variance in accuracy with no outliers.

Following, Logistic Regression was evaluated. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression estimates the parameters of a logistic model (Tolles, 2016). This model performed well at an accuracy of over 90%. However, as can be seen by the box plots, it had a weakness to outliers.

Lastly, Linear Support Vector Classification (SVC) was evaluated. Linear SVC is similar to SVC, but it is implemented in terms of lib-linear, giving it more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples (Scikit-Learn). As can be noticed, Linear SVC had the highest mean accuracy of all the models and had a weakness to only one noticeable outlier for each ATT&CK object.

After performing these evaluations, Linear SVC was chosen as the best performing model to use for this research. All work performed here after is under the use of the Linear SVC model algorithm.

TABLE 12 MODEL ACCURACIES

Model Name	Mean Accuracy
LinearSVC	.936542
LogisticRegression	.908763
MultinomialNB	.831703
Random Forest	.695300

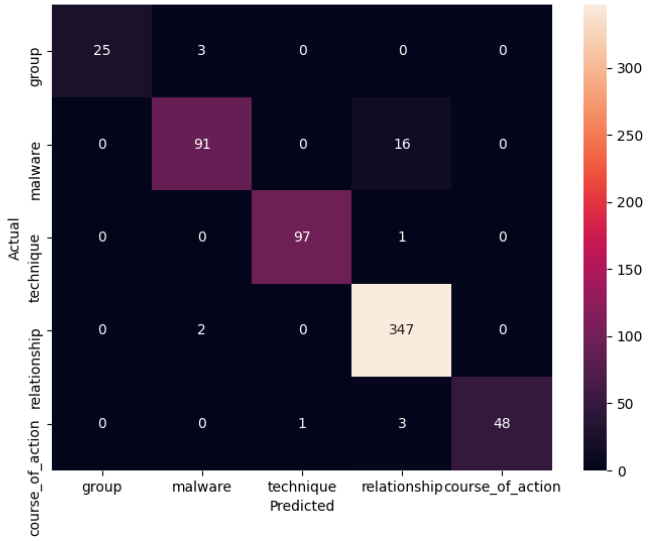
3.3.7 SOLUTION EVALUATION

While the previous section analyzed the results of our machine learning implementation, it is important to utilize this result and compare it to the wholistic problem at hand. In this section, the results will be analyzed at a lower level, still leaving the larger conclusion to be presented in a later section.

As can be seen in the figure below, most of the predictions made by our model visualize a linear heat map, which is the desired outcome. This shows that most of our

predictions are being accurately categorized with our model. However, as can be seen throughout there are some misclassifications.

FIGURE 19 MODEL PREDICTION ACCURACY HEATMAP



The only ATT&CK category with many misclassifications are the number of predictions that are being classified as relationships when in reality they are malware. In this case, it can be useful to analyze the descriptions of these malware types and try to make assumptions as to why they may have been misclassified.

TABLE 13 MALWARE MISCLASSIFICATIONS

Actual ATT&CK Type	Predicted ATT&CK Type	Description
Malware	Relationship	is malware that was injected into a signed ver...
Malware	Relationship	is a loader component that has been observed l..
Malware	Relationship	is an OS X trojan that relied on a valid devel...
Malware	Relationship	is a malware platform that uses a modular appr...
Malware	Relationship	steals banking information through man-in-the-...
Malware	Relationship	is a software suite and network that provides ...
Malware	Relationship	is a malicious DLL that has been used by duri…
Malware	Relationship	is an automatic SQL Injection tool distributed...
Malware	Relationship	is a malware-as-a-service offered on the darkw...
Malware	Relationship	is a loader crafted to be launched via abuse o...
Malware	Relationship	can be used to find or change information with...
Malware	Relationship	is a Trojan used by to remotely execute comma...
Malware	Relationship	is a kernel-mode that attempts to add victims...
Malware	Relationship	is a Web shell. It has been modified by actor…
Malware	Relationship	is a signature backdoor used by that is capab...
Malware	Relationship	is a kernel-mode rootkit used for cryptocurren...

Certain assumptions can be made after reviewing the text that was misclassified, but none of these can be for certain. First, these descriptions have two or one subjects, which is a common feature that can be expected from relationships. Next, these typically are more text heavy and have a connection between what the malware does and how it has been used in the real world, again another feature of what we expect from relationship descriptions.

Lastly, it is important to evaluate the final scores from all the ATT&CK types. Below is a classification report with the following analysis points: precision, recall, f1-score, and support. Precision shows that for each prediction for each ATT&CK type, how often did it pick the correct one. Notably, groups and courses of action were predicted with 100% precision, with the other types not far behind. Highly precise categories can most likely be attributed to how unique their description feature is compared to the other types. Recall is the percentage of all the types used for testing that were correctly placed in their right classification. All performed well, however malware preformed the worst at 85%. This can be attributed to how diverse in features malware descriptions were. This makes sense due to the many different types of malware as well as use cases.

Next to analyze are the f1-scores. F1-scores are the harmonic mean of the precision and recall. This can serve as a good indicator of a total score, which all the categories reflect 90% or above. Lastly, the support column is the number of instances used for each ATT&CK category for testing. As a reminder, our train/test ratio was 80:20, so most of our data went towards training our model.

TABLE 14 SOLUTION PREDICTION RESULTS

ATT&CK Type	Precision	Recall	F1-score	Support
Group	1.00	0.89	0.94	28
Malware/Software	0.95	0.85	0.90	107
Technique	0.99	0.99	0.99	98
Relationship	0.95	0.99	0.97	349
Course of Action/Mitigation	1.00	0.92	0.96	52

Further, the scores were compared with one another by calculating the averages. In total, 634 ATT&CK objects were used for testing the machine learning model, and we were met with highly accurate results with an average f1-score of 0.96. These scores are even further highlighted by the fact that the text for each ATT&CK type came from the same source, which typically results in less accurate results as the unstructured text can be bias to the author and hard to differentiate. Below is a table showcasing our results, along with the macro-average of the scores as well as a weighted average as the different types had different amounts of objects. Lastly, all these results can be easily replicated by anyone by utilizing the RedAI solution with this data or by experimenting with their own.

TABLE 15 WEIGHTED SOLUTION SUMMARY RESULTS

	Precision	Recall	F1-score	Support
Accuracy	n/a	n/a	0.96	634
Macro Average	0.98	0.93	0.95	634
Weight Average	0.96	0.96	0.96	634

Part IV

Conclusions & Future Work

4. CONCLUSIONS & FUTURE WORK

4.1 RESEARCH QUESTION 1

Research Question 1 - How can the MITRE ATT&CK framework serve organizations as a cyber threat intelligence (CTI) ontology?

The work in this thesis took a deep dive into the topic of cyber threat intelligence and the current landscape in which organizations can consume and share this intelligence. MITRE ATT&CK serves as a framework for describing post-compromise behavior that has been observed in the wild. By utilizing MITRE ATT&CK, organizations can take advantage of the research and intelligence gathering performed by a well-known organization and model it across their own enterprise.

ATT&CK provides an ontology that allows organizations to focus on high-level adversary behavior instead of low-level details. ATT&CK organizes techniques into sets of tactics that provide context to the situations. This allows organizations to pivot their focus from low-level defense maneuvers like IP blacklisting and instead utilize wholistic defense measures that are tailored to the "why" of an adversary attack.

ATT&CK also maintains strong intelligence on known adversary groups and their targeted industries. Organizations can analyze which adversary groups pertain to their organizations, link their known tactics and techniques, and then proactively tailor their cyber defense to those strategies. This method of defense allows organizations to be proactive, efficient, and intelligent.

4.2 RESEARCH QUESTION 2

Research Question 2 - Can open-source intelligence (OSINT) be a viable method for collecting and consuming cyber threat intelligence (CTI)?

Open-source intelligence (OSINT) is one of the most widely used forms of intelligence gathering. It also is one of the most available sources of intelligence and surrounds us both in the real and digital worlds. Being able to harness this vast array of information has many benefits and can be applied to a diverse set of solutions.

One of the key issues with OSINT is it can be represented in many different forms and is often not produced in formats that are easily consumable. Historically, OSINT needs to be manually sifted and analyzed by humans which typically causes it to be an inefficient and costly form of intelligence gathering. However, as the digital landscape and need for OSINT has evolved, so has the methods in which we can store, share, and consume it.

The Structured Threat Information Expression (STIX) language provides a common format for organizations to share and consume threat intelligence. STIX is open source and free, which naturally eliminates any barrier of entry for those who want to contribute and consume. All aspects of threat intelligence can be represented through STIX data objects and can be visually represented for an analyst or stored as JSON to be machine readable. This openness allows for organizations to integrate into tools and solutions easily. It is because of these principles that STIX proves to be an excellent enabler for OSINT.

While having STIX as an option for OSINT, it does not inherently prove that threat intelligence can still be effectively collected. This thesis proved this was possible by the ease at which intelligence was collected from MITRE's ATT&CK data repository. This data repository is the entirety of the ATT&CK intelligence framework all stored as STIX objects. By using STIX, ATT&CK was able to link all its threat intelligence research effectively to one another and easily consumable by anyone, globally. This thesis demonstrated the ease and effectiveness of the OSINT collection and serves as a demonstration of how organizations and entities can do the same.

4.3 RESEARCH QUESTION 3

Research Question 3 - Can machine learning prove to be an accurate and efficient method for harnessing the benefits of a cyber threat intelligence ontology?

Machine learning can be applied to solve many difficult and complex problems across a wide range of industries. These applications typically shine best when the problems at hand require a lot of manual intervention or analysis by a human. These are the current issues facing the cyber security landscape as there is such a large volume of data and not enough analysts to analyze and make decisions from the data. These factors are the recipe for a machine learning solution to be applied. However, this machine learning solution needs to be a tangible result for organizations to reach. A proper solution needs to be both accurate and effective for the problem at hand and needs to show added benefit over the current-day manual approach.

This thesis focused on the applicability of consuming unstructured OSINT threat intelligence and leveraging machine learning models to be able to classify unknown unstructured OSINT threat intelligence. This thesis was able to demonstrate extremely accurate results while also comparing and visualizing the difference in model types. Further, this solution demonstrated that a large amount of data proved to not be an issue for the models as this solution dealt with thousands of artifacts of data of various types and was able to predict accurately with an average F1-score of 95%. These working demonstrations show that organizations can leverage machine learning, as well as natural language processing, to efficiently classifying threat intelligence. Further, this thesis showed that a machine learning model can be fed in training data that is cyber threat intelligence specific. Even further, this solution's datasets were collected from data sources that package their intelligence from real-world events, further emphasizing their applicability. Lastly, the machine learning model, aligned with MITRE ATT&CK, showed that a cyber threat intelligence ontology can be applied to solving complex problems within the cyber defense landscape.

4.4 FUTURE WORK

One of the core themes and foundations of the work of this thesis is data. While it is important for organizations to have a strong understanding of cyber threat intelligence principles and frameworks, realizing these principles into tangible and beneficial solutions is a daunting task. One of the key features of building better solutions is developing larger and more robust data collection pipelines. It is vital that more and more data be collected with focus on sharing and consumption.

The work in this thesis can be further developed by utilizing more diverse and higher volumes of data. There is a growing amount of OSINT feeds being made available to the cyber security community. Examples of OSINT feeds include AlienVault's Open Threat Exchange (AlienVault, 2021), the FBI's InfraGard (Federal Bureau of Investigation, n.d.), and FireEye's Threat Intelligence solutions (FireEye, 2021). Further, custom tailored solutions can be developed as well by collecting intelligence using social media sites, dark web sites, and tools like Shodan (Shodan, 2021).

Collecting more cyber threat intelligence data will increasingly make the machine learning model smarter and able to predict more accurately. Next, even more diverse datasets can be consumed by the solution. Collecting ATT&CK types from a diverse pool of data streams will allow the machine learning model to be built from a stronger foundation and can be applied to more diverse problems.

Lastly, machine learning is one of the fastest growing fields within computer science. There are new libraries and strategies being released to the community which provide even more capabilities. It was not long ago that many of the ML libraries utilized in this thesis did not exist. As the machine learning field develops, so too can the models they produce.

REFERENCES

Air Force Institute of Technology. (2007). *Center for MASINT Studies and Research.*

AlienVault. (2021). *Open Threat Exchange*. Retrieved from https://otx.alienvault.com/

Ang, C. K. (2020). *Open source intelligence gathering and topic modelling on cyber security incidents.* Nanyang Technological University.

Association of Computer Machinery. (2016). Acing the IOC Game: Toward Automatic Discovery and Analysis of Open-Source Cyber Threat Intelligence. *SIGSAC Conference on Computer and Communications Security*, (pp. 755-766).

Barnum, S. (2014). *Strandardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX).* White Paper. Retrieved January 2021

Bird, S. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit.* O'Reilly Media; 1st edition.

Bromander, V. M. (2017). Cyber Threat Intelligence Model: An Evaluation of Taxonomies, Sharing Standards, and Ontologies within Cyber Threat Intelligence. *European Intelligence and Security Informatics Conference (EISIC)* (pp. 91-98). Athens, Greece: EISIC.

Crowdstrike, Kurt Baker. (2021, February 18). *What is Cyber Threat Intelligence?* Retrieved from Crowdstrike: https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/

Federal Bureau of Investigation. (n.d.). *Infragard*. Retrieved from infragard.org

FireEye. (2021). *Cyber Threat Intelligence Reports*. Retrieved from FireEye: https://www.fireeye.com/mandiant/threat-intelligence.html

FireEye. (2021). *Fiere Eye Threat Intelligence*. Retrieved from https://www.fireeye.com/mandiant/threat-intelligence.html

FireEye. (2021). *WHAT IS THE DIFFERENCE BETWEEN INFORMATION AND INTELLIGENCE?* Retrieved from FireEye: https://www.fireeye.com/mandiant/threat-intelligence/what-is-cyber-threat-intelligence.html

Geitgey, A. (2018, July 18). *How computers understand Human Language*. Retrieved from Medium: https://medium.com/@ageitgey/natural-language-processing-is-fun-9a0bff37854e

Géron, A. (n.d.). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media; 2nd edition.

Ghaith Huasri, E. A.-S. (2017). TTPDrill: Automatic and Accurate Extraction of Threat Actions. 2-7.

Gronberg, M. (2019). An Ontology for Cyber Threat Intelligence. Norway: University of Oslo.

Iriondo, R. (2018, October 15). *Machine Learning (ML) vs. Artificial Intelligence (AI) — Crucial Differences*. Retrieved from TowardsAI: https://pub.towardsai.net/differences-between-ai-and-machine-learning-and-why-it-matters-1255b182fc6

Kseib, N. (2019). Making Sense of Unstructured Threat Intelligence Data. IACB.

McCallum, A. (2019). *Graphical Models, Lecture 2: Bayesian Network Represention.*

McLaughlin, M. (2021). *Using open soure intelligence for cybersecurity intelligence.*

Mitchell, T. (1997). Artificial Intelligence: A Modern Approach, Third Edition. New York, New York.

Mitchell, T. (1997). *Machine Learning.* New York: McGraw Hill. ISBN 0-07-042807-7. OCLC 36417892.

MITRE ATT&CK Enterprise. (2021). *MITRE ATT&CK Enterprise*. Retrieved from https://github.com/mitre/cti/tree/master/enterprise-attack

NLTK Project. (2021). *NLTK*. Retrieved from https://www.nltk.org/

OASIS. (2017, February 24). STIX™ Version 2.0. Part 2: STIX Objects.

Oasis-Open. (n.d.). Retrieved from https://oasis-open.github.io/cti-documentation/examples/identifying-a-threat-actor-profile

Pandas. (n.d.). NumFOCUS. Retrieved from https://pandas.pydata.org/

Rajaraman, A. (2011). *Data Mining.* Retrieved from Rajaraman, A.; Ullman, J.D. (2011). "Data Mining" (PDF). Mining of Massive Datasets. pp. 1–17. doi:10.1017/CBO9781139058452.002. ISBN 978-1-139-05845-2.

Recorded Future. (2017, October 17). *Threat Intelligence: Difference Between Platforms and Providers*. Retrieved from Recorded Future: https://www.recordedfuture.com/threat-intelligence-platform/

Recorded Future. (2019, January 9). *How Artificial Intelligence Is Shaping the Future of Open Source Intelligence*. Retrieved from Recorded Future: https://www.recordedfuture.com/open-source-intelligence-future/

Recorded Future. (2019). *Understand Your Attacker: A Practical Guide to Identifying TTPs With Threat Intelligence.* Retrieved from Recorded Future: https://go.recordedfuture.com/hubfs/white-papers/identifying-ttps.pdf

Richelson, J. (2016). *The US Intelligence Community* (Vols. ISBN 978-0813349183).

Sahrom Abu, S. R. (2018). *Cyber Threat Intelligence – Issue and Challenges.* Universiti Teknikal Malaysia Melaka.

Scikit-Learn. (n.d.). Retrieved from https://scikit-learn.org/stable/

Shodan. (2021). *Shodan*. Retrieved from Shodan: shodan.io

Stanford University. (2019). *Machine Learning and AI via Brain simulations.*

The MITRE Corporation. (2017). *FINDING CYBER THREATS WITH ATT&CK-BASED ANALYTICS.* Whitepaper, McLean, VA. Retrieved December 7, 2020

The MITRE Corporation. (2018, July). MITRE ATT&CK: Design and Philosophy.

The MITRE Corporation. (2018). THREAT-BASED DEFENSE.

The MITRE Corporation. (2021). *MITRE ATT&CK*. Retrieved January 2021

The MITRE Corporation. (2021). *MITRE CTI Github*. Retrieved from GitHub: https://github.com/mitre/cti/

TK, H. (1998). *The Random Subspace Method for Constructing Decision Forests.* IEEE Transactions on Pattern Analysis and Machine Intelligence.

Tolles, J. (2016). *Logistic Regression Relating Patient Characteristics to Outcomes.*

Xiaojing Liao, K. Y. (2016). Acing the IOC Game: Toward Automatic Discovery and Analysis of Open-Source Cyber Threat Intelligence. 1-6.

Y. Ghazi, Z. A. (2018). A Supervised Machine Learning Based Approach for Automatically Extracting High-Level Threat Intelligence from Unstructured Sources. 129-134. Retrieved January 6, 2021

Yse, D. L. (2019, January 15). *Your Guide to Natural Language Processing (NLP)*. Retrieved from Toward Data Science: https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1