Exploring and Mapping Exploitable Code on Paste Sites to the MITRE ATT&CK Framework for Proactive Cyber Threat Intelligence

By

Tala Vahedi

A Master's Paper Submitted to the Faculty of the

DEPARTMENT OF MANAGEMENT INFORMATION SYSTEMS

ELLER COLLEGE OF MANAGEMENT

In Partial Fulfillment of the Requirements

For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

2022

STATEMENT BY AUTHOR

This paper has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona. Brief quotations from this paper are allowable without special permission, provided that an accurate acknowledgement of the source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part must be obtained from the author.

SIGNED: Tala Vahedi

APPROVAL BY MASTERS PAPER ADVISOR

This paper has been approved on the date shown below:

_____ _____

Dr. Mark Patton Date

Lecturer of Management Information Systems

Table of Contents

Abstract

Malicious cyber activities impose substantial costs on the U.S. economy and global markets. Cyber-criminals often use information-sharing social media platforms such as paste sites (e.g., Pastebin) to share vast amounts of plain text content related to Personally Identifiable Information (PII), credit card numbers, exploit code, malware, and other sensitive content. Paste sites can provide targeted Cyber Threat Intelligence (CTI) about potential threats and prior breaches. In this research, we propose two novel models to categorize similar pastes to identify the types of malicious content present on pastes as well as map pastes to the MITRE ATT&CK cyber risk management framework (CRMF) to identify adversarial tactics and techniques to aid proactive CTI. First, we introduce Bidirectional Encoder Representation from Transformers (BERT) with Latent Dirichlet Allocation (LDA) model to categorize pastes automatically. Our proposed BERT- LDA model leverages a neural network transformer architecture to capture sequential dependencies when representing each sentence in a paste. BERT-LDA replaces the Bag-of-Words (BoW) approach in the conventional LDA with a Bag-of-Labels (BoL) that encompasses class labels at the sequence level. Second, we propose a Convolutional BiLSTM Transformer (CBT) multi-label classification (MLC) method that automatically maps lengthy exploit code on paste sites to the MITRE ATT&CK CRMF to identify adversarial techniques in support of proactive CTI. Our MLC method consists of a convolutional neural network layer placed before a Transformer block, a concatenated pooling from a global max pooling and global average, and a BiLSTM pair-wise function within the Transformer to capture word and or sequence orders. Results of our BERT- LDA and CBT case studies suggest that pertinent information may be extracted from paste sites to serve as proactive CTI. The insights provided by this study could be used by organizations to proactively mitigate potential damage on their cyber or IT infrastructure.

Introduction

Malicious hacking tools developed by cyber-criminals are becoming increasingly more complex and dangerous. Global cybercrime costs are estimated exceed $10.5 trillion annually by 2025 (Morgan et al., 2021). To mitigate these costs, cybersecurity experts recommend that organizations invest in proactive Cyber Threat Intelligence (CTI) and cybersecurity risk management frameworks (CRMF) (Samtani et al., 2019; Paul and Wang, 2019).

One pertinent and prevailing source of proactive CTI are information-sharing platforms, e.g., paste sites (Vahedi et al., 2021). A paste site is a large information-sharing platform that allows millions of users to anonymously post large quantities of plain text content (Vahedi et al., 2021). For example, Pastebin, a well-known paste site, currently has over 18 million registered users, 150 million public pastes, and over 16 million monthly visitors (Vahedi et al., 2021). Personally Identifiable Information (PII), credit card numbers, exploit code, malware, and other sensitive

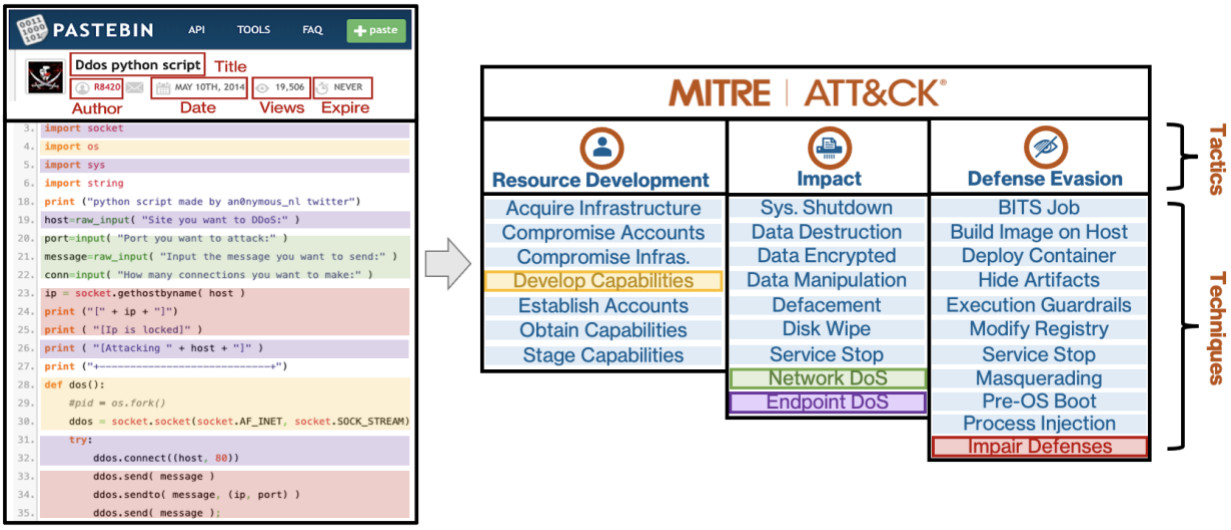

Figure 1. A Sample Paste from Pastebin (On Left with Metadata on Top and Malicious Source Code On Bottom For A Distributed Denial Of Service (DDoS) Attack) Being Mapped to Multiple MITRE ATT&CK Techniques under separate Tactics (on right).

content are often accessible on paste sites (Vahedi et al., 2021).

Each paste includes metadata such as paste title, author, post date, views, and expiration date. Below the post metadata are the plain-text post contents (e.g., DDoS at the bottom of Figure. 1), which may include various predominant keywords appearing in sequential order (e.g., "DDoS" on line 7 appears before "Attacking" on line 16 of Figure. 1). Overall, hundreds of thousands of illicit users share and contribute millions of pastes. Consequently, paste sites can serve as a

valuable data source for understanding cyber-criminal tactics, techniques, and procedures (TTPs).

Automatically extracting themes from pastes on paste sites and labeling the malicious technique(s) can assist in identifying gaps within an organization's current security posture. Therefore, our objective in this study is two-fold. First, we propose a novel Bidirectional Encoder Representation from Transformers with Latent Dirichlet Allocation (BERT-LDA) topic model to automatically extract themes from large pastes on paste sites. The proposed BERT-LDA has two novelties: 1) BERT-LDA leverages BERT, a prevailing neural network transformer architecture, to capture sequential dependencies within the text of a paste to represent each sentence, and 2) BERT-LDA replaces the Bag-of-Words (BoW) used in the conventional LDA model with Bag-of-Labels (BoL) that encompasses class labels at the sequence level when extracting topics. Second, we develop a novel Convolutional BiLSTM Transformer (CBT) multi-label classification (MLC) method that automatically maps lengthy malicious content on paste sites to the MITRE ATT&CK CRMF to identify adversarial techniques in support of proactive CTI. Our CBT model has three key novelties: 1) a Convolutional Neural Network (CNN) layer placed before the Transformer block to extract and learn relative locality information, 2) multiple pooling to reduce the feature vector space and capture low and high-level features, and 3) a BiLSTM pair-wise function to capture word/sequence orders and provide internal memory.

Literature Review

We review four areas of literature for this study. First, we examined recent research studying paste sites to identify their objectives and prevailing methods. Second, neural network-based topic models were reviewed to identify prevailing unsupervised algorithms for automatically categorizing large quantities of text. Third, we reviewed BERT to identify how the prevailing pre-trained language model could be incorporated into a topic model. Finally, we reviewed the basic Transformer model and BERT to identify how the prevailing pre-trained language model could be leveraged for lengthy contiguous sequential dependencies.

Paste Site Analysis

Paste sites allow users to freely share text online anonymously. Unlike other hacker social media platforms (e.g., forums) a paste site is a repository to store plain-text only (i.e., no multi-media). Recent literature analyzing paste sites have focused on using a single paste site as a data testbed (Riesco et al., 2019; Pakari et al., 2020; Brengal et al., 2018; Imai et al., 2018). These studies have focused on identifying leaked PII (e.g., emails, passwords, phone numbers, credit cards, and secret keys) based on the metadata associated with each paste, and not the paste content (Riesco et al., 2019; Pakari et al., 2020; Brengal et al., 2018; Imai et al., 2018). As a result, the knowledge about the major categories of content on paste sites is mostly unexplored and unknown. When choosing analytical methods, recent literature has primarily relied on classical machine learning algorithms (Riesco et al., 2019), manual labeling and analysis (Riesco et al., 2019), and theoretical methods (Pakari et al., 2020; Imai et al., 2018; Eseryel et al., 2020) to analyze each paste's metadata and execute their research objectives. Such methods often cannot reveal the major themes (i.e., topics, categories) of content within paste sites. Since topic modeling is a common approach to automatically categorize large quantities of unstructured text (Samtani et al., 2020b), we review prevailing neural network-based topic models next.

Neural Network-based Topic Models

Topic modeling is a common approach to categorize text in social media platforms (Samtani et al., 2020). LDA is the prevailing unsupervised topic modeling approach for categorizing text in hacker social media platforms where there is limited prior knowledge (Samtani et al., 2020; Li et al., 2016). LDA is an unsupervised generative statistical model that uses a BoW approach for representing text. Despite its prevalence, LDA's use of Bag-of-Words (BoW) often overlooks word sequences in text and limits performance (Barbieri et al., 2013). In recent years, researchers have improved the performance of the LDA by incorporating neural network components to capture specific aspects of the input data (e.g., capture co-occurrences) (Chai et al., 2019). There are three major categories of neural network topic modeling approaches:

- **Feed Forward Neural Network (FFNN)-based topic models**, such as DeepLDA (Bhat et al., 2020) and Autoencoding Variational Inference for Topic Model (AVITM) (Srivastava et al., 2017) that use feed-forward neural networks to represent input texts as flat feature vectors for input into topic models.

- **Generative-based topic models**, such as Gaussian- Bidirectional Adversarial Training (G-BAT) (Wang et al., 2020) and Weibull Hybrid Autoencoding Inference (WHAI) (Zhang et al., 2018), infer a deep probabilistic topic model with a generative encoder network (e.g., adversarial network) to capture the hierarchical document latent representations.
- **Recurrent Neural Networks (RNN)-based topic models**, such as TopicRNN (Zhou et al., 2016) utilize Long Short- Term Memory (LSTM) and/or Bidirectional LSTM (BiLSTM) (Dieng et al., 2018) to capture word positions and sequences of text for input into LDA or another topic model.

Among the different categories of neural network topic models, those that rely on recurrent architectures such as RNN, LSTM, and BiLSTM architectures can capture and learn from word positions, co-occurrences, and sequences. However, the performances of these models often suffer when operating on text with long contiguous sequences and dependencies (e.g., pastes). As a result, an alternative neural network architecture is required to capture and represent the lengthy sequential dependencies present in pastes. Therefore, we review BERT in the following sub-section.

Transformers and BERT

A Transformer is a state-of-the-art Natural Language Processing (NLP) architecture that aims to solve sequence-to-sequence tasks while handling long-distance dependencies (Vaswani et al., 2017). The architecture (Figure 2) includes a multi-head attention (MHA), normalization, and Feed-Forward (FF) sublayers (Vaswani et al., 2017). One advantage of Transformers lies in the MHA's (Figure 2, point A) ability to attend to input from representation subspaces at various positions concurrently, making it ideal for capturing long-distance dependencies (Vaswani et al., 2017). Despite their success, Transformers have two major limitations:

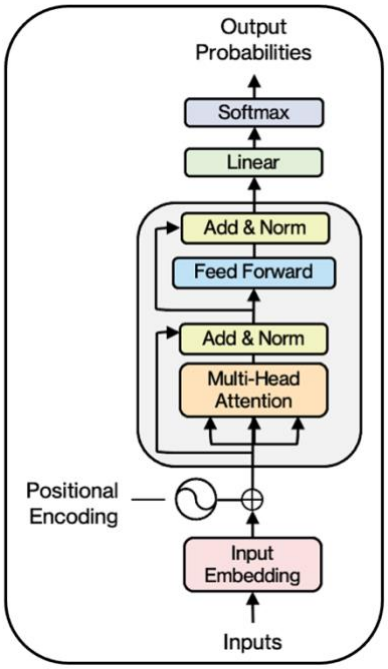

Figure 1. Transformer Model Architecture Modified for Classification Tasks

The Transformer's self-attention is a representation of a sequence that is calculated by linking distinct words in the same sequence,

ignoring the relationship of neighboring elements (i.e., n-grams) (Lin et al., 2018; Yang et al., 2018).

Convolutional Neural Network (CNN) convolutional kernels/filters can be thought of as n-gram extractors for Natural Language Processing (NLP) tasks, converting each n-gram into a vector capturing semantic and syntactic information (Kim et al., 2014; Li et al., 2021). The token-wise feed-forward (FF) (Figure 2, point B) layer analyzes each input token embedding individually, ignoring word and sequence positions (Yun et al., 2019). Bi-directional Long-Short Term Memory (BiLSTM) can capture the underlying context of sequential text and leverages bi-directional word positions to offer internal memory (Sukhbaatar et al., 2019).
BERT is a transformer-based Pre-trained Language Model (PTLM) that aims to represent unlabeled text (Devlin et al., 2019). BERT has consistently outperformed recurrent models in numerous unsupervised NLP tasks due to its ability to operate on significantly longer blocks of text than conventional LSTMs (Devlin et al., 2019; Qiu et al., 2020). We present BERT's architecture in Figure. 3.

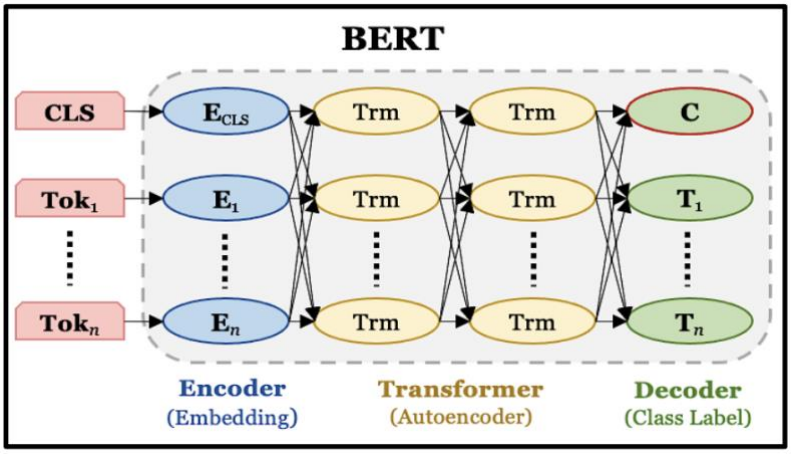

Figure 2. BERT Architecture. Note: C = Class Label, CLS = Classification, E = Encoder, T = Token Output, Tok = Token, Trm = Transformer.

BERT includes three major components. First, an encoder reads, tokenizes, and creates a sentence embedding vector from the text input. Second, a transformer using an autoencoder learns contextual relationships between words in text (i.e., tokens) bidirectionally. Third, a decoder produces a class label for each sequence within the text. These class labels can be aggregated to produce a Bag-of-Labels (BoL) i.e., frequency of labels for all sequences in an input text. Despite the potential utility of BERT and its ability to produce a BoL that retains

sequential information from long texts (and can therefore potentially address the limitations of prevailing topic models), how to incorporate BERT into LDA to categorize content on paste sites requires additional study.

Research Gaps and Questions

We identified four research gaps from our literature review. First, prior studies limited their research to a single platform to perform targeted analysis (i.e., identification of breached records) instead of comprehensively analyzing all categories of pastes across multiple paste sites. Uncovering the full threat landscape of paste sites can provide targeted CTI about potential threats and prior breaches. Second, the prevailing topic model for categorizing hacker content in social media content, LDA, operates on a BoW model that can miss word order. Third, BERT uses transformers for unsupervised learning and can capture and represent word order and sequential dependencies as a BoL. However, how to integrate BERT into LDA requires further study. Finally, Transformers have achieved state-of-the-art performance on natural language processing (NLP) tasks; however, running them on lengthy malicious source code for multi-label classification (MLC) tasks requires further study.

To address to those gaps, we pose the following research questions for study:

- What categories of malicious content exist within prevailing paste sites?
- How can BERT be integrated into LDA to capture lengthy contiguous sequential dependencies from paste site content when categorizing pastes?
- How can we map Paste Site source code to multiple MITRE ATT&CK Techniques and learn more about their technical capabilities to enable CTI?
- How can we augment the Transformer model to capture word order and locality of text from lengthy source code on paste sites while mapping them to MITRE ATT&CK Techniques?

Research Design

We propose two distinct research frameworks to 1) categorize and extract themes from long and contiguous pastes to identify malicious content, and 2) map the malicious content found within pastes to the MITRE ATT&CK CRMF.

BERT-LDA

Our first research framework, BERT-LDA, includes three major components (Figure. 4): (1) Data Collection and Pre-Processing, (2) BERT-LDA, and (3) Evaluation & Case Study. We describe each major component in the following sub-sections.

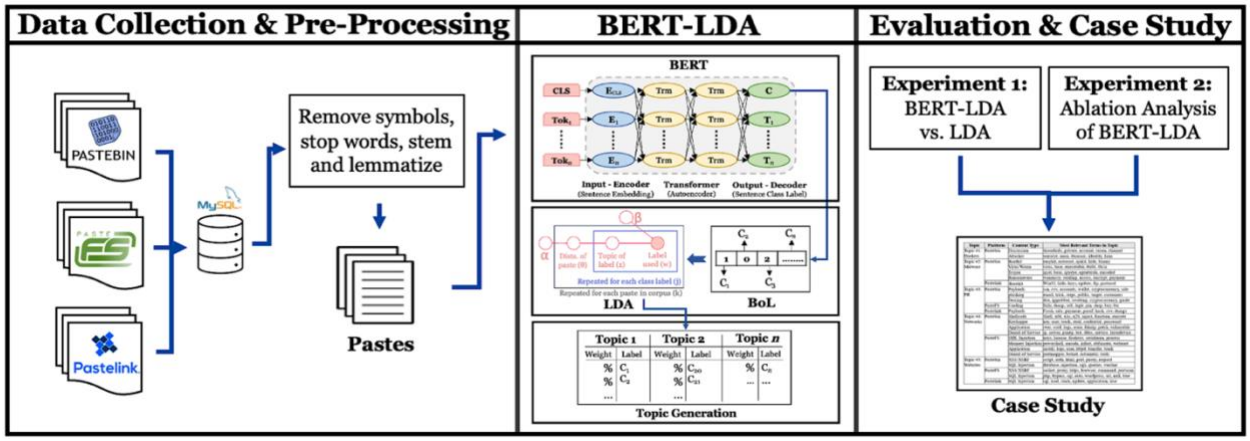

Figure 3. Proposed Research Framework for Categorizing Malicious Content on Paste Sites.

Data Collection & Pre-Processing

We identified three prevailing paste sites for collection based on feedback from cybersecurity experts: Pastebin, PasteFS, and Pastelink. Custom web crawlers were developed to collect each paste and their associated metadata (e.g., title, author). A summary of our collection is presented in Table I.

Table 1. Summary of Data Collection

Name	Start-End Date		Posts	Authors	Views
Pastebin	8/5/07	1/16/21	304,321	42,214	507,930,398
PasteFS	6/3/15	12/10/20	238,250	1,495	519,060
Pastelink	2/27/15	1/2/21	3,711,882	N/A	4,098,966
3 Sites	**8/5/07-1/16/21**		**4,254,453**	**43,709**	**512,548,424**

Our testbed includes over 4,254,453 pastes from 8/5/2007 to 1/16/2021 made by over 43,709 authors. Pastelink was the largest site in our collection, with 3,711,882 pastes. Overall, our collection exceeds the largest data collection in prior research by over 500,000 pastes (Riesco et al., 2019). Following best practice in hacker social media analytics, we pre-processed each paste

by removing symbols, white space, special characters, and stop words and stemming, and lemmatizing each word (Li et al., 2016).

Proposed BERT-LDA Model

Given the limitations of prevailing topic models as it pertains to categorizing long, contiguous text (e.g., pastes), we incorporate BERT into LDA. We present the proposed BERT-LDA in Figure 5.

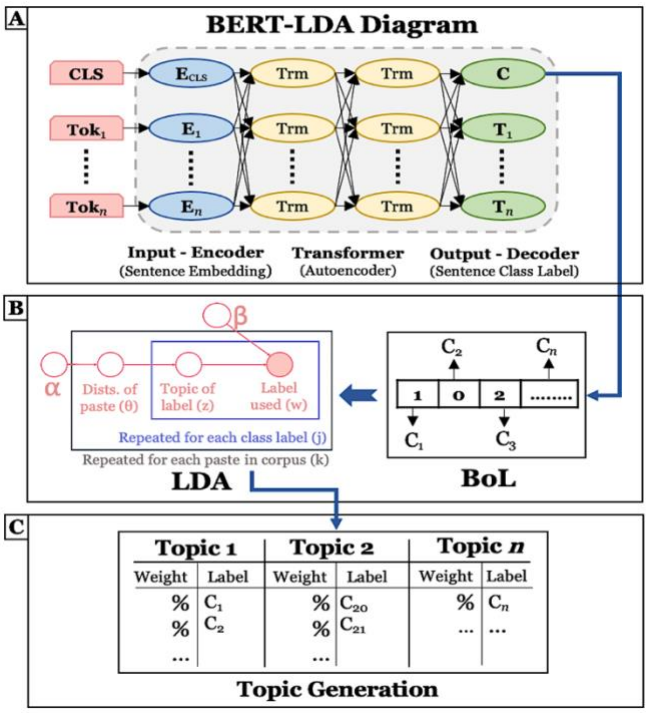

Figure 4. Proposed BERT-LDA Architecture for Paste Categorization

BERT-LDA consists of three major components:

A. **BERT:** BERT's encoder tokenizes each word in each sequence (sentence) for every inputted paste. Tokens are used to create a sentence embedding to represent each sequence. The transformer's autoencoder reads sequences bidirectionally to create a sequence label.

B. **BoL and LDA**: the model replaces the traditional BoW in the conventional LDA model with the BoL (produced by BERT) to create a vocabulary of all unique labels. Replacing the BoW with a BoL helps to capture information at about each paste's semantics at the sequence-level, rather than at the word-level.

C. **Topic Generation**: LDA produces topics based on each paste's BoL.

BERT-LDA's core novelty resides in replacing LDA's BoW with a BoL generated from BERT. As such, we illustrate BERT-LDA's BoL functionality in further detail in Fig. 5. BERT-LDA's BoL operates as follows:

1. Each inputted paste is tokenized and sequenced. BERT inserts a "CLS" token at the beginning of first sentence and a "SEP" token at end of each sentence in a paste.
2. The sequenced tokens are passed to the encoder layer to create a sentence embedding vector.
3. The transformer then generates sequence level class labels from each vector (e.g., "ddos" and "attack").
4. All sequence labels are appended to a BoL to capture label frequencies at the sequence level.
5. The BoW in the conventional LDA is replaced with the BoL. Finally, LDA outputs topics based on the inputted BoL.

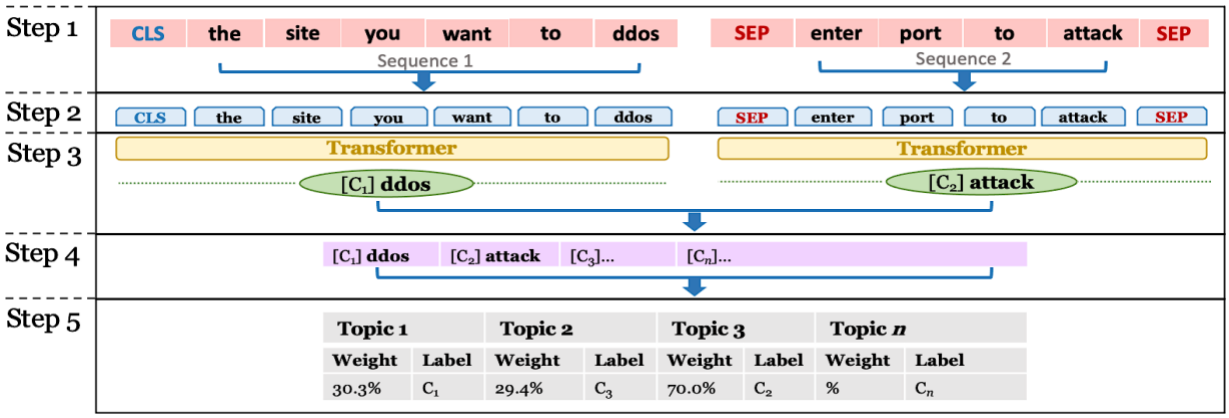

Figure 5. Illustration of BERT-LDA's Process of Generating BOL for LDA

Evaluation and Case Study

We evaluated BERT-LDA with two sets of experiments. In Experiment 1, we compared BERT-LDA against the conventional LDA model, the prevailing approach for unsupervised topic modeling (Samtani et al., 2020; Li et al., 2016). Experiment 2 is an ablation analysis for BERT-LDA that investigates how substituting the BERT in BERT-LDA with other prevalent PTLMs impacts performance. PTLMs used for ablation analysis include DistilBERT, Cross-lingual language model (XLM), and GPT-2. These models were selected due to their cutting-edge

performance in various unsupervised text mining tasks (Qiu et al., 2020). DistilBERT's architecture is designed to have 40% less parameters and runs 60% faster than BERT (Sanh et al., 2019). XLM provides a robust pre-training method for cross-lingual understanding tasks (Conneau et al., 2020). Finally, GPT-2 includes a transformer decoder as the language model for text generation (Radford et al., 2018).

We executed both experiments on each of the collected paste sites (4,254,453 total pastes). The performances for each algorithm were measured using the perplexity metric. Perplexity measures how well a model predicts a given sample and is frequently used to compare topic models (Blei et al., 2003; Ampel et al., 2020). Lower perplexity scores indicate higher performances (Blei et al., 2003). Benchmarking topic modeling algorithms in this fashion is a commonly accepted practice in hacker social media analytics literature (Samtani et al., 2020a; Li et al., 2016b).
In addition to conducting experiments to evaluate BERT-LDA's performance, we executed a case study to demonstrate the BERT-LDA's potential practical utility for possible CTI applications. To execute the case study, we applied BERT-LDA to extract topics from each collected paste site. Following common practice in topic modeling literature, we manually assign names to each of the outputted topics (Li et al., 2016b).

Results and Discussion

Experiment 1: BERT-LDA vs. LDA

In Experiment 1, we compared the proposed BERT-LDA against the conventional LDA model. We present the results of Experiment 1 in Table 4. The top performances are highlighted in boldface.

Table 2. EXPERIMENT 1 RESULTS: BERT-LDA VS LDA

Model	Topics	Pastebin	PasteFS	Pastelink
LDA	5	4,781.17	726.35	3,491.82
	10	4,016.33	546.82	3,491.82
	15	3,567.04	494.56	2,385.04
	20	2,475.64	446.88	1,121.44
BERT-LDA	**5**	**171.29**	**363.49**	**254.91**
	10	229.92	462.48	351.41
	15	276.91	521.12	403.65
	20	307.75	568.36	458.73

BERT-LDA achieved its best (i.e., lowest) perplexity scores across all platforms with 5 topics. In contrast, LDA achieved its best performances with 20 topics. With five topics, our BERT-LDA model outperformed the traditional LDA model for Pastebin (171.29 to 4,781.17), PasteFS (363.49 to 726.35), and Pastelink (254.91 to 3,491.82). BERT-LDA applies a BoL model to capture label frequencies and presumably nearly holds a dozen labels in its vocabulary. In contrast, LDA utilizes a BoW model. While this model captures word frequencies, it commonly has a vocabulary in the thousands. Therefore, it is plausible that the difference in label quantity between BERT-LDA's BoL and LDA's BoW resulted in BERT-LDA outperforming LDA.

Experiment 2: Ablation Analysis of BERT-LDA

In Experiment 2, we evaluated how BERT-LDA performed when BERT was replaced with alternative PTLMs perform. We present the results of Experiment 2 in Table 5. The top performing algorithm is highlighted in boldface.

Table 3. EXPERIMENT 2 RESULTS: ABLATION ANALYSIS OF BERT-LDA

Model	Topics	Pastebin	PasteFS	Pastelink
XLM-LDA	5	270.80	402.60	322.07
	10	370.90	445.19	428.70
	15	439.69	452.28	496.46
	20	496.49	501.68	565.89
GPT2-LDA	5	173.84	422.58	227.87
	10	228.92	511.85	300.69
	15	265.09	570.37	346.09
	20	290.14	608.08	391.79
DistilBERT -LDA	5	256.49	**362.46**	274.79
	10	333.18	439.25	346.48
	15	386.73	491.31	406.58
	20	428.26	531.24	446.38
BERT-LDA (ours)	5	**171.29**	363.49	**254.91**
	10	229.92	462.48	351.41
	15	276.91	521.12	403.65
	20	307.75	568.36	458.73

Across all three platforms for each model, 5 topics resulted in lower perplexity scores than 10, 15 and 20 topics. GPT2-LDA outperformed XLM-LDA for Pastebin (173.84 to 270.80) and Pastelink (227.88 to 322.07), while only being 19.97 higher (422.58 to 402.61) on the PasteFS dataset. DistilBERT-LDA outperformed GPT-2-LDA and XLM-LDA for PasteFS (362.46), although underperformed against GPT2-LDA for Pastebin (256.49) and Pastelink (274.79). BERT-LDA achieved the best perplexity scores for Pastebin (171.30) and Pastelink (254.92), while only being 1.03 above the DistilBERT-LDA perplexity for PasteFS. PasteFS posts often exceed BERT's acceptable token range (i.e., 512 tokens) allowing DistilBERT's inference layer to be more powerful than BERT's. Overall, BERT-LDA's bidirectional transformer encoder layer shows clear improvements over XLM-LDA, GPT2-LDA, and DistilBERT-LDA.

Case Study Results

The proposed BERT-LDA model was applied to all pastes collected from Pastebin, PasteFS, and Pastelink platforms. We manually assigned names to five prevailing topics extracted by our model: (1) hackers, (2) malware, (3) networks, (4) websites, and (5) PII. We present case study results in Table IV. Results are sorted by topic, platform, content type, and relevant terms in the topic.

We identified several noteworthy topics in the case study. First, Topic 1 included two distinct topics relating to hacker discussions and attackers. The keywords "threatinfo" and "channel" refer to hacker forum conversations and threads that might insinuate a cybercriminal's malicious intent. Topic 2 included four major malware types: (1) rootkits, (2) viruses, (3) trojans, and (4) ransomware. We discovered a significant ransomware termed "WannaCry Ransomware" on Pastebin, which is globally known for the major cyberattack on the U.S. National Health Service (NHS) in May of 2017. Topic 3 pertains to exploitable strategies for vulnerable networks and systems. Pastebin and PasteFS contain various Proof-of-Concept (PoC) exploits that could be used by hackers to carry out attacks. We discovered six unique exploits: shellcode, keylogger, application, DoS, DDLi, and memory injection. Similar to Topic 3, Topic 4 includes website application-related exploit content. Two major website application exploits were identified: cross site scripting/cross-site request forgery (XSS/XSRF) and SQL injections (SQLi). Additionally, instantaneous Proof-of-Concept (PoC) exploit code, lists of vulnerable websites, servers, and

databases are uploaded on all three paste sites. Organizations can examine these exploit codes to identify attack, vulnerability, and malicious trends among adversarial actors for proactive CTI. In addition to identifying topics pertaining to malicious actors and their exploits, we also identified three major categories of PII in Topic 5: (1) Stolen SSNs (2) dumps, and (3) carding.

Table 4. TARGETED ANALYSIS AND FINDINGS FROM EXPERIMENTS. NOTE: DDLI = DYNAMIC-LINK LIBRARY INJECTION, DOS = DENIAL OF SERVICE, MI = MEMORY INJECTION, XSS/XSRF = CROSS SITE SCRIPTING/CROSS SITE REQUEST FORGERY, SQLI = SQL INJECTION

Topic	Platform	Content Type	Relevant Terms in Topic
Topic 1: Hackers	Pastebin	Discussion	threadinfo, private, account, recon
		Attacker	terrorist, anon, 0xoosec, identity, luna
Topic 2: Malware	Pastebin	Rootkit	easykit, autoroot, ajakit, hide, binary
		Virus	virus, base, executable, 0x01, 0x1c
		Trojan	njrat, base, spyeye, agenttesla
		Ransomware	wannacry, stealing, access, encrypt
	Pastelink	Rootkit	win32, hide, keys, update, ftp
Topic 3: Network	Pastebin	Shellcode	shell, x01, x1c, x28, inject, function
		Keylogger	key, user, track, steal, credential
		Application	rwrr, void, logs, error, fakeip, patch
		DoS	ip, server, gsmtp, bot, ddos, service
	PasteFS	DDLi	keys, license, firekeys, serialnum
		MI	powershell, encode, infect, obfuscate
		Application	install, logs, user, httpd, transfer, track
		DoS	portmapper, botnet, automatic, tools
Topic 4: Website	Pastebin	XSS/XSRF	script, auth, html, port, proxy, request
		SQLi	database, injection, sqli, queries, char
	PasteFS	XSS/XSRF	socket, proxy, https, browser, portscan
		SQLi	php, bypass, sql, auto, wordpress, url,
	Pastelink	SQLi	sql, load, stack, update, app, true
Topic 5: PII	Pastebin	Stolen SSNs	ssn, cvv, accounts, wallet, sale
	PasteFS	Dumps	fullz, dump, sell, legit, pin, shop, buy
	Pastelink	Carding	fresh, sale, payment, proof, hack, cvv

All three sites included breached and stolen data as well as other sensitive information, which may be associated with key terms such as "ssn," "account," "shop," and "cvv." These sensitive and personal information are often listed on paste sites for sale ensuing a data breach. Illicit users also share dark web links to Darknet Markets and anonymous financial intermediary platforms to

commence sales. Proactively identifying PII on paste sites can help alert organizations of potential breaches to their infrastructure.

Multi-Label Convolutional BiLSTM Transformer (CBT) Model

Our second research framework, Multi-label Transformer, includes three major components as well (Figure. 5): (1) Data Collection and Pre-Processing, (2) Proposed Methodology, and (3) Evaluation & Case Study.

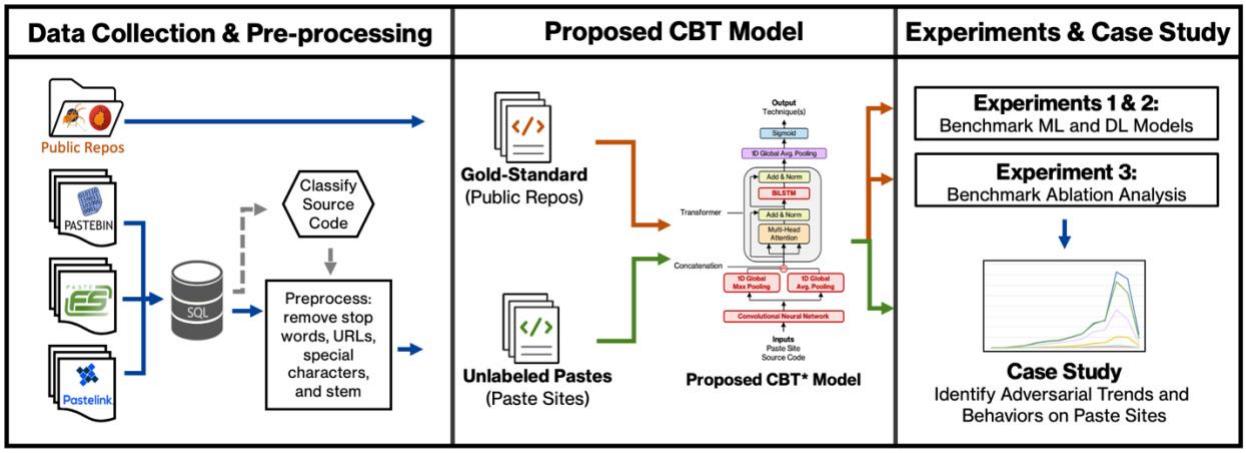

Figure 6. Multi-Label Classification Research Design

Data Collection and Preprocessing

The data collection and preprocessing done in this framework follows the same guidelines as the BERT-LDA framework. To develop a dataset viable for multi-label classification, a professionally vetted dataset is needed. Two sources were identified for collection: Professionally vetted public repositories (e.g., ExploitDB) and Pastebin, a prevailing paste site. Data was processed by filtering posts with python code, converting case-folding, removing special characters, symbols and stop words (Table 2). Additionally, we ran several manual methods (e.g., regular expressions, keyword search, etc.) on the Paste Site dataset to identify python source code, and their data characteristics (table 2).

Table 5. Summary Statistics of Python Source Code in Research Testbed

Data Source	Start-End Date		Python Posts	Avg. Line of Code	Avg. Size of Code	All Modules	Top 3 Modules
Public Repos	4/26/00	10/29/19	959	200	7,884	2,523	sys, socket, requests
Pastebin	8/5/07	1/16/21	23,242	101	3,691	35,511	os, sys, requests
2 Sources	**8/5/07 - 1/16/21**		**24,201**	**151**	**5,788**	**38,034**	**sys, requests, os**

From August 2007 to January 2021, a total of 24,201 posts with python code were collected with Python code, including over 613,976 functions. A single post typically includes 151 lines of code, over 5,000 characters, and 4 imported modules, with "sys," "requests," and "os" being the most frequently used modules. All professionally labeled posts within the public repositories were used to create a multi-class gold-standard dataset for preliminary analysis.

Our gold-standard dataset contains 21 unique ATT&CK Techniques while spanning across 6,128 instances (Table 3). All unique instances belong to at least 1 technique, while only 7 instances belong up to 5 techniques (Table 4). The lengthy sequences of source code on paste sites require a novel method to extract contiguous dependencies and capture locality.

Table 6. Instances with N labels.

1 Technique	2 Techniques	3 Techniques	4 Techniques	5 Techniques
6,017	5,862	114	21	7

Table 7. Distribution Count of Techniques in our Gold-Standard Dataset.

Technique Description	Count
Obfuscated Files or Information	1,921
Hijack Execution Flow	1,658
Peripheral Device Discovery	1,094
Adversary-in-the-Middle	610
Impair Defenses	133
Use Alternate Authentication Material	116
Exploitation for Client Execution	94
Exploitation for Privilege Escalation	74
Exploit Public-Facing Application	71
Valid Accounts	66
Steal or Forge Kerberos Tickets	51
Stage Capabilities	46
Drive-by Compromise	34
Endpoint Denial of Service	33
User Execution	33
Command and Scripting Interpreter	20
Abuse Elevation Control Mechanism	19
File and Directory Discovery	17
Exploitation for Defense Evasion	15
Server Software Component	12
Network Denial of Service	13

Process Injection	11
21 Total Techniques	**6,128**

Proposed Methodology

Our proposed methodology can be broken down into 4 major steps.

1. **CNN**: Pass input data to a Convolutional Neural Network (CNN) layer. The CNN layer comes before the Transformer block and collects low and high-level features as encoded representations without using any positional encodings. The transformer learns from CNN's local feature representation which offers relative positional information required for discovering long-range relationships between local concepts (Mohamed et al., 2020).
2. **Concatenated Pooling (CP):** Pass the CNN output to a Global Max Pooling (GMP) and Global Average Pooling (GAP) for concatenation. Multiple pooling's reduce the sequence of feature vectors and extract context from both inconclusive and fine details of data (Zeng et al., 2019). The $\mathrm{CP}_i = \left(\left[\frac{1}{mxn}\sum_a^m \sum_b^n x_{ab}\right] \frown \left[\max_{(a,b)\in mxn}(x_{ab})\right]\right)$, where (a, b) is a pixel index, c is an index of the channels, m is the number of rows, and n is the number of columns.
3. **Transformer:** Pass concatenated pooling to a transformer block then goes through multi-head attention and normalization layers. Next, we pass the output to a fully connected BiLSTM layer, replacing the FF layer in the standard transformer model. Replacing FF with a BiLSTM layer will help induce internal memory and capture word orders via a feedback loop from input to output (Sukhbaatar et al., 2019).
4. **Output**: Sigmoid activation function produces multi-label outputs

To evaluate how well our proposed methodology will perform in mapping exploit source code to the MITRE ATT&CK CRMF, we conduct several evaluations against

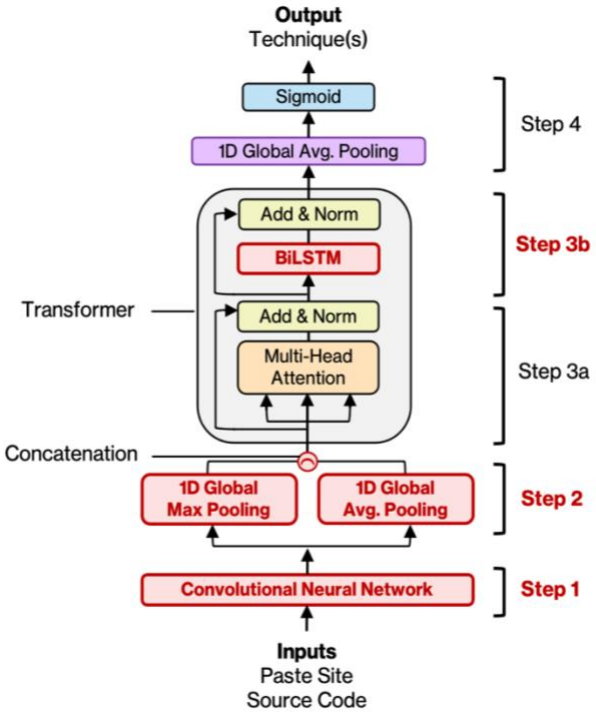

Figure 7. Proposed Convolutional BiLSTM Transformer (CBT) Model

classical machine learning and deep learning models in the next section.

Evaluations

To evaluate our proposed CBT model, we conducted 3 sets of experiments (Table 9): 1) benchmark classical Machine Learning (ML) models, 2) benchmark conventional Deep Learning models, and finally 3) perform an ablation analysis on our proposed model.

Table 8. Description of Experiments 1 and 2, models selected, evaluation metrics used and their justifications.

Evaluation Type	Models	Metrics	Model Justification
Experiment 1: Benchmark Machine Learning Models	Logistic Regression (LR)	Accuracy; Precision Recall; F1-Score; Hamming Loss (Lin et al., 2018).	Riesco et al., 2018; Samtani et al., 2020 Lakhdhar et al., 2021; Kuppa et al., 2021; Ampel et al., 2021; Lin et al., 2018; Guo et al., 2021
	Support Vector Machines (SVM)		
	K-Nearest Neighbor (KNN)		
	Light Gradient Boost Machine (LightGBM)		
	Decision Tree (DT)		
	Random Forest (RF)		
Experiment 2: Benchmark Deep Learning Models	Recurrent Neural Network (RNN)		
	Gated Recurrent Unit (GRU)		
	Long Short-Term Memory (LSTM)		
	Bidirectional Long Short-Term Memory (BiLSTM)		
	Bidirectional Long Short-Term Memory + Attention		
	Transformer		
	Convolutional Neural Network (CNN)		
Experiment 3: Benchmark Ablation Analysis	Transformer w/ Bi-directional Long-Short Term Memory		
	CNN + Transformer w/ BiLSTM		
	CNN + Concat Pooling + Transformer w/ BiLSTM (CBT)		

We used Accuracy, Precision, Recall, F1-Score and Hamming Loss as our evaluation metrics, based on their usage in prior studies. Hamming loss is the fraction of labels that are incorrectly predicted (i.e., lower hamming loss is better) (Lin et al., 2018).

Experiment Results

Table 9 presents benchmark results on Machine Learning (ML) and Deep Learning (DL) models. Our proposed CBT model outperformed all other models in Accuracy (78.57%), Recall (70.62%), F1-Score (79.54%), and Hamming Loss (1.59%), and underperformed against CNN

by 4.9% in Precision. Overall, the proposed CBT model illustrates clear improvements over deep learning-based models (e.g., LSTM, BiLSTM, etc.) and deems most suitable for processing lengthy source code.

Table 9. Results on Benchmark Machine Learning and Deep Learning Models.

Type	Model	Accuracy	Precision	Recall	F1-Score	Ham
Machine Learning	Logistic Regression (LR)	4.06%	43.32%	46.81%	37.42%	13.48%
	Support Vector Machines (SVM)	1.00%	41.85%	20.17%	24.91%	3.72%
	K-Nearest Neighbor (KNN)	10.36%	68.01%	34.09%	40.75%	3.56%
	Light Gradient Boost Machine (LightGBM)	8.63%	70.59%	44.97%	49.37%	3.50%
	Decision Tree (DT)	13.42%	65.58%	42.02%	46.64%	3.41%
	Random Forest (RF)	13.92%	73.82%	45.87%	52.21%	3.36%
Deep Learning	Recurrent Neural Network (RNN)	42.32%	49.34%	20.36%	28.65%	3.82%
	Gated Recurrent Unit (GRU)	57.24%	64.07%	49.19%	55.11%	2.98%
	Long Short-Term Memory (LSTM)	59.35%	71.05%	48.37%	57.42%	2.71%
	Bidirectional Long Short-Term Memory (BiLSTM)	58.57%	67.10%	50.16%	57.46%	2.83%
	Bidirectional Long Short-Term Memory + Attention (BiLSTM + Attention)	66.19%	73.94%	52.13%	61.28%	2.52%
	Transformer	69.27%	73.32%	65.04%	68.96%	2.23%
	Convolutional Neural Network (CNN)	76.29%	**86.39%**	65.04%	73.97%	1.72%
Ablation Analysis	Transformer w/ BiLSTM	69.54%	74.21%	64.82%	69.22%	2.19%
	CNN + Transformer w/ BiLSTM	72.10%	75.42%	61.84%	76.88%	2.01%
	CBT (Ours)	**78.57%**	81.49%	**70.62%**	**79.54%**	**1.59%**

To stay consistent with the sample paste we presented in the introduction of this paper, we fed the DDoS exploit code found on Pastebin to our CBT model to see how well the model maps an exploit to its respective MITRE ATT&CK Techniques. Figure 5 illustrates an end-to-end model comparison between our proposed CBT model and the next best performing model, Convolutional Neural Network (CNN).

Our CBT model correctly mapped the DDoS paste to the Endpoint DoS and Impair Defenses techniques; however, it incorrectly mapped the paste to Obfuscated Files/Information. The CNN model, on the other hand, correctly mapped the DDoS paste to the Develop Capabilities technique and incorrectly mapped it to the Valid Accounts technique.

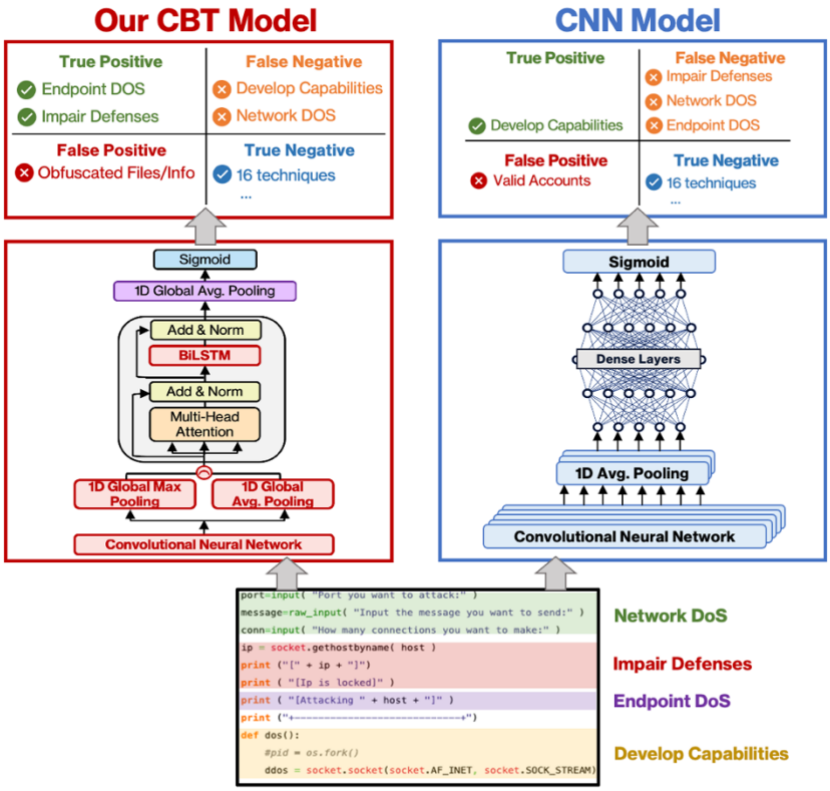

Figure 8. Illustration of Our CBT Model (On the Left in Red) and CNN, The Next Best Performing Model (On Right In Blue)

While false positives yield incorrect information that may be misleading to analysts, having a high fraction of incorrectly predicted labels can be rendered impractical to an organization altogether. In comparison to the CNN model, our model's coupling of a convolutional layer with multiple pooling's and a BiLSTM pair-wise function resulted in less than 2% of error, due to its focus on long-term dependencies of source code.

Case Study Results

The CBT model was applied to Pastebin Python source code to map it to the ATT&CK Techniques. Figure 6 presents the Top 6 MITRE ATT&CK Techniques found on Pastebin from 2008 to 2021. Between 2018 and 2020, the "Obfuscated Files or Information" and "Server

Software Component" techniques, e.g., T1027 and T1505, were predominately shared on Pastebin. Figure 7 illustrates the same top 6 MITRE ATT&CK techniques distributed at the tactic level. The "Defense Evasion", "Privilege Escalation", "Persistence", "Execution", and "Initial Access" were the most common Tactics employed by Pastebin users from 2008 to 2021. Pastebin users mainly used "Exploitation for Client Execution" techniques to exploit software vulnerabilities in client applications while in the Execution tactical lifecycle.

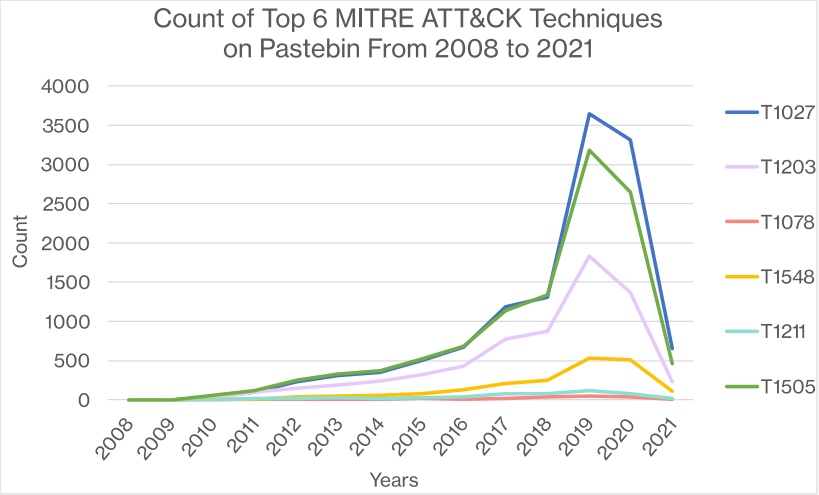

Figure 9. Count of Top 6 MITRE ATT&CK Techniques on Pastebin Over Time.

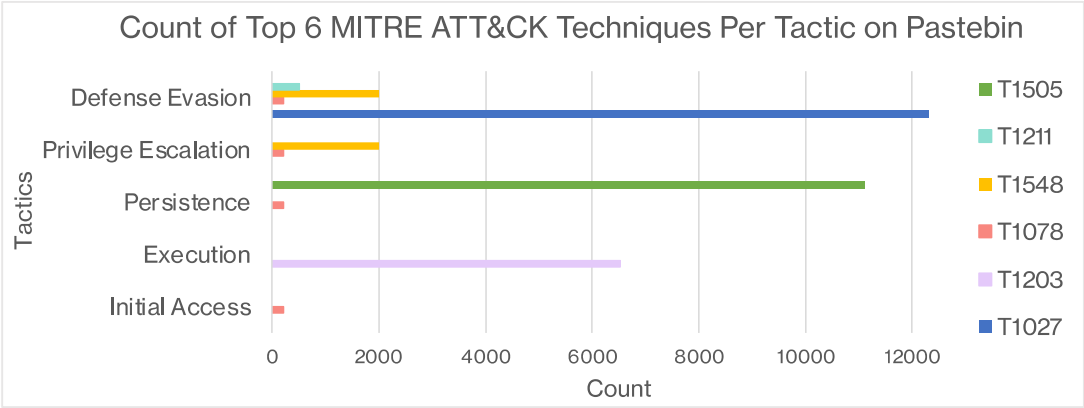

Figure 10. Count of Top 6 MITRE ATT&CK Techniques per Tactic on Pastebin

Table 7 delves into the technical details of each MITRE ATT&CK Tactic discovered on Pastebin. Among all Tactics, Pastebin users frequently leveraged command lines (e.g., PowerShell, Kali), remote access tools (e.g., njRAT, NinjaOne), vulnerability scanners (e.g., nmap, Nessus), and other open-source projects (e.g., MimiKatz, Hulk) as their primary exploit tools. The top Python libraries and modules used among Pastebin users are Socket, Requests,

URLLib, Scapy, PTY, SubProcess, and TelnetLib. Among those python libraries and modules, the most common functions used are httplib, parse, path, decode, unhexlify, sock stream, and SSHClient. Overall, our case study demonstrated the various types of tactics, techniques, and procedures (TTP) available on Pastebin that may assist organizations in improving their prevention and detection capabilities.

Table 10. Technical Details of All Tactics Found on Pastebin Source Code.

Tactic	Top Tools	Top Library & Modules	Top Functions	Sample Code Snippet

Defense Evasion	PowerShell, VBA, netstumbler	struct, pty, socket, shutil, tornado, requests, scapy	sock_stream, error copytree, ioloop	`def accessAdminShare(computerName,executable):`
				`remote = r"\\"+computerName+"\c$"`
				`local = "Z:"`
				`remotefile = local + "\\"+executable`
				`os.system("net use "+local+" "+remote)`

Persistence	PowerSploit, PowerShell	base64, codecs, socket, urllib, scapy, request	httplib, decode, stdout,run AF_Packet	`if p.haslayer(IP):` `decoyIP = [ip for ip in [p[IP].src, p[IP].dst] if ip in decoys]`
Execution	Remote access tools (e.g., njrat)	struct, os, zipfile, binascii, shutil, random	cookielib, unhexlify, decode, unpack	`if os.system("schtasks /query /tn SecurityScan") == 0:` `os.system("schtasks /delete /f /tn SecurityScan")`

Privilege Escalation	beroot, hashcat, malware	base64, codecs, pty, zipfile, os, requests	optionparser, decode, fork, path	`def do_GET(self):`
				`queries = parse_qs(urlparse(self.path).query)`
				`print("Username: %s, Password: %s"%(queries["u`
				`self.send_response(300)`

Initial Access	nessus, netsparker	paramiko, telnetlib, pty, codecs, telnetlib	globals, httplib, path, paramiko	`conn = sqlite3.connect(firefoxPath)`
				`c = conn.cursor()`
				`c.execute("SELECT * FROM moz_cookies")`

Collection	nmap, kismet, intruder	beautifulsoup, urllib, re, socket, scapy, requests	prettify, stdout, get, path, parse	`while True:`
				`spoofer(targetIP,gatewayIP)`
				`spoofer(gatewayIP,targetIP)`
				`print("\r[+] Sent packets "+ str(packets)),`
				`sys.stdout.flush()`

Credential Access	ettercap, mimikatz	urllib, subprocess, socket, os, requests	httpdigestauth, run, parse, devnull	`if cookie[4].endswith(domain) and cookie[2]` `print("%s %s %s" % (cookie[4], cookie[2]`

Impact	Slowloris, nikto, hulk	psutil, sys, scapy, os, socket, requests, parakimo	SSHClient, invoke_shell, hexdigits, parakimo	`try:`
				`ddos.connect((host, 80))`
				`ddos.send( message )`
				`ddos.sendto( message, (ip, port) )`

Resource Development	development environments	socket, re, requests, urllib, subprocess	get_host, get_ip, port, request, parse	`def dos():`
				`#pid = os.fork()`
				`ddos = socket.socket(socket.AF_INET,`

Conclusion and Future Directions

Malicious cyber activities exact significant financial costs upon the global economy. Hackers are increasingly using social media platforms such as paste sites to share malicious content for cyber-attacks and breaches. In this study, we proposed two novel methods to automatically extract major themes of malicious content from prevalent paste sites and map them to the MITRE ATT&CK CRMF for potential CTI purposes.

BERT-LDA operates by replacing the BoW used in the standard LDA with a BoL created by BERT that includes class labels at the sequence level. BERT-LDA outperforms the conventional LDA and BERT-LDA variants in terms of perplexity across three paste sites vs other PTLMs.

We illustrated that adding A CNN layer placed before the Transformer block to extract and learn relative locality information, multiple pooling to reduce the feature vector space and capture low and high-level features, and finally, replacing the feed-forward pair-wise function with a BiLSTM can capture word/sequence orders and provide internal memory. Finally, we exhibited the potential practical utility of the proposed models in two case studies that identified significant hacker community activity, malicious and destructive code, network, and website vulnerabilities, as well as PII from prevailing paste sites. Organizations can use the findings of this study to proactively reduce potential damage on their infrastructure.

There are three promising directions for future study. First, future studies may concentrate on using key post metadata to perform threat trend analysis and detect emerging trends. Second, we can adapt deep transfer learning strategies implemented in prior CTI research to automatically classify the malicious code discovered by BERT-LDA. Third, we can leverage knowledge distillation methods to map unseen exploit code to multiple techniques under different distilled tactics. Each direction can help provide more comprehensive CTI to combat emerging cyber-attacks.

Acknowledgement

This work was supported in part by the National Science Foundation under grant numbers DGE-1921485 (SFS), OAC-1917117 (CICI), and CNS-1850362 (SaTC CRII).

References

Barbieri, Nicola & Manco, Giuseppe & Ritacco, Ettore & Carnuccio, Marco & Bevacqua, Antonio. (2013). Probabilistic topic models for sequence data. Machine Learning. 93, 5–29.

Benites, F., & Sapozhnikova, E.P. (2015). HARAM: A Hierarchical ARAM Neural Network for Large-Scale Text Classification. 2015 IEEE International Conference on Data Mining Workshop (ICDMW), 847-854.

Benjamin, Victor & Valacich, Joseph & Chen, Hsinchun. (2019). DICE-E: A Framework for Conducting Darknet Identification, Collection, Evaluation with Ethics. MIS Quarterly. 43. 1-22.

Blei, David & Ng, Andrew & Jordan, Michael. (2013). Latent Dirichlet Allocation. Journal of Machine Learning Research. 3. 993.

Brengel, Michael & Rossow, Christian. (2018). Identifying key leakage of bitcoin users. In International Symposium on Research in Attacks, Intrusions, and Defenses. 623–643.

Chai, Yidong & Li, Weifeng (2019). Towards Deep Learning Interpretability: A Topic Modeling Approach. International Conference on Information Systems, ICIS 2019. 1-10.

Chen, Tse-Hsun Peter & Thomas, Stephen & Hassan, Ahmed E.. (2015). A survey on the use of topic models when mining software repositories. Empirical Software Engineering, ESS. 21.

Conneau, Alexis & Khandelwal, Kartikay & Goyal, Naman & Chaudhary, Vishrav & Wenzek, Guillaume & Guzman, Francisco & Grave, Edouard & Ott, Myle & Zettlemoyer, Luke & Stoyanov, Veselin. (2020). Unsupervised Cross-lingual Representation Learning at Scale. The Association for Computational Linguistics, ACL. 8440-8451.

Devlin, Jacob & Chang, Ming-Wei & Lee, Kenton & Toutanova, Kristina. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Association for Computational Linguistics, ACL. 4171—4186.

Dieng, Adji & Wang, Chong & Gao, Jianfeng & Paisley, John. (2016). TopicRNN: A Recurrent Neural Network with Long-Range Semantic Dependency. 5th International Conference on Learning Representations, ICLR. 1-13.

Eckhardt, Jennifer & Kaletka, Christoph & Pelka, Bastian. (2018). Copy Here, Paste There? On the Challenges of Scaling Inclusive Social Innovations. Universal Access in Human-Computer Interaction. Methods, Technologies, and Users. 50-62.

Eseryel, U. & Wie, Kangning & Crowston, Kevin. (2020). Decision-making Processes in Community-based Free/Libre Open Source Software-development Teams with Internal Governance: An Extension to Decision-making Theory. Communications of the Association for Information Systems. 484-510.

Geva, M., Schuster, R., Berant, J., & Levy, O. (2021). Transformer Feed-Forward Layers Are Key-Value Memories. ArXiv, abs/2012.14913.

Gulati, A., Qin, J., Chiu, C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., & Pang, R. (2020). Conformer: Convolution-augmented Transformer for Speech Recognition. ArXiv, abs/2005.08100.

Guo, Y., Liu, J., Tang, W., & Huang, C. (2021). Exsense: Extract sensitive information from unstructured data. Comput. Secur., 102, 102156.

Hemberg, E., Kelly, J., Shlapentokh-Rothman, M., Reinstadler, B., Xu, K., Rutar, N., & O'Reilly, U. (2020). BRON - Linking Attack Tactics, Techniques, and Patterns with Defensive Weaknesses, Vulnerabilities and Affected Platform Configurations. ArXiv, abs/2010.00533

Imai, H., & Kanaoka, A. (2018). Time Series Analysis of Copy-and-Paste Impact on Android Application Security. (2018) 13th Asia Joint Conference on Information Security.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. EMNLP.

Kontaxis, Georgios & Polakis, Iasonas & Ioannidis, Sotiris. (2011). Outsourcing Malicious Infrastructure to the Cloud. SysSec Workshop (SysSec). 1-8.

Kuppa, A., Aouad, L.M., & Le-Khac, N. (2021). Linking CVE's to MITRE ATT&CK Techniques. The 16th International Conference on Availability, Reliability and Security.

Lakhdhar, Y., & Rekhis, S. (2021). Machine Learning Based Approach for the Automated Mapping of Discovered Vulnerabilities to Adversarial Tactics. 2021 IEEE Security and Privacy Workshops (SPW), 309.

Li, P., Zhong, P., Mao, K., Wang, D., Yang, X., Liu, Y., Yin, J., & See, S. (2021). ACT: an Attentive Convolutional Transformer for Efficient Text Classification. AAAI.

Li, Weifeng & Chen, Hsinchun & Nunamaker, Jay F. Jr. (2016a) Identifying and Profiling Key Sellers in Cyber Carding Community: AZSecure Text Mining System, Journal of Management Information Systems, 33:4, 1059-1086.

Li, Weifeng & Yin, Junming & Chen, Hsinchun. (2016b). Targeting key data breach services in underground supply chain. IEEE International Conference on Intelligence and Security Informatics, ISI. 322-324.

Lin, J., Su, Q., Yang, P., Ma, S., & Sun, X. (2018). Semantic-Unit-Based Dilated Convolution for Multi-Label Text Classification. ArXiv, abs/1808.08561.

Lin, M., Chen, Q., & Yan, S. (2014). Network In Network. CoRR, abs/1312.4400.

Liu, S., Zhang, L., Yang, X., Su, H., & Zhu, J. (2021). Query2Label: A Simple Transformer Way to Multi-Label Classification. ArXiv, abs/2107.10834.

Mohamed, A., Okhonko, D., & Zettlemoyer, L. (2019). Transformers with convolutional context for ASR. ArXiv, abs/1904.11660.

Morgan, Steven. "Cybercrime to Cost the World $10.5 Trillion Annually by 2025." Cybercrime Magazine, 27 Apr. 2021, https://cybersecurityventures.com/cybercrime-damages-6-trillion-by-2021/.

Nam, J., Kim, J., Mencía, E.L., Gurevych, I., & Fürnkranz, J. (2014). Large-Scale Multi-label Text Classification - Revisiting Neural Networks. ArXiv, abs/1312.5419.

OCCUPY4ELES. (2015, August 11). How To: Easily Find an Exploit in Exploit DB and Get It Compiled All from Your Terminal. WonderHowTo. https://null-byte.wonderhowto.com/how-to/easily-find-exploit-exploit-db-and-get-compiled-all-from-your-terminal-0163760/

Pakhari, Muhammad & Jamil, Norziana & Rusli, Mohd & Rahim, Azril. (2020). Implementation of Token Parsing Technique for Regex Based Classification of Unstructured Data for Cyber Threat Analysis. International Conference on Mobile Computing and Ubiquitous Networking. 395-398.

Paul, J. A., & Wang, X. J. (2019). Socially optimal IT investment for cybersecurity. In Decision Support Systems, 122, 113069.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. NAACL.

Qiu, Xipeng & Sun, Tianxiang & Xu, Yige & Shao, Yunfan & Dai, Ning & Huang, Xuanjing. (2020). Pre-trained models for natural language processing: A survey. Science China Technological Sciences. 63. 1872-1897.

Radford, Alec., & Narasimhan, Karthik & Salimans, Tim & Sutskever, Ilya. (2018). Improving Language Understanding by Generative Pre-Training. Computer Science. 12.

Rezaee, Mehdi & Ferraro, Francis. (2020). A Discrete Variational Recurrent Topic Model without the Reparameterization Trick. Neural Information Processing Systems, NeurIPS. 1-16.

Riesco, Adrián & Fidalgo, Eduardo & Al Nabki, Wesam & Jáñez-Martino, Francisco & Alegre, Enrique. (2019). Classifying Pastebin Content Through the Generation of PasteCC Labeled Dataset. Lecture Notes in Computer Science Hybrid Artificial Intelligent Systems, 2019, p. 456-467.

Samtani, S., Abate, M., Benjamin, V.A., & Li, W. (2019). Cybersecurity as an Industry: A Cyber Threat Intelligence Perspective.

Samtani, S., Zhu, H., & Chen, H. (2020a). Proactively Identifying Emerging Hacker Threats from the Dark Web. ACM Transactions on Privacy and Security (TOPS) 1 - 33.

Samtani, Sagar & Kantarcioglu, Murat. (2020b). Trailblazing the Artificial Intelligence for Cybersecurity Discipline: A Multi-Disciplinary Research Roadmap. ACM Transactions on Management Information Systems. 11. 1-18.

Samtani, Sagar & Zhu, Hongyi. (2020c). Proactively Identifying Emerging Hacker Threats from the Dark Web: A Diachronic Graph Embedding Framework (D-GEF). ACM Transactions on Privacy and Security. 23. 1-33.

Sanh, Victor & Debut, Lysandre & Chaumond, Julien & Wolf, Thomas. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. Neural Information Processing Systems, NeurIPS. 1-5.

Squire, Megan & Smith, Amber. (2015). The Diffusion of Pastebin Tools to Enhance Communication in FLOSS Mailing Lists. Open Source Systems. 451. 45-57.

Steinbeck, M., & Koschke, R. (2021). Javadoc Violations and Their Evolution in Open-Source Software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 249.

Strom, B. E., Battaglia, J. A., Kemmerer, M. S., Kupersanin, W., Miller, D. P., Wampler, C., ... & Wolf, R. D. (2017). Finding cyber threats with ATT&CK-based analytics. MITRE, Technical Report No. MTR170202.

Sukhbaatar, S., Grave, E., Lample, G., Jégou, H., & Joulin, A. (2019). Augmenting Self-attention with Persistent Memory. ArXiv, abs/1907.01470.

Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. International Journal of Data Warehousing and Mining (IJDWM), 3(3), 1-13.

Vahedi, T., Ampel, B., Samtani, S. & Chen, H. (2021). Identifying and Categorizing Malicious Content on Paste Sites: A Neural Topic Modeling Approach. 2021 IEEE International Conference on Intelligence and Security Informatics (ISI), 1-6.

Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. ArXiv, abs/1706.03762.

Walkhowski, D. (2021). MITRE ATT&CK: What It Is, How it Works, Who Uses It and Why. Accessed from https://www.f5.com/labs/articles/education/mitre-attack-what-it-is-how-it-works-who-uses-it-and-why. Nov. 29, 2021.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2019). HuggingFace's Transformers: State-of-the-art Natural Language Processing. ArXiv, abs/1910.03771.

Yang, B., Tu, Z., Wong, D.F., Meng, F., Chao, L.S., & Zhang, T. (2018). Modeling Localness for Self-Attention Networks. EMNLP.

Yang, B., Wang, L., Wong, D.F., Chao, L.S., & Tu, Z. (2019). Convolutional Self-Attention Networks. NAACL.

Yu, A.W., Dohan, D., Luong, M., Zhao, R., Chen, K., Norouzi, M., & Le, Q.V. (2018). QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. ArXiv, abs/1804.09541.

Yun, C., Bhojanapalli, S., Rawat, A.S., Reddi, S.J., & Kumar, S. (2020). Are Transformers universal approximators of sequence-to-sequence functions? ArXiv, abs/1912.10077.

Zeng, H., Wang, Q., Li, C., & Song, W. (2019). Learning-Based Multiple Pooling Fusion in Multi-View Convolutional Neural Network for 3D Model Classification and Retrieval. J. Inf. Process. Syst., 15, 1179-1191.

Zhang, Y., & Wang, T. (2021). CCEyes: An Effective Tool for Code Clone Detection on Large-Scale Open-Source Repositories. 2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE), 61-70.