NLP-based Cyber Threat Data Analysis with MITRE ATT&CK Techniques

Haochen Zhang∗, Haoyang Hong∗, Binghan Shen∗, Marlon Esteban Brenes Rojas∗, Ezzeldin Tahoun†, Umme Zakia∗‡

*∗Department of Computer Science, New York Institute of Technology, Vancouver, BC V5M 4X5, Canada*

*†Cypienta, Calgary, Canada*

‡Corresponding author: Umme Zakia (e-mail: [uzakia@nyit.edu)](mailto:uzakia@nyit.edu)

*Abstract*— Recognizing and classifying attacks, threats and actors plays vital role in cyber threat intelligence (CTI). Analyzing potential threats correctly require adequate knowledge, experience and resources. Most enterprises do not have a clear classification plan after the cyberattacks to detect or prevent future threats. In this paper, we examined effective categorizing of the attacks through MITRE ATT&CK's classification to provide targeted protective measures. This article describes classifying cyberattacks via manual labeling and automated labeling process using a machine learning approach, i.e., the BiLSTM classifier, RF and LGBM model by describing them through the MITRE ATT&CK framework. Both the BiLSTM and the RF model obtained accuracies around 56%-62% while the LGBM classifier obtained higher accuracy of 99% in categorizing cyber threats and warnings using natural language processing. The ML approach can help faster automated labeling and creating good repository of MITRE ATT&CK for organizations and professionals in CTI process.

Keywords— cyber threat intelligence, MITRE ATT&CK technique, data scraping, alert manual labeling, machine learning, BLSTM, RF, LGBM,

# Introduction

Cyber threat intelligence (CTI) encompasses knowledge, skills, and experience-based insights regarding the occurrence and evaluation of both cyber and physical threats, as well as threat actors. This information aims to help mitigate potential attacks and harmful events in cyberspace. MITRE ATT&CK [1] is a free tool widely adopted by organizations across various sectors and industries, including both private and public entities. Its users range from security defenders, penetration testers to cyber threat intelligence teams focusing on building secure systems, applications, and services. The extensive information on attacks and attackers it provides can assist organizations in determining whether they are collecting the right data to effectively detect attacks and in evaluating the effectiveness of their current defenses. By adopting an attacker’s perspective, MITRE ATT&CK helps organizations understand how adversaries plan, prepare for, and execute attacks. This makes MITRE ATT&CK an excellent resource in the field of cybersecurity or threat intelligence, exploring attacks and mitigations. According to [1], “MITRE ATT&CK® is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations," and enterprise techniques represent “how an adversary achieves a tactical goal by performing an action.”. In [2], authors used machine learning (ML) techniques to classify malware with the MITRE ATT&CK and identify malware using malware threat data. Its labeling techniques are helpful in learning MITRE ATT&CK tactics. According to [3], authors used unsupervised labeling to establish a connection between MITRE ATT&CK Techniques and common vulnerabilities and exposures (CVE). In order to analyze cybersecurity-related documents, such as CVEs, books, research articles, and news, authors in [4] apply the natural language processing (NLP) model. In [5], authors use a bi-directional long term short memory (BiLSTM) classifier in order to identify cyber threats with high accuracy throughout the several stages of the MITRE ATT&CK framework. This article uses BiLSTM's capacity for sequential data analysis to improve cyber-attack detection. In [6], authors analyzed a combination of modern-day orchestrated security attacks using stacking of light gradient boosting machine (LGBM) and random forest with high prediction accuracy among a set of classifiers.

In this paper, we proposed a cybersecurity analysis framework that can identify security alerts and provide mitigation or prevention solutions. At first, alerts were collected from online resources and by scraping websites. Then we manually labeled them using MITRE framework. Finally, machine learning algorithms such as BiLSTM and LGBM classifiers were implemented for automating the mapping process of scraped attacks to MITRE tactics and techniques. This automated labeling used NLP as baseline cybersecurity analytics framework to identify security alerts and to provide recommended mitigation according to MITRE ATT&CK Enterprise Techniques.

The alerts and warnings used in this research were generated by threat analysis from different organizations’ attack data. Some alerts were also collected from scraping few websites. We created a Lookup Table (LUT) using regular expression for scraping websites and implemented a general method for associating every CVE with an appropriate MITRE ATT&CK approach to create label our dataset. In addition, as our dataset is text-based, we trained the NLP-based models for automated attack analysis. Unsupervised labeling was not considered because of unrelated attack dataset since security alerts are not that common among different sites and resources. Fig. 1 shows the overall framework of the proposed processes and their internal relationships in three phases:

Phase I: collecting alert and warnings data,

Phase II: manual labeling alerts to MITRE ATT&CK framework via lookup table, and

Phase III: auto-labeling alerts to MITRE ATT&CK framework via machine learning techniques.

This article is organized as below. Section II describes the methodology of alert and warning data collection via web scraping, extracting required data and storing the processed data. In Section III, the process of manual mapping of alerts to MITRE ATT&CK and labeling the warnings are described. Automated process of labeling using ML techniques are described in detail in Section IV. Section V have results and we conclude in Section VI with future work.

1. Data Sources

| Sources | Sources |
| --- | --- |
| Mitre Att&ck | https://attack.mitre.org/techniques/enterprise/ |
| Mitre Engenuity | https://attackevals.mitre-engenuity.org/results/enterprise |
| FortiSIEM Rules | https://help.fortinet.com/fsiem/Public\_Resource\_Access/7\_1\_0/rules/rule\_descriptions.htm |
| Snort | https://snort.org/downloads/#rule-downloads |
| Github Repository | * https://github.com/SigmaHQ/sigma/tree/master/rules * https://github.com/SecurityRiskAdvisors/TALR/blob/master/Rules/SRA/lateral\_movement/MSBuild\_Inv oked\_by\_WMI.yml * https://github.com/fugawi/mate/tree/master/tests * https://github.com/vadim-hunter/Detection-Ideas-Rues * https://github.com/mdecrevoisier/SIGMA-detection-rules/ * https://github.com/mdecrevoisier/SIGMA-detection-rules/ * https://github.com/P4T12ICK/Sigma-Rule-Repositor y/tree/master/detection-rules * https://github.com/endgameinc/eqllib/tree/master/eqllib/analytics * https://github.com/netevert/sentinel-attack/tree/master/detections * https://github.com/sbousseaden/EVTX-ATTACK-SAMPLES/ * https://github.com/Hestat/ossec-sysmon/tree/master * https://github.com/teoseller/osquery-attck * https://github.com/chronicle/detection-rules/tree/main * https://github.com/splunk/security\_content/ * https://github.com/OTRF/OSSEM-DM/tree/main * https://drive.google.com/file/d/1UrBPevQK12EpoRGpEcV7i0LE24xXcqT5/view?usp=sharing * https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1003.001/T1003.001.yaml * https://research.splunk.com/stories/ * https://research.splunk.com/detections/ * https://github.com/sublime-security/sublime-rules/blob/main/detection-rules/attachment\_any\_html\_unsol cited.yml * https://github.com/GoogleCloudPlatform/security-analytics/tree/main |



1. Overview of themaual and auto labelingprocess of alerts to Mitre Att&ck techniques.

# Data Collection: Alerts and Warnings

In this study, we started by collecting 2000 alerts and warnings from scraping several websites. We used MITRE ATT&CK framework to associate and label these warnings to appropriate techniques, and then extended the work to automate labelling using machine learning algorithm. The process of crawling website to obtain data included the following steps:

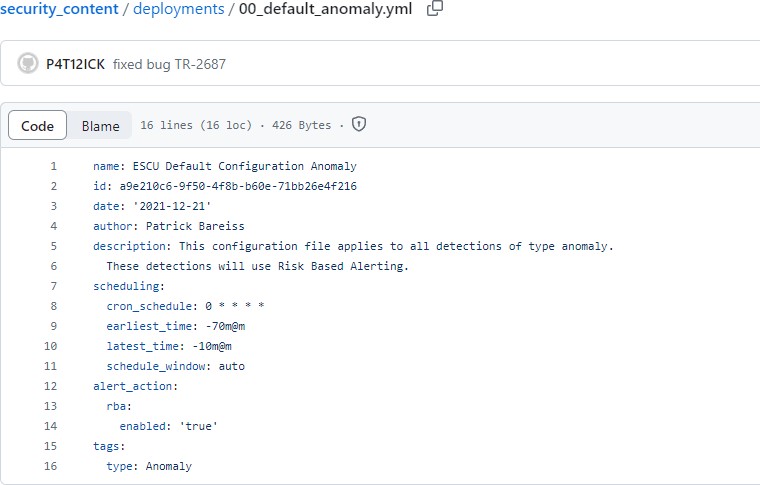
* determining the crawling target,
* extracting the required data, and
* storing the processed data

## Determining the crawling target

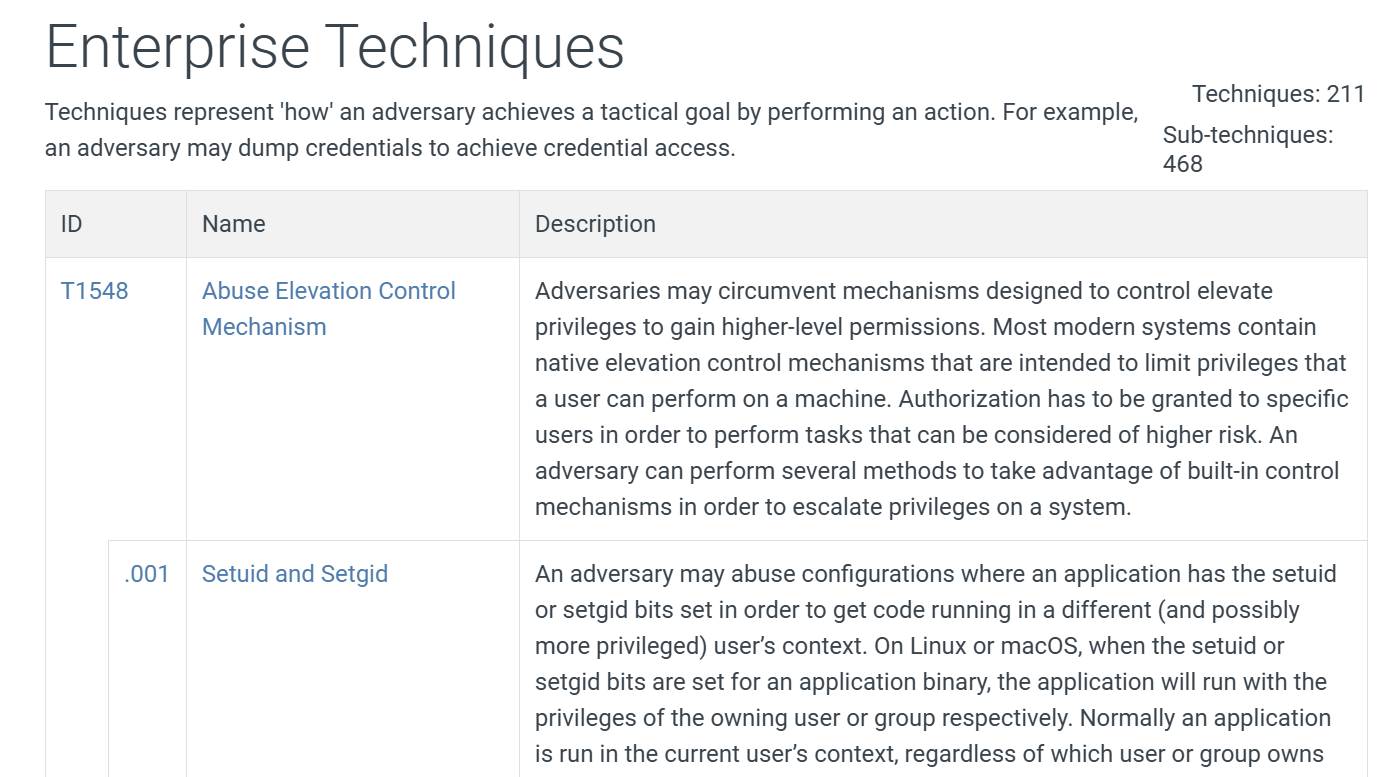
Data used in this research were the alerts and warnings collected from websites that listed and described them and also from the GitHub repository, as shown in Table I. We crawled down over 50,000 pieces of data from these websites. For example, we used the MITRE ATT&CK official website to scrape to get alert and warnings listed in ‘Enterprise Techniques’ for organizations’ cyberattack, is shown in Fig. 2.

## Extracting the required data

In this step, we parsed HTML source codes of the websites, as listed in Table I. In order to collect alert data from the official websites and also from the repositories, we used the Beautiful Soup crawler tool along with the HTML parser. Fig. 3 is an example of getting the name information from the HTML phrasing result. After that, data cleaning, filtering, formatting, and other processing of the extracted data was performed to obtain the required structure and format. We had to use a JSON file parser which helped us get the information from the YAML file structures.



1. Github Details



1. Mitre Att&ck Official Website Details

## Storing the processed alert data

These extracted data were stored into databases, files, or other storage media for subsequent analysis and use. We stored these data as csv files with ID, name, and description in the desired format of the look up/LUT table, discussed later. In total, we scraped over 50,000 alerts from these websites.

# MAnual Labeling of Alerts to Mitre att&ck

Once the alert dataset was collected, the manual labeling of these alerts to map to the MITRE ATT&CK techniques were carried out following the steps in Table II. A lookup table (LUT) was generated and populated held the necessary information of the alerts being manual labeled, as shown in Fig. 5.

Table III. Description of LUT number

| Range | Meaning |
| --- | --- |
| 0.5 >= LUT | The recommended technique must be in technique 1st, technique 2nd, and technique 3rd. |
| 0.25 > LUT | The recommended technique should not be in  technique 1st, technique 2nd, or technique 3rd. |
| 0.25<=LUT<0.5 | technique is allowed to be in technique 1st,  technique 2nd, and technique 3rd. but it does not  have to be |

1. Steps for Manual Labeling

**Input**: An alert is received.

LUT table ← {data, text, TRUE, DISCARD, technique 1st, technique 2nd, technique 3rd, note, regex lut addition, regex lut technique, lut\_tech\_code, lut\_tech, new technique flag}

**Compute:**

Technique\_id ← {keyword, weight, regex}

**Match:**

If Technique\_id exists in LUT table:

Use same Technique\_id for mapping

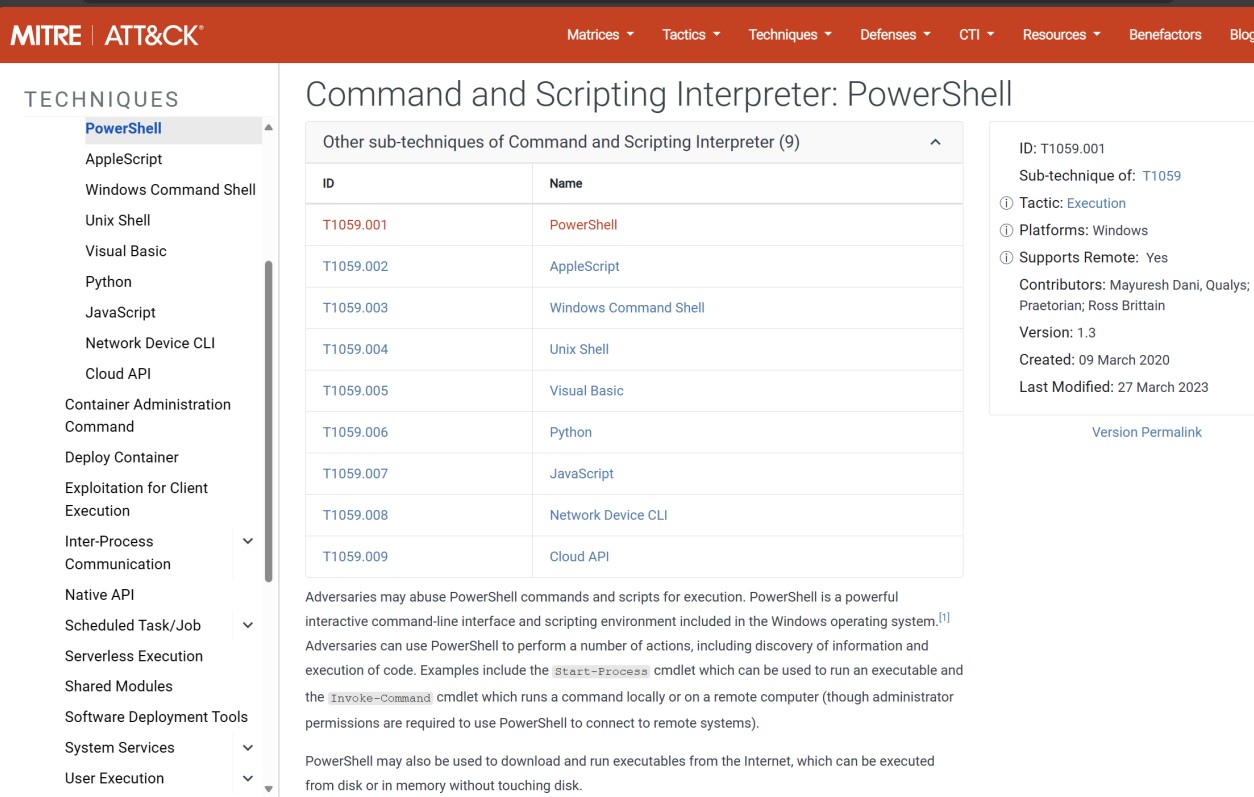
Else:

Update LUT table with computed Technique\_id

**End**

## Read the alert text and extract keywords

First, one alert was selected. The alert included attack name, description and information of the type of attack in details. We extracted the keywords from the alert description. For an example alert “powershell.exe executes the shellcode from the registry by calling the CreateThread() API” – we extracted two keywords ‘powershell’ and ‘registry’, as shown in Fig. 4.



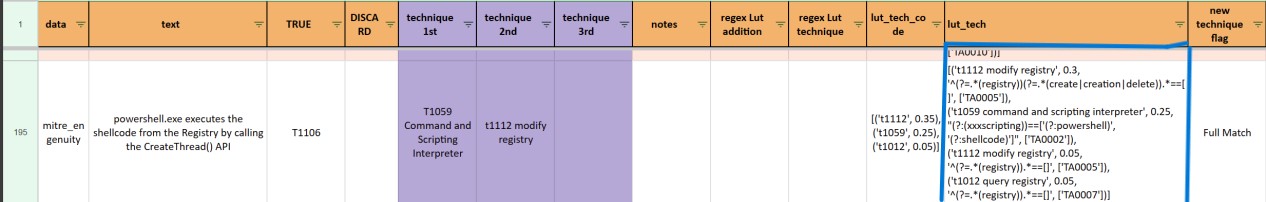
1. Mitre Att&ck Official Website

## Search apporpriate technique using extracted keywords

## We conducted searching these two keywords, in the Mitre Attack official website. The first keyword "powershell" as shown in Fig. 4 we used to search "powershell site attack.mitre.org" in the Google Dork browser. The search result returned ‘T1059:Command and Scripting’. From the introduction of T1059, we determined if it fitted the description of this alert. In this case,T1059 was the correct answer. In addition, the most related answer was, ‘001:powershell’, one of its sub-techniques. It was not necessary to label the sub-technique in the LUT table, this helped us better understand how it worked. Similarly, we got the technique ‘T1112 Modify Registry’ for the other keyword ‘registry’.

## Assign weight to new technique

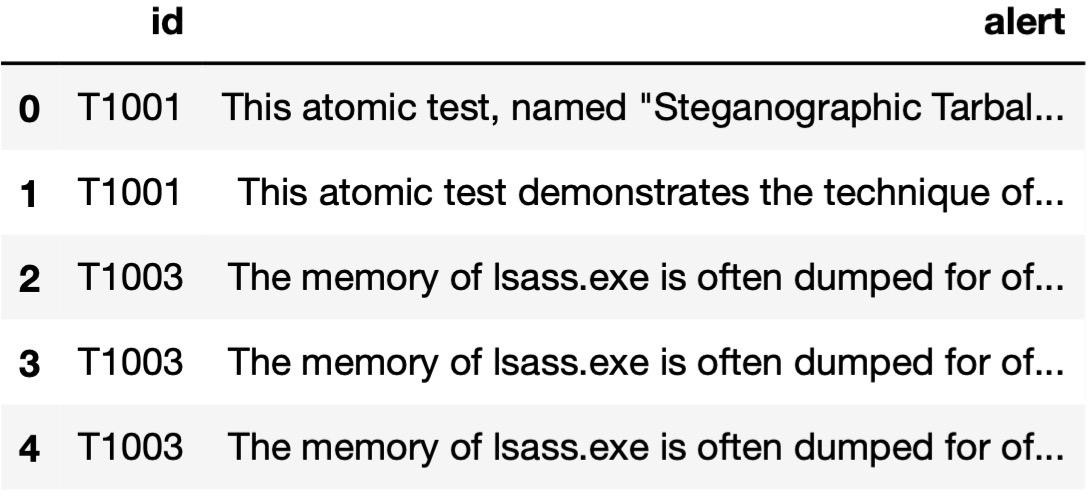
Based on the above search results, matching of the keywords to the techniques were inserted in the LUT table as technique 1st and technique 2nd, as shown in Fig. 5. These techniques were assigned as technique 1st or 2nd based on the weights in lut\_tech\_code in the LUT table. Table III shows the explanations of LUT weight assign. This manual weight allocation based on how much they matched, allowed technique either assign to technique 1st, technique 2nd, and technique 3rd. For the above alert example, the LUT weight for technique T1112 was 0.35 and technique T059 was 0.25. Therefore, T1112 and T1059 was written in the technique tag column as technique 1st, technique 2nd.



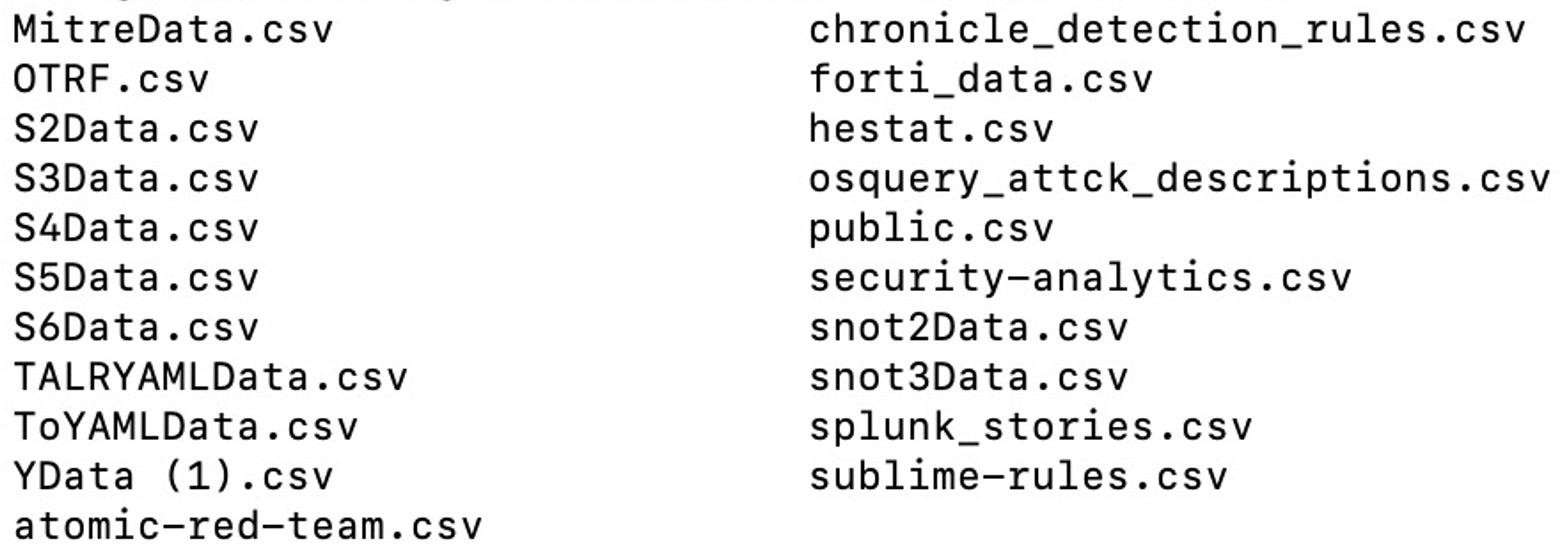
1. Look up table (LUT) details.

## Additional verifications with Regex

Regular Expressions (Regex) Regular Expression, or regex or regexp in short, is a powerful tool in searching and manipulating text strings, particularly in processing text files. Not every alert required us to write regex expressions; for alerts that were not recognized by following steps A to C, we used regex to determine the type of technique. Fig 5, shows regex expression for the above two techniques: T1112 and T1059 listed in 'lut-tech' column in the LUT table. Regex translated the alert description with more details and keywords. For another alert, "Network connection to Scranton (10.0.1.4) over port 5985.", there was no matching in the LUT table, neither got results from extracted keywords search through Google Dork. In such case, regex was needed to add the appropriate technique in the LUT table. For this additional verification, the written regex expression was put in "https://regex101.com" to verify whether the regex matched the alert or not. If the statement matched, the “1 match” in the upper right corner in the LUT table was used to indicate. The regex expression was included in the LUT table finishing the manual label process of the selected alert and then proceed to the next alert.



1. Dataset header List

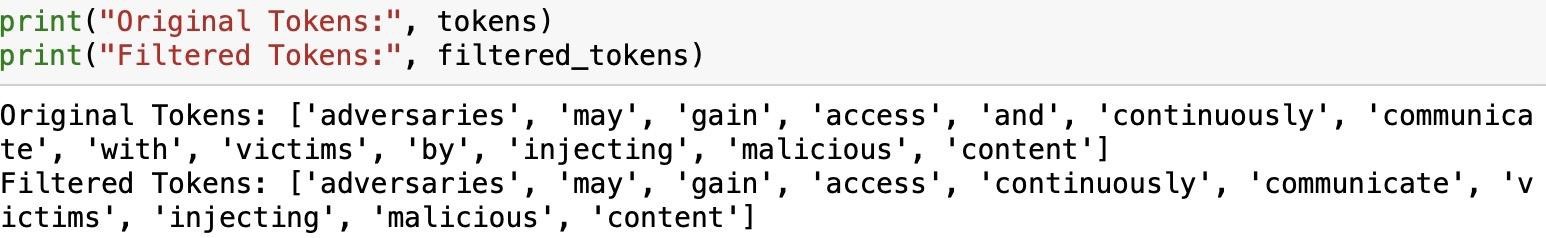


1. Raw Data List

# Automated Labeling of Alerts with ML Techniques

Manual labeling process was time consuming, required human intervention and prone to error. So to ease the process, in this section, auto labeling of the alerts were developed using several ML techniques. At the initial phase, we used three key technologies: the Natural Language Toolkit (NLTK) for processing text data, the attackcti Python module to access and manipulate the MITRE ATT&CK framework data, and the stix2 library to handle structured threat information in STIX format. For ML techniques, we employed Bidirectional Long Short Term Memory (BiLSTM), Random Forest (RF) and Light Gradient Boosting Machine (LightGBM) models for mapping alerts to MITRE ATT&CK techniques.

## Data Preparation



1. Sample Alert Processed by NLTK

Our raw data contained 21 csv files, as shown in Fig. 6. These 21 file were uploaded to Jupyter Notebook, organized as Python Pandas Data Frame, and combined into one single data frame. Then all data of this data frame was formatted into String, and all unused columns were dropped. Our clean data contained 2 columns: id and alert. The column “alert” was the cybersecurity alert and “id” was the correct correspond MITRE technique, as known as label. The head of our data set is shown in Fig. 7.

## Utilizing “attackcti” for Accessing and Manipulating updated MITRE ATT&CK Data

After collecting the alerts, we had to ensure updated information of the label - the MITRE ATT&CK techniques. We used attackcti, a Python module to access up-to-date ATT&CK content available in STIX via public TAXII server, so that the label was meaningful.

## Using “NLTK” to Analyze Data

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. It supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities. Data was cleaned of stop symbol, vectorized, and tokenized via NLTK. Fig. 8 shows an example of a processed alert.

## Using Attack\_Pattern from STIX2 to Analyze Data

Structured Threat Information Expression (STIX) is a language and serialization format used to exchange cyber threat intelligence (CTI). We generated our data into this language to make our data understandable for the ML models.

## Cosine Similarity

The cosine similarity score, which measures the cosine of the angle between two vectors projected in a multi-dimensional space, was used this score to find a correct MITRE ATT&CK technique for the alert. It calculated the score between the TF-IDF vectors of the alerts and the technique. The technique with the highest cosine similarity score to an alert was considered the best match, and its identifier was used as the label for that alert.

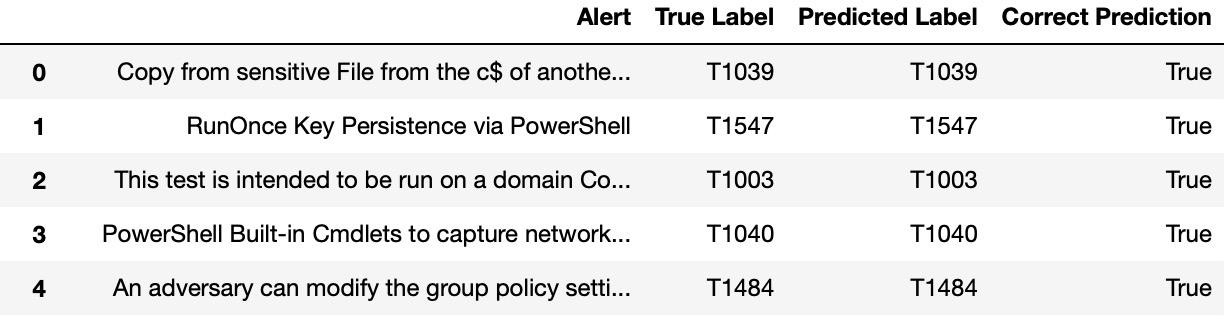
## ML Techniques

In this section, we evaluated three supervised machine learning pipelines for MITRE ATT&CK-based cybersecurity alert classification: a BiLSTM model, an RF model and a LGBM model. All approaches leveraged Natural Language Processing (NLP) transformations—specifically TF-IDF vectorization—and address class imbalance via Synthetic Minority Oversampling Technique (SMOTE). The dataset consisted of manually and synthetically generated alerts mapped to 14 ATT&CK tactics and multiple techniques. Initially the BiLSTM model was implemented, because of the low accuracy results, we extended our research towards other two models.

### BiLSTM model

A Bidirectional Long Short-Term Memory (BiLSTM) network is a specialized form of recurrent neural network (RNN) designed to handle time-series sequential data by analyzing it in both forward and backward directions [7]. This dual processing enables the model to understand context from both earlier and later elements in a sequence, which is especially valuable in tasks like natural language processing (NLP), where the meaning of a word often depends on its surrounding words.

We implemented a pre-trained BiLSTM model, trained on the alert dataset and predicted the label (technique id). Fig. 9 shows predicted labels with accuracies compared to true label (manually set label). We got prediction accuracy of 62%.



1. BiLSTM model - Auto Labeling Result.

### RF model

The RF baseline integrates TF-IDF features, cosine similarity scoring, and randomized hyperparameter search. RandomizedSearchCV samples parameter combinations from user defined distributions, providing a more computationally efficient alternative to exhaustive grid search [8].

The optimal configuration for this work was identified as: n\_estimators=300, max\_depth=None, min\_samples\_split=2, in\_samples\_leaf=1. Despite these optimizations, RF training on 2,279 alerts required over 3 hours and consumed approximately 8 GB of RAM. The model achieved 56.17% technique-level accuracy.

### LGBM Model

LightGBM, developed by Microsoft, is a gradient boosting framework designed for high efficiency and scalability. In this study, LightGBM was configured with architectural innovations like gradient-based one-side sampling (GOSS), exclusive feature bundling (EFB), histogram-based decision trees, leaf-wise growth strategy [9].

These optimizations allowed LightGBM to train on 2,000 synthetic alerts in under 20 minutes, using less than 1.5 GB of RAM—over a 10× speed improvement and an 80% memory reduction compared to RF. The model achieved 99.00% accuracy at the technique level and 100% at the tactic level. Unlike RF, LightGBM adopted a hierarchical classification approach, predicting tactics first and then refining predictions to techniques. This structure preserved semantic context within the MITRE ATT&CK framework, improved rare-class recognition, and reduced misclassification between closely related techniques.

# Resutls & Discussions

In this study, the baseline alert dataset with appropriate MITRE ATT&CK technique was created by manual labeling in collaboration with our industrial partner, Cypienta. A look up table was populated with required information to map alerts scraped from variety of websites, carefully used tools and techniques. Such as, HTML parsing with Beautiful Soup for alert extraction from sites, searching with keywords in Google Dork, assigning weights to mapped techniques, applying Regex for mapping alerts as additional process – these curated the manual labeling process appropriate and acceptable.

Then we expanded our research to investigate auto labeling of an alert to a MITRE ATT&CK technique using a machine learning technique. The manual annotation process was inherently time-consuming, required continuous human oversight, and was susceptible to errors, thereby limiting scalability and consistency. To address these limitations, several ML algorithms were implemented and their performances were compared in predicting the label of the alert. We used the baseline dataset (LUT table) collected during the manual labeling process as the model input data. Data was preprocessed, cleaned and fewer columns were selected for model training. Few tools and techniques were also used during this process- attackcti for updated MITRE ATT&CK, NLTK for text-based data analysis, STIX to exchange CTI, cosine similarity to find the best-matched technique for an alert. Several ML models were trained once data was ready. Among the models, we used a pre-trained BiLSTM that was retrained with our baseline dataset while the other two models were trained directly with the baseline dataset. 5-fold cross validation was conducted for evaluating the models. Performance of the BiLSTM, RF and LGBM models were 62%, 56% and 99%, respectively. Table IV summarizes the comparative performances of these models.

The BiLSTM model’s performance was affected by the model architecture, retraining data and increased computational overhead.

* Bidirectional processing in BiLSTM architectures effectively doubled the computational requirements compared to their unidirectional counterparts.
* Although BiLSTMs offered enhanced contextual representation by leveraging both past and future information, their training process was more intricate and demanded larger datasets to mitigate the risk of overfitting.

Table IV Performance MAtrix

| Metric | BiLSTM | RF | LGBM |
| --- | --- | --- | --- |
| Accuracy of predicting Technique ID | 62% | 56.17% | 99% |
| Accuracy of predicting Tactic ID | NA | NA | 100% |
| Training Time |  | >3 hrs | <20 min |
| Memory Usage |  | 8 GB | <1.5 GB |

While the RF model remained attractive for its interpretability and robustness on moderate-sized datasets, its scalability and speed in large-scale, high-dimensional cybersecurity datasets was constrained. The RF model performance limitations stemmed from:

* Inability to efficiently capture complex feature interactions in high-dimensional TF-IDF spaces.
* A flat classification structure that disregarded the hierarchical relationships between tactics and techniques.
* Memory overhead from repeated cross-validation during parameter tuning.
* Its scalability and speed in large-scale, high-dimensional cybersecurity dataset was constrained.

The experimental results demonstrate that LightGBM not only surpassed the BiLSTM and RF models in accuracy but also dramatically improved training efficiency and scalability. The architectural features of LightGBM were particularly advantageous for:

* High-dimensional NLP features: EFB and histogram-based trees handled sparse TF-IDF matrices without requiring dense transformations.
* Operational deployment: Reduced training time and memory usage made retraining feasible in production environments with streaming alert data.
* Hierarchical threat mapping: The tactic → technique model design enhanced interpretability and aligned with real-world security workflows.

LightGBM’s combination of speed, precision, and scalability made it the preferred choice for production-grade threat intelligence systems.

# Conclusion & Future Work

In this paper, we proposed a solution to analyze cybersecurity alerts and warnings scraped from different websites and utilized a cybersecurity analysis framework (MITRE Technique) for alert identification with corresponding mitigation solution. This research was carried out in two phases: i) manual labeling: scraping alerts from public websites and GitHub by python scripts, archiving manual labeling by google Dork and regular expression, and ii) auto labeling: building an auto labeling model with NLP, STIX, TF-IDF with the BiLSTM, RF and LGBM models. The integration of machine learning methodologies with the MITRE ATT&CK framework has the potential to significantly enhance both the speed and precision of cyber threat detection, while also improving the scalability and reliability of the overall system. The proposed cybersecurity analysis framework, with automating threat classification can help cybersecurity organizations to strengthen their security posture by responding quickly and effectively to emerging threats.

Future work will focus on refining the machine learning models and expanding the dataset to include more diverse threat types to further improve the framework's ability to detect and mitigate complex cyberattacks. The framework can be extended to incorporate advanced techniques for malware recognition and automated labeling. This enhancement would contribute to transforming the system into a comprehensive and scalable solution for Cyber Threat Intelligence (CTI) analysis, thereby improving its effectiveness in real-world threat detection and response scenarios. In future work, we aim to automate the currently manual processes of data collection and annotation. This advancement will facilitate the development of a fully automated platform for mapping Cyber Threat Intelligence (CTI) to MITRE ATT&CK techniques, thereby enhancing the efficiency and scalability of threat analysis.

##### Acknowledgment

This work was conducted as part of MS degree program requirement in cybersecurity at New York Tech, Vancouver campus in conjunction with a Canadian company, Cypienta, that provides security architecture support. We are grateful to work on this RIIPEN project and contribute in a state-of-art threat analysis framework.

##### References

1. MITRE, “MITRE ATT&CKTM,” Mitre.org. https://attack.mitre.org/
2. V. Legoy, M. Caselli, C. Seifert, and A. Peter, “Automated Retrieval of ATT&CK Tactics and Techniques for Cyber Threat Reports,” Apr. 2020.
3. A. Kuppa, L. Aouad, and N.-A. Le-Khac, “Linking CVE’s to MITRE ATT&CK Techniques,” The 16th International Conference on Availability, Reliability and Security, Aug. 2021, doi: https://doi.org/10.1145/3465481.3465758.
4. T.-M. Georgescu, “Natural Language Processing Model for Automatic Analysis of Cybersecurity-Related Documents,” Symmetry, vol. 12, no. 3, p. 354, Mar. 2020, doi: https://doi.org/10.3390/sym12030354.
5. Gurinder Pal Singh, Jasminder Kaur Sandhu, and Manish Kumar Hooda, “BiLSTM Classifier: A New Approach for Detecting Cyber-Attacks in MITRE ATTACK Framework,” Sep. 2023, doi: https://doi.org/10.1109/ic3i59117.2023.10398021.
6. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
7. Siami-Namini, Sima, Neda Tavakoli, and Akbar Siami Namin. "The performance of LSTM and BiLSTM in forecasting time series." 2019 IEEE International conference on big data (Big Data). IEEE, 2019.
8. Scikit-learn Developers, “sklearn.model\_selection.RandomizedSearchCV,” Scikit-learn Documentation. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.RandomizedSearchCV.html.
9. G. Ke et al., “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” in Advances in Neural Information Processing Systems, 2017.