Exploring topic models to discern zero-day vulnerabilities on Twitter through a case study on log4shell

Yue Wang, Md Abul Bashar, Mahinthan Chandramohan, and Richi Nayak

***Abstract*—Twitter has demonstrated advantages in providing timely information about zero-day vulnerabilities and exploits. The large volume of unstructured tweets, on the other hand, makes it difficult for cybersecurity professionals to perform manual analysis and investigation into critical cyberattack incidents. To improve the efficiency of data processing on Twitter, we propose a novel vulnerability discovery and monitoring framework that can collect and organize unstructured tweets into semantically related topics with temporal dynamic patterns. Unlike existing supervised machine learning methods that process tweets based on a labelled dataset, our framework is unsupervised, making it better suited for analysing emerging cyberattack and vulnerability incidents when no prior knowledge is available (e.g., zero-day vulnerability and incidents). The proposed framework compares three topic modeling techniques(Latent Dirichlet Allocation, Non-negative Matrix Factorization and Contextualized Topic Modeling) in combination of different text representation methods (Bag-of-word and contextualized pre-trained language models) on a Twitter dataset that was collected from 47 influential users in the cybersecueity community. We show how the proposed framework can be used to analyse a critical zero-day vulnerability incident(Log4shell) on Apache log4j java library in order to understand its temporal evolution and dynamic patterns across its vulnerability life-cycle. Results show that our proposed framework can be used to effectively analyse vulnerability related topics and their dynamic patterns. Twitter can reveal valuable information regarding the early indicator of exploits and users behaviours. The pre-trained contextualized text representation shows advantages for the unstructured, domain-dependent, sparse Twitter textual data under the cybersecueity domain.**

***Index Terms*—Social media analysis, topic modeling, cyber threat intelligence, zero-day vulnerability, text representation, language models**

I. Introduction

THE rapid expansion of digital products has led to a significant increase in software vulnerabilities and corresponding incidents. Recently (Dec, 2021), log4shell and its related vulnerabilities allow attackers to remotely execute malicious code on target systems that resulted in massive financial loss to organizations. To combat increasing cyberthreats, organisations need to be abreast of Cyber threat intelligence (CTI) about vulnerabilities, exploits, incidents, and available countermeasures to stay informed about emerging threats that may pose risks to their IT products and infrastructures.

National Vulnerability Database (NVD)[1] [1] is a well-established structured database to investigate, announce, and organize new vulnerabilities, which serve as a general standard for organizations in prioritizing vulnerability remediation activities. However, recent studies have indicated that NVD is not always up-to-date [2], with newly discovered vulnerabilities discussed on social media platforms(e.g., Twitter) often long before the NVD public disclosure [3]. This trend can grant a small window of time advantage for cybersecurity professionals to early discover exploitable vulnerabilities by analysing Twitter discussions among the cybersecurity community [4].

Twitter users in the CTI community can serve as social sensors that monitor real-time cyberattack incidents and exploit incidents [5]. Analyzing the rich source of CTI data from Twitter can help organisations improve their risk assessment and incident response procedures by revealing evidence of attack actions, attack patterns, and people's perceptions [6]. However, such evidence and patterns are usually hidden in the large volume of unstructured tweets. Manually analyzing and organizing such unstructured information is time-consuming and nearly impossible. In the CTI application domain, automating the process to ease the information overload of cybersecurity professionals for processing massive unstructured documents is not a want but a necessity.

Supervised machine learning approaches have been proposed to automatically filter CTI related information on Twitter. Given the labelled training dataset, a supervised classifier can be built to collect cyberthreat indicators from Titter stream [7]. Alternatively, a supervised Named Entity Recognizer [8] can be built to extract Indicator of Compromise from Tweets. Such methods mostly rely on a substantial number of labelled data and static features. However, the cybersecurity landscape is ever-evolving and threat patterns are constantly changing [2]. When new cybersecurity incidents emerge (e.g. zero-day [2] vulnerability), the CTI community is mostly unaware or have little knowledge about them [9]. Consequently, a labelled training data is impractical to obtain.

This work was supported by the Oracle Labs, Australia and the QUT Centre for Data Science, Queensland University of Technology

Yue Wang is with the School of Computer Science and QUT Centre for Data Science, Queensland University of Technology, Brisbane 4000, Australia, and also with Oracle Labs, Brisbane, 4000, Australia (e-mail:y355.wang@hdr.qut.edu.au)

Md Abul Bashar and Richi Nayak are with the School of Computer Science and QUT Centre for Data Science, Queensland University of Technology, Brisbane, 4000, Australia (e-mail:m1.bashar@qut.edu.au; r.nayak@qut.edu.au)

Mahinthan Chandramohan is with Oracle Labs, Brisbane, 4000, Australia (e-mail:mahin.chandramohan@oracle.com)

[1] https://nvd.nist.gov/

[2] Zero-day vulnerability refers to a software vulnerability that is unknown to vendors before disclosure. An exploit that targets zero-day vulnerability is called zero-day exploit.

In other words, supervised methods are incapable to discover zero-day vulnerabilities and exploits.

Topic Modelling techniques are unsupervised text mining methods for performing exploratory analysis and discovering hidden patterns from a collection of documents. This is the exact use case in the application of conducting cyberattack incident analysis, where there is little or no prior knowledge of the data. The idea behind topic modelling is that documents can be thought of as a collection of topics, each of which is interpreted as a cluster of words describing a specific semantic meaning [10]. During the incident investigation, topics serve as high-level summaries of the documents, greatly facilitating the efficiency of document processing and analysis. Based on the topic of interest, cybersecurity professionals can limit the scope and prioritise the investigation.

Topic modeling techniques have a long history. Various topic models, such as Non-negative Matrix Factorization (NMF), Latent Dirichlet Allocation (LDA), and the emerging Neural Topic Models(NTM), have been used in generic social media mining tasks. However, their applications in cybersecurity have yet to be investigated thoroughly in this emerging research field of "vulnerability tracking via social media platforms". In this paper, we address the gap by proposing a novel vulnerability discovering and monitoring framework to assist cybersecurity professionals to collect, process and organise unstructured tweets into interpretable topics for conducting incident analysis. Traditional topic models, in particular, are assumed to be static; however, in this study, we investigate the dynamic representation of discovered topics. Given a self-defined time frame, cybersecurity professionals can select security concerned topics and inspect their evolution overtime. The dynamic representation of topics enables cybersecurity professionals to easily understand how cyberattack incidents emerge, evolve, and decay over time, revealing dynamic patterns and evidences. The proposed framework incorporates three distinct topic modelling techniques: NMF, LDA, and CTM. These techniques were run with different combinations of text representations(Bag-of-words and contextualized representation) to compare the relative performance on the studied domain. We demonstrate the framework's feasibility with a case study on log4shell, which is a critical zero-day vulnerability incident that occurred in December 2021. The dataset (dubbed the log4shell dataset) was gathered from a group of Security Information Providers (SIPs) [1] on Twitter during the lifecycle of the log4shell incident, when it was first discovered, exploited, and disclosed. Experiments on the log4shell dataset are conducted comparing various topic modeling techniques to understand the performance of topic modeling methods, and the vulnerability and threat patterns that occur in Twitter conversations.

We present the selected dynamic topic representation of log4shell against the NVD public disclosure date. Experiments show that (1) the proposed framework can be effectively used for investigating and analysing cybersecurity incidents and their dynamic patterns. (2) Twitter can reveal valuable information regarding early exploitable patterns, attackers' strategies and behaviours. Emerging threat patterns were identified on Twitter before the well-established NVD announcement. This shows the potential of proposed framework in early discover vulnerability incidents and even zero-day vulnerabilities.

To our best of knowledge, this is the first work to propose a vulnerability discovery and monitoring framework based on topic modelling. More specifically, the contribution of this study is listed below:

- We propose a novel unsupervised vulnerability discovery and monitoring framework that aids cybersecurity professionals to effectively collect, analyze and organise unstructured tweets into interpretable dynamic topics for analyzing vulnerability incidents and cyberthreat monitoring. Results of a case study on log4shell vulnerability show that Twitter is a timely information source for cyberthreat identification compared with the well-established NVD announcements.
- Several topic modleing techniques with varied text representation methods are compared on the Log4j dataset. Specifically, the study investigated classical topic modeling techniques (NMF and LDA) with Bag-of-word representation and Neural Topic models with the state-of-the-art contextualized representation from BERT [11]. Evaluation shows that NTM found more coherent, diverse and informative topics which is more suitable for the short, sparse, noisy and domain-dependent CTI data on Twitter.
- A case study on log4shell was conducted to investigate how vulnerability related information is diffused based on the vulnerability life cycle. We analyze the unique characteristics of Twitter data in cybersecurity community to identify common patterns.

II. Related studies and the background

A. Cyberthreat and vulnerability analysis via Twitter

Twitter's near-real-time nature makes it an ideal information source to investigate critical social events [12]. In the CTI domain, Twitter has been used to analyze the malicious activities such as spam, botnets, phishing, fake Retweet [13] and data manipulation [14] [15]. There are only a few works that investigate social media data for the purpose of analysing software vulnerabilities and related incidents.To uncover the state of a vulnerability and the intrusions behind, authors [16] [17] [18] proposed a well-known vulnerability life-cycle model that conclude a vulnerability into the black risk phase, gray risk phase and white risk phase three stages. Sauerwein et al. in [2] conducted an empirical analysis of zero-day vulnerabilities by mapping a large collection of tweets and its CVE mentions to the vulnerability lifecycle model. They discovered that one quarter of the examined zero-days were discussed on Twitter before public disclosure by NVD. Similarly, Shrestha et al. [19] conducted network topology analysis to understand how discussions about software vulnerabilities spread on social platforms. They discovered that highly severe vulnerabilities have significantly deeper, broader and more viral discussions on Twitter. Despite the proven advantages of vulnerability information shared on Twitter, little attention has been paid on how to collect, process and organize these large volume of tweets effectively. The information collection and analysis

process has been commonly done by keyword search and manual investigation which are labour intensive and time consuming.Our research fill the gap by proposing end-to-end framework to ease the workload of cybersecurity professionals to automate the workflow for social media analysis to track vulnerabilities.

Using the labelled data, supervised machine learning approaches can assist cybersecurity professionals in automatically distinguishing threat related and non-threat tweets based on either engineered features or predefined patterns [20] [21]. Common supervised machine learning algorithms such as support vector machine (SVM) [22], random forest [3], linear regression and Naive Bayes [23] have achieved threat detection accuracy of up to 90% on tweets on features such as regular text patterns (CVE, IP address), account status (active or compromised), user interaction (Retweets, follower/following relationship), and so on. These methods relies on a large number of labelled training datasets to learn the data patterns which poses a difficulty for event detection (i.e. vulnerability) on Twitter as CTI topics are constantly changing over time [24]. Distinct from these works, we leverage pure text feature learnt from the raw tweets through Natural Language Processing techniques without any labelled information. This is more suitable in the scenario of analyzing emerging cybeattacks and vulnerability incidents when little background information in known.

B. Cyberthreat detection with unsupervised methods

Unsupervised methods such as topic modelling and clustering can discover the intrinsic textual features and group them together, which is desirable in the CTI analysis of online social media conversations [25] [26]. An early work investigated multiple document clustering algorithms for computer forensic analysis [27]. Nassif et al. compared six representative clustering algorithm to process large amount of unstructured document for forensic analysis. This work shares a similar objective with us, but there are fundamental differences between clustering and topic modeling techniques, as there is no concept of "topics" in clustering. Similarly, a hierarchical clustering-based approach was used to detect cyber threat events [28]. Huang et al. [29] explored LDA to gain an understanding of threat and vulnerability related discussion on Twitter. These two approaches did not explore the dynamic analysis of the discovered threat events as we do. Liu et al. proposed CyberEM, an event evolution model for finding cybersecurity events from tweet. CyberEM uses NMF with term-frequency–inverse-document-frequency (TF-IDF) features to discover and aggregate cybersecurity events across multiple time intervals from Twitter [6]. Although this work discovers dynamic associations between various events, it does not shows how an event evolves over time. Different from CyberEM, our proposed framework generates dynamic topic representations at each time-step, allowing topics vary smoothly over time. Such linear representation are useful for investigating when and how attack patterns emerge, evolve and decay.

Moreover, the above discussed methods use the vector space model (i.e Bag-of-Words) for text representation which have inherent limitation with unstructured, high-dimensional and short social media data under the CTI domain. The complexity of the CTI data will be discussed next.

C. The social media Data Challenges

Many researchers have addressed the challenges of handling social media text in the general text mining domain.It is reported to be worse with social media data under the CTI domain due to unique characteristics and traits [30] [24].

Social media text is a common type of user-generated data that is known to be short, unstructured, sparse, and likely to contain many "noisy" patterns such as emoji, HTML, misspelling and symbols.Such data generally require sophisticated pre-processing steps before machine learning techniques can be applied, for example, stemming, lemmatilization, stopword removal etc. Our explonary analysis with CTI tweets further reveal that CTI data contain significant domain specific words that require expert knowledge to understand the context. As an example, Cyber security content typically includes a large number of technical terms related to software, IT infrastructure, networking, and programming. Abbreviations and distinctive naming conventions are also prevalent. The term "dirty cow" refers to a "copy-on-write" security vulnerability, and "XXS" refers to a "cross-site scripting vulnerability" in cyber security. Without understanding the domain context, terms like "BlackCat", "WannaCry" and "Bad Rabbit" [31] [32] [5] are unlikely to be associated with ransomware attacks. These terminologies can lead to ambiguity and be a significant barrier to machine learning's ability to learn effective text features and their underlying semantics.

D. Text representation learning for topic models

Text representation learning is the study of converting raw text into numerical features in order to facilitate effective machine learning techniques, which is an important component of Natural Language Processing (NLP) [33]. Traditional topic modeling (e.g. LDA and NMF) methods generally use bag-of-words(BoW) to learn the text representation which are calculated purely based on word occurrence and frequency [33]. Such text representation methods are criticized for ignoring word order and text semantics [34]. Given the domain-dependent, noisy, and low-occurrence nature of CTI textual data, the BoW representation is found less-effective in social media analysis [35]. For example, in BoW representation, "vulnerable" and "vulnerabilities" are considered as two different features despite their similar meaning, unless stemming and lemmatization are performed to unify the word's morphological variations [36]. However, this processing becomes computational expensive for large datasets as well as it losses expressiveness of the data.

With the recent success of deep neural networks, distributed text representations have gained popularity due to their ability to capture context in documents [33]. The most well-known architecture in this category, Bidirectional Encoder Representations from Transformers (BERT) [11], enriches the contextualised feature representation by pre-training from large-scale corpora such as Wikipedia and the Book Corpus.

The use of BERT and its derived language models as input has improved state-of-the-art performance in a wide range of text mining tasks, including classification and topic modelling [36] [37].

BERT and its variations can introduce three benefits into the topic models. (1)BERT learns bidirectional, semantic word relations to better capture the contextualized meaning from the document. (3) Contextualized representation from pre-trained language models can reduce the need for extensive data pre-processing (e.g., stemming, lemmatization, stop-word removal etc.). This is because pre-training is commonly performed on raw text. (2)There are numerous pre-trained language models that have been fine-tuned for various domains and genres from which we can select the one that best suits our needs. For example, SentenceBERT [38] is optimized for semantic search and sentence embedding while CyBERT is a language model for cybersecurity domain [39]. While the state-of-the-art text representation methods claim to lead better results in several down stream text mining tasks, they have not yet been fully explored in the domain of CTI. Considering the unique characteristics of CTI data, there exists no in-depth comparative study on the task of topic modeling. We address this gap by exploring various topic modelling approaches in combine with different text representation methods (e.g. BoW, TF-IDF, BERT). We conduct extensive experiments on a real-world case study of log4shell vulnerability. To the best of our knowledge, this is the first study that extensively investigates topic modelling techniques for understanding cyberthreat events on social media.

III. The Vulnerability Discovery and Tracking Framework

This section introduces the proposed framework and its workflow, as depicted in Figure 1. This framework consists of four phases: (1) Data Collection, (2) Preprocessing, (3) Topic Modeling and (4) Topic Analysis. The framework begins by collecting Twitter data generated by a list of influential users over a defined time period. The collected data is pre-processed and transformed into different feature representation (BoW, TF-IDF, BERT) for topic modelling. Various topic modelling methods are used to identify topics, which are then thoroughly evaluated using defined metrics. Finally, the topic insights and dynamic visualisations will be presented and analysed. The insights into the topics can help the cybersecurity professionals identify how a cyberthreat-related topic emerges, evolves, and fades over time, allowing them to prioritise patching resources and plan risk mitigation strategies. In the following subsections, we will go over each phase in detail.

A. Data collection

In social media text analysis, data (i.e. tweets) can be collected by two approaches.The first method involves collecting tweets that contain a set of pre-defined search keywords. However, this method is prone to overlooking new emerging keywords and surrounding topics. The second method is to find a group of users and collect their tweets. This approach ensures a good coverage of topics that may related to potential unknown threats and vulnerabilities. This framework adopts the second approach.

Vulnerability information is frequently shared on Twitter by a group of cybersecurity professionals. They use Twitter as a crowd-sourcing platform [19] to discuss vulnerability descriptions, exploit demonstrations, and potential counter-measuresFrei et al. defined this group of people as Security Information Providers (SIP) [1] who play critical roles in the cybersecurity ecosystem by gathering and disseminating timely security news. The decision to collect data from SIP accounts ensures that we have a consistent, high-quality, reliable, and accountable source of data for further analysis.

We identified 47 SIP Twitter users with a significant number of followers for data collection from the Twitter timeline API [3]. These user are identified based on mutual followers, and

[3]https://developer.twitter.com/

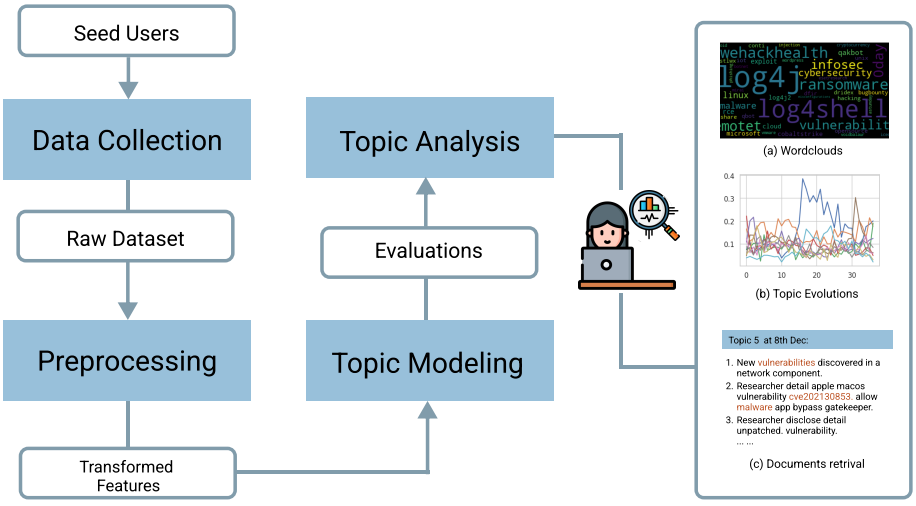


Fig. 1: The proposed social media vulnerability tracking framework

validated manually with the help of two cybersecurity experts. They include cybersecurity researchers, threat hunters, OSINT (open-source intelligence) investigators, malware researchers and CTI projects. This user pool can be expanded or customized. A community detection algorithms [40] can assist with automatically identify the user pool, but it is beyond the scope of this study.

B. Preprocessing

Data preprocessing involves several key steps to clean data and transform raw text into vectors with text representation methods. The procedures are as follows:

1) Tweets are arranged based on time sequence.
2) Special characters, including punctuation, symbols, URLs are removed. Twitter symbols such as "", "#", "RT" and emojis are removed using regular expressions. We further removed file hashes, IP addresses and file signatures, which are low level Indicator-of-Compromises commonly found in CTI tweets.
3) We perform Tokenization, lemmatisation and stop-word removal on each single word in the dataset. We use NLTK toolkit [4] for Tokenization, lemmatisation and stop-word removal.
4) Words with frequency lower than two are removed, and tweets whose length than two words are excluded.The resulting dataset is then ready for feature transformation.

Text representation is to transform raw tweets into numeric vectors that is suitable for machine learning algorithms. Specifically, we compare the traditional context-free BoW representation with contextualized pre-trained language models. For BoW, we include term frequency based methods and TF-IDF based methods. For BERT, We include Sentence-BERT and CyBERT. Each representation method is detailed as follows:

- ***BoW*** creates a vocabulary with all unique word in the corpus[5] and represents each document, in our case, tweet, as a vector of word occurrence. We implement the Term Frequency method with Gensim [6].
- **TF-IDF** is a variant of BoW that computes the product of term frequency (TF) and inversed-document-frequency (IDF) for each word in the document, implemented with Gensim TF-IDF transformer.
- **SentenceBERT(SBERT)** is a framework to embed a sentence or a document into contextualized dense vectors [38]. SBERT is fine-tuned on pre-trained language models on the similarity matching training objective so that SBERT excels at clustering, semantic matching, and other unsupervised tasks [38].
- **CyBERT** is a Contextualized language model for the Cybersecurity Domain introduced by [39]. CyBERT is based on BERT architecture and fine-tuned on a large cybersecurity corpus consisting vulnerability reports and CVE databases. We include CyBERT to investigate whether a domain-specific representation outperforms the domain independent SBERT on the topic modeling task. We keep CyBERT's initial weights and extract features from the models' last four hidden layers. To remain consistent with SBERT, we use the pooling strategy to extract the document level vector [38].

[4]https://www.nltk.org/
[5]In NLP, corpus refers to a collection of texts
[6]https://radimrehurek.com/gensim/

C. Topic Modeling

Topic modelling is an unsupervised approach for discovering latent topics in a collection of documents. The goal of topic modelling is to model the relationships between three concepts: document (in this case, tweets), word, and topic.

Preliminaries. Formally, let $D = \{d_1, d_2, ..., d_j\}$ denotes a collection of documents with $W = \{w_1, w_2, ..., w_i\}$ unique words. Let $Z = \{z_1, z_2, ..., z_K\}$ denotes number of K latent topics discovered by a method, where K can be user-defined or automatically set by a hyper-parameter searching method. A topic model decomposes the $Documents \times Words$ (DW) matrix into a $Documents \times Topics$ (DZ) matrix, and a $Topics \times Words$ (ZW) matrix, as illustrated in Figure 2. Each matrix is in $(Key, Value)$ format. The DW matrix is commonly found by a BoW method [10]. The ZW matrix is used to gain the topic representation, with each topic represented by the $top - f$ most probable words. The DW matrix is used to gain the topic distribution over documents.

Dynamic topic representation. Considering documents are collected during $T = \{t_1, t_2, ..., t_m\}$ days and each document is attached to a time step $t_d \in \{1, 2, \ldots, m\}$. Specifically, the topic weights are calculated using the mean of the topics-document distribution for all documents at t_m time steps. By arranging documents based on the time sequence, the topic model generates dynamic representations that allow topics vary smoothly over time.

1) Topic model Methods

There exist several topic modeling methods and no single approach works for all data. Methods differ depending on how data is represented, how topics are identified, and how the model is evaluated. In this framework, we provide a means to adopt different family of topic models and evaluate their performance on the CTI data to choose the best fitting method. The framework includes traditional topic models like LDA and NMF, as well as a neural topic model method named CTM. [36]. We evaluate and compare them based on topic coherence, topic divergence, topic coverage, and runtime. This

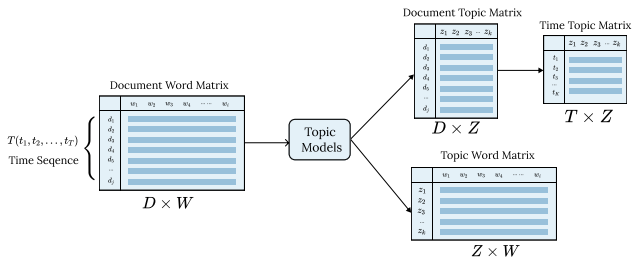


Fig. 2: Topic modeling concepts

enables cybersecurity professionals to choose their prefered topic model based on their use cases.

Non-negative Matrix Factorization(NMF) is a linear algebraic-based method for obtaining a low rank representation (with non-negative values) from the high-dimensional input data $W \times D$. NMF aims to find the optimal multiplication of $W \times Z$ and $Z \times D$ that approximately reconstruct $W \times D$ through a loss function. Matrix multiplication can be implemented as computing the column vectors of $W \times Z$ as linear combinations of the column vectors in $Z \times D$ using coefficients supplied by columns of $W \times D$. We include online-NMF [41] in the framework as it is known to perform well for short, sparse social media data. We derived the TF-IDF representation for computing $W \times D$ following the common practice.

Latent Dirichlet Allocation(LDA) is a generative topic model where each document is assumed to be represented as random mixtures over latent topics $P(T|d)$, and each topic is represented as a multinomial probability distribution over the words $P(z|W)$ [42]. Each topic distribution contains every word in the corpus, but their probability varies. The $top-k$ most likely words are used to represent each topic. k is usually set to 10 or 15. We include the LDA multi-core implementation from Gensim [43] in the framework because it is one of the most widely used topic model. BoW with term frequency is used as the text representation for LDA [44].

Contextualized topic model (CTM) is a neural topic modeling method based on the Variational Auto-Encoders (VAE) architecture [45].A VAE architecture contains an encoder and a decoder. The encoder directly approximates the topic distribution denotes by μ for each document $P(Z|d)$. The decoder then estimates the word distribution denotes by σ^2 for each topic $P(w|z)$ by reconstructing a document's BoW data from its topic distribution. These probabilities are typically parameterised by deep neural networks [10]. A continuous topic representation z is then sampled from a pair of μ and σ^2 that is regularized by a Gaussian distribution (known as the prior) $\mathcal{N}(\mu, \sigma^2)$.

Particularly, CTM [36] is a special form of NTM that takes both BoW and contextualized representation(e.g., SBERT) as input. It uses the contextualized document representation to reconstruct the symbolic BoW representation. This flexible setting enables CTM to incorporate external knowleadge from pretrained language models to produce more coherent and diverse topics. We follow the implementation from [36]. CTM is trained with 20 epochs to prevent over-fitting.

2) Evaluations

The performance of various topics models are evaluated from topic coherence, topic diversity, topic coverage and runtime four aspects.

Topic Coherence measures the ability of topics being coherent and consistent for human's interpretation. Röder et al. [46] proposed a systematic framework to automatically evaluate topic coherence. Specifically, we choose C_V, which was found best reflects the topic correlation through human judgement [47] and C_{W2V}, which uses a Word2vec model to indirectly estimate the cosine similarity [48] of the $top-f$ words in a topic. We use a Word2vec model that is trained on the cybersecurity domain [49] to calculateC_{W2V}.The values of C_V and C_{W2V} range from [0,1]. A higher score indicates more human interpretative topics.

Topic Diversity measures how diverse a topic are by calculating the percentage of unique words in the top 25 topic words for all topics [50]. Topic diversity ranges from [0,1]. A higher score indicates more diverse and possibly more informative topics, whereas a low score may indicate topics that are redundant.

Topic Coverage is proposed to estimate the relevance of produced topics. In our particular use case, ideally, we would like the topic model to produce more cybersecurity related topics. As Twitter users commonly use hashtags to annotate tweets. We define the topic coverage by the percentage of overlapped words between all topic words and the $top-50$ most frequent hashtags in the tweets. Let A denotes the top 50 most frequent hashtags extracted from the dataset. Let B be a list of topic words generated by the topic model. We define topic coverage as:

$$T_{coverage} = \frac{|A \cap B|}{A}$$

Runtime. We report the wall time measured in seconds for each model excluding the embedding operations. LDA and NMF are computed on a Intel(R) Xeon(R) CPU @ 2.20GHz instance, and CTM is computed on a Tesla T4 instance. Both of the resources are freely available on Google Colab [7].

D. Topic Analysis

This phase presents the topic modeling results with various visualizations to assist the cybersecurity professionals in understanding topic dynamics and insights. Specifically, the Word Cloud, Heat Map and line charts are used to present the results. We will demonstrate the usage through the empirical analysis.

IV. Empirical Analysis

To demonstrate the feasibility of the proposed vulnerability discovery and monitoring framework, we conduct a case study to analyze the critical log4shell zero-day vulnerability incident from Twitter discussions. To gain a quantitative understanding of the log4shell incident, we first map the log4shell-related time points in accordance with the vulnerability lifecycle to provide some background information [16] [17], followed by the description of the log4shell dataset. We then present results and analysis from the proposed vulnerability discovery and monitoring framework.

Prior empirical analysis [2] [19] reveals that vulnerabilities are likely to be discussed on Twitter prior to their public disclosure, providing a window of "attainable time advantage" for early detection and detection of cyber threats and attacks. Given the severity and enormous attentions received by the log4shell incident, we aim to use the proposed framework to uncover following questions:

1) Whether the proposed framework discover log4shell zero-day related information

[7]https://colab.research.google.com/

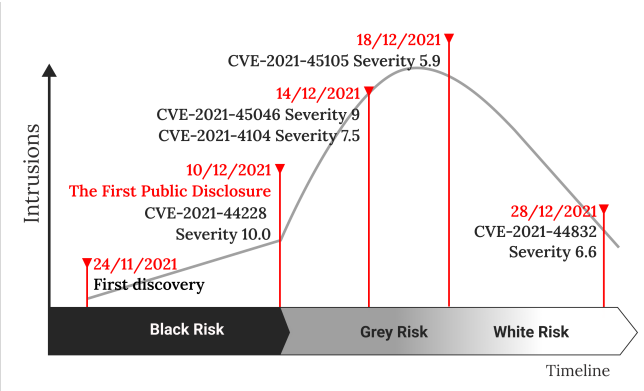
2) Which topic modeling method performs best on a CTI Twitter dataset?
3) How log4shell related information is discussed on Twitter during different stages of vulnerability lifecycle?
4) Whether log4shell zero-day are discussed on Twitter before the NVD public disclosure.

A. The vulnerability lifecycle and log4shell incident

The vulnerability lifecycle introduced by [2] [18] [16] divide the lifecycle of a vulnerability into black risk, grey risk and white risk three phases. It provides a dynamic and quantitative understanding of the log4shell at different risk exposure phases.

Log4j is an open-source Java library that is part of the Apache Logging Services. The log4shell vulnerabilities discovered in the log4j open-source library fall under the category of Remote Code Execution (RCE) [51]. log4shell enables attackers to remotely execute arbitrary code and potentially gain complete control of the system. As Logging is a basic function widely used in many Java applications, the log4shell incident impacted many large software companies and online services (e.g., Amazon, Apple iCloud, Cisco, Cloudflare, ElasticSearch, Red Hat, Steam, Tesla, Twitter), causing tremendous financial loss [51] [52] [53].

- ***Black Risk*** is the period between a vulnerability's first discovery and public disclosure. Typically, the time of discovery is the earliest time a software vulnerability is identified as posing a security risk [18]. During this period, only a small group of people are aware of the vulnerability and the information is only shared in a closed group. The log4shell vulnerability was reported to be found on 24th November by the Alibaba Security Group. Usually, a vulnerability is not publicly known until its disclosure by an authorized body such as the NVD. The log4shell was publicly announced at December 10, 2021 by NVD (as CVE-2021-44228). There is evident that log4shell was exploited by hackers prior to the public disclosure that make log4shell a typical form of zero-day vulnerability that received a severity score 10 out of 10 [51].



[18] [16]

Fig. 3: Mapping the log4shell vulnerability incident to the vulnerability life cycle model

- The ***Gray Risk*** period begins from the public disclosure and ends with the release of countermeasures from authorized vendors. The public disclosure is defined as the time when vulnerability information is freely available to the public and published by a trusted body (e.g., NVD) [18]. By the time of public disclosure, the vulnerability had been examined by secueiry experts and rik assessments were included (e.g., The CVSS vulnerability metrics by NVD) [2]. Once the vulnerability information became public, it began to draw increased attention, and the corresponding intrusions skyrocketed [16]. In the case of log4shell, it was reported that nearly 10 million exploitation attempts per hour were identified after the public disclosure at 10 December 2021 [53].Given the severity and magnitude of the log4shell incident, we anticipate a broader social impact and corresponding Twitter discussions.
- ***White Risk*** refers to the period between patch availability and patch implementation. Vulnerabilities, once discovered, are often fixed by implementing patches(e.g., software updates) by end users [2]. Since this period highly depends on the end users, we do not differentiate the time point between grey risk and white risk in Figure 3.

B. Dataset.

Following the methods discussed in the framework, we tracked 47 SIPs in the CTI community and collected tweets between November 23rd and December 29th, 2021, which covered the lifecycle of the log4shell incident. Each tweet has a distinct ID, user, text, and timestamp. To analyse the dynamic topic patterns and trends, we ordered the tweets chronologically and used a single date as a time step. A glance of the dataset is included in TableI. The final log4shell dataset includes 24407 raw tweets from 47 SIPs over the duration of 37 days. This dataset is available online at [8].

C. Topic Modeling and Analysis

1) Quantitative analysis

We run NMF, LDA, and CTM with K set to 10-50 with a step of 10 on four evaluation metrics to gain a comparative performance on each model. We present the top-10 most probable words in each topic as the topic representation for assessing the topic quality.

With the quantitative analysis, we aim to answer: (1) Which topic modeling method performs best on the given log4shell dataset? (2) How the hyper-parameter K influence the models performance. Results are reported in FigureII.Once fitted, we select the most suitable model for topic analysis.

Topic coherence. C_V is the intrinsic measurement of the topic words being coherence and similar in the vector space. The results show that CTM with two contextualized representations (SentenceBERT and CyBERT) consistently outperform NMF and LDA with BoW based representations. This is consistent with the experimental results from [36] that CTM

[8]https://github.com/joywang233/log4shell_dataset

is able to produce more coherent topics. We infer that SBERT and CyBERT introduce more contextualized information into the topic representation, making the top-10 topic words more semantically co-related. When looking into C_{w2v}, all C_{w2v} scores are higher than the corresonding C_v but there is no significant differences between each models at different K values. As C_{w2v} is calculated based on a Word2vec model trained on cybersecurity corpus, high C_{w2v} scores indicate a strong correlation between topic words and the cybersecurity domain. However, because C_{w2v} is calculated on each fine-grained word vector, some of them were not present in the Word2vec's vocabulary, resulting in an out-of-the-vocabulary (OOV) issue, making the resulting scores less distinguishable.

Topic diversity. In terms of topic diversity, all models' diversity decreases as the number of K increases,implying that the optimal K for the given dataset is less than 20. When K is set to 10 and 20 separately, CTM finds the most diverse topic that significantly outperforms NMF and LDA. However, as K increases, the topic diversity of CTM dramatically decreases, indicating redundant topic words are generated. By contracst, the diversity of NMF and LDA appear to be more resilient with different K values. This could indicate that CTM is overfitting or has some degree of "posterior collapse", which is a common problem with VAE model [10] [45]. As a result, the learned parameters for the posterior distribution become uni-formative, and so does the sampled topic words. Similar trend is found for the topic coverage measurements. When K is set to 10, all topic models find the best topic coverage, this some how reflect that $K = 10$ is the optimal hyper-parameter among all K values.

Runtime. In terms of runtime, as the K increases, all models experience an increase in runtime. NMF shows supreme computing efficiency on the given task, whereas CTM is more resource intensive considering the use of GPU computing. Aside from that, CTMs, or neural topic models, may show benefits on large datasets because they use gradient-based approaches for optimization [54].

2) Qualitative analysis

To further examine the topic quality, we present the topics from each model when $K = 10$ in Table III. As C_V best reflects the topic coherence, we marked topics with highestC_V in bold, and topics with lowest C_V are underlined. We can observe that all topic models found log4shell related topics with the highest C_V scores, indicating that log4shell has recived great attentions and viral discussion on Twitter. Topic words such as"vulnerability","exploit","rce", "apache" and "cve202144228" are coherent and informative given the log4shell background information.

Further observations confirmed that CTM topics are more well-organized and semantically related than those discovered by NMF and LDA. This finding is consistent with the topic diversity measurement, which shows that when $K = 10$, CTM produces more diverse and informative topics. For example, in NMF-$t5$, the word "log4shell" appears under the same topic as "love", "sure" and "thing", which do not appear to be related. In LDA, a similar trend is observed, with the word "log4j" spreading across 6 other topics. In a nutshell, CTM with contextualised representations produces more coherent, diverse, and informative topics than other methods.

3) Dynamic topic pattern analysis

This section presents dynamic topic analysis and insights. We used the methods discussed in SectionIII-D to generate the visualizations from the CTM model with the CyBERT embedding. Through the analysis we aim to find out:

- How log4shell related information is discussed on Twitter during different stages of vulnerability lifecycle?
- Whether log4shell zero-day are discussed on Twitter before the NVD public disclosure.

Overview. Figure 4 depicts a heat-map of ten topics and their dynamic representations over time. The X-axis displays the dates in chronological order, and the Y-axis displays the Topic weights. Each square represents the topic weights on a particular day. A higher value was indicated by a lighter colour. The $top - 3$ words from each topic are extracted as Y-labels.

From the heat map, we can easily examine the topic subject. CTM, in particular, discovered eight out of ten well-defined topics: The second topic is about Linux users. Topics 3 and 9 are about log4shell vulnerabilities; Topic 4 is about Covid-19; Topic 6 is about ransomware; and Topic 7 is about Christmas. Topic 8 is a little ambiguous because of the word "wehackhealth". According to further investigation, "wehackhealth" is a fitness-related promotional campaign that was popular in December 2021. Finally, Topic 10 goes over general cybersecurity.

Log4j related topics. To gain a closer look of how log4shell related topics change overtime. We present some of the experiment results in Figure 5. Figure 5a shows dynamic patterns for log4shell related topic. In general, Topic 3 and Topic 9 shared a similar trend. Starting on December 9th, there is a significant increase in Topic 9's weights, and it peaks perfectly on December 10th, which is known as log4shell's public disclosure. This trend can be explained by the Vulnerability lifecycle model, which states that when a critical vulnerability is publicly disclosed, it garners significant attention, and the associated intrusion attacks skyrocket. A closer examination of

ID	Date	Text	User
[ID]	2021-12-10	This is karmic payback for everyone dogpiling on the Raspberry Pi thing	[User]
[ID]	2021-12-10	The events of the day in a nutshell log4shell	[User]
[ID]	2021-12-10	Me Huh vacation is nice but maybe I should check in on things	[User]
[ID]	2021-12-10	Do we have a CVE for kalikali yet or should AHax apply for one Asking for a friend AHax	[User]
[ID]	2021-12-10	Fxxk money Were gonna start posting IOCsraw data for hosts exploiting Apache Log4J CVE202144228 as often as we ca…	[User]

TABLE I: Example tweets from the log4shell dataset. [ID] and [user] is used to anonymise the tweet identifier and the actual user ID, the date is converted in a YYYY-MM-DD format for analysing the daily dynamic patterns and trends.

		C_V ↑(Mean±Std.)	$C_{Word2Vec}$ ↑(Mean±Std.)	Diversity↑	Topic Coverage ↑	Runtime (Secs)↓
K=10	NMF	0.37±0.1	0.72±0.07	0.86	0.11	**8.9**
	LDA	0.3±0.05	**0.73±0.04**	0.58	0.12	88.6
	CTM	**0.46±0.14**	0.67±0.04	**1.00**	**0.14**	90.7
	CyBERT_CTM	0.44±0.12	0.68±0.08	0.94	0.13	107.9
K=20	NMF	0.35±0.15	0.7±0.05	0.77	0.09	**7.6**
	LDA	0.32±0.09	**0.71±0.05**	0.58	0.11	86
	CTM	0.36±0.13	0.64±0.07	0.84	0.06	87.3
	CyBERT_CTM	**0.44±0.12**	0.68±0.08	**0.94**	**0.13**	107.9
K=30	NMF	0.37±0.14	0.67±0.04	**0.73**	0.08	**10.2**
	LDA	0.32±0.11	**0.69±0.06**	0.60	**0.09**	85.2
	CTM	0.37±0.11	0.63±0.07	0.67	0.05	88.8
	CyBERT_CTM	**0.41±0.18**	0.63±0.08	0.51	0.05	103.3
K=40	NMF	0.35±0.12	0.67±0.04	**0.70**	0.05	**20.5**
	LDA	0.33±0.09	**0.69±0.05**	0.60	**0.07**	91.7
	CTM	0.38±0.12	0.63±0.07	0.55	0.03	88.1
	CyBERT_CTM	**0.42±0.14**	0.63±0.07	0.45	0.04	101.7
K=50	NMF	0.34±0.11	**0.67±0.05**	**0.67**	**0.06**	**16.2**
	LDA	0.32±0.07	0.68±0.06	0.66	**0.06**	94.5
	CTM	0.39±0.12	0.64±0.07	0.49	0.03	87.2
	CyBERT_CTM	**0.42±0.14**	0.63±0.07	0.45	0.04	101.7

TABLE II: Experimental results for all topic models with K set to 10-50 with a step of 10. Final scores are computed based on the top-10 most probable words in each topic. We report the mean and standard deviation for topic coherence C_V and C_{W2V}

Model	Topics
NMF	z_1: **log4j,vulnerability,log4shell,exploit,find,patch,rce,vulnerable,code,cve202144228**
	z_2: work, day, nice, security, great, new, long, mate, version, year
	z_3: time, thank, need, today, have, day, wehackhealth, get, great, share
	z_4: not, think, look, will, sure, know, tell, log, oh, wait
	z_5: thing, sure, threat, ransomware, get, love, log4shell, actor, mean, cyber
	z_6: know, new, want, use, cybersecurity, ransomware, group, attack, come, linux
	z_7: good, think, luck, hope, point, thread, old, person, bad, security
	z_8: go, to, look, right, way, fun, cyber, get, weekend, lot
	z_9: yeah, try, like, run, oh, dude, bad, list, issue, stop
	z_{10}: people, like, think, see, lot, feel, want, sta, have, happen
LDA	z_1: get, year, new, security, time, have, say, holiday, go, people
	z_2: work, log4j, happy, time, different, dude, issue, file, see, xmas
	z_3: **log4shell, log4j, ransomware, good, attack, not, thing, exploit, vulnerability, update**
	z_4: linux, log4j, bad, exploit, know, want, look, use, need, vulnerability
	z_5: like, get, cybersecurity, yeah, think, thank, tweet, week, oh, read
	z_6: new, look, like, christmas, security, have, see, learn, good, friend
	z_7: not, know, cyber, good, people, go, thing, stuff', hope, new
	z_8: not, log4j, go, new, think, right, use, work, version, fix
	z_9: day, like, time, need, log4j, thing, sta, go, look, have
	z_{10}: log4j, not, vulnerability, like, think, log, log4shell, yeah, time, come
SBERT_CTM	z_1: year, join, today, holiday, free, infosec, company, week, cyber, 2022
	z_2: linux, use, log, run, file, app, command, window, java, add
	z_3: youth, nuanced, detailsofthecase, unidentified, fillin, whic, devlife, empower, locallhost, cincinnati
	z_4: youth, grinoch, wehackhealth, thank, morning, fastest, locallhost, man, violence, fillin
	z_5: **0day, cve202144228, exploit, vulnerability, log4j, apache, exploitation, rce, detect, release**
	z_6: ransomware, new, researcher, attack, group, target, hacker, malware, threat, publish
	z_7: de, c2, el, la, en, un, los, qakbot, pi, es
	z_8: threat, see, blog, security, cloud, amp, incident, vendor, webinar, product
	z_9: lot, people, time, go, work, have, think, not, pretty, get
	z_{10}: yeah, feel, like, mean, hard, guess, think, not, ill, sure
CyBERT_CTM	z_1: yeah, panel, creative, feel, hard, guess, maybe, toon, caption, like
	z_2: linux, file, unix, user, password, window, command, shell, download, script
	z_3: log4shell, detection, attacker, network, block, server, rule, tool, test, find
	z_4: omicron, people, time, covid, game, vaccine, person, case, hour, have
	z_5: twitter, customer, work, source, tweet, product, open, trust, vendor, public
	z_6: ransomware, malware, target, group, researcher, attack, discover, campaign, botnet, hacker
	z_7: christmas, mate, happy, merry, sleep, soon, thank, santa, evening, awesome
	z_8: squat, wehackhealth, weight, range, lift, belt, techdi, igual, protein, authorisation
	z_9: **log4j, vulnerability, exploit, java, update, cve202144228, 0day, log4shell, vulnerable, apache**
	z_{10}: cybersecurity, cyber, join, security, team, 2022, live, 2021, threat, black

TABLE III: Output topics from all TMs when $K = 10$. Topics with highest C_v are marked in bold. Topics with lowest C_v are underlined.

Figure 5a reveals that CTM discovers log4shell-related topics appear 1-2 days before the public disclosure date. This could imply that SIPs shared early indicators and pre-exploitable patterns regarding zero-day exploits on Twitter, which could be associated with log4shell vulnerabilities, though the official CVE of log4shell is not available at that time. Such early indicator and pre-exploitable patterns are indicated by the terms"exploit", "0day", "RCE" found by topic models. This could also be due to time zone differences between different regions. The early morning of December 10th, for example, is still December 9th in America. Moreover, there are significant fluctuations between December 12th and December 17th. We believe this is because another two critical log4shell-related vulnerabilities (CVE-2021-4104 [9] and CVE-2021-45046 [10]) are announced on December 14th, which continues to drive a high volume of discussion among the SIPs community.

Linux and ransomware topics. Figure 5b displays the dynamic topic patterns of Topic 2, Topic 6 and Topic 10. During the 26th and 9th of December, all three topics followed a very similar trend, and the topic weights remain in relatively high value. Considering the associations between the words: "Linux", "user", "malware", "attack", "discover", we infer there is a strong link between Linux systems and ransomware (or malware) activities. This is supprted by the 2022 IBM Security report [55] that malware targeting Linux environments dramatically increased during 2021. Furthermore, they claim that the monthly number of ransomware incidents was high in December, ranking third in 2021.

Non-cybersecurity topics. We compare two non-cybersecurity topics and their dynamic patterns in Figure 5c and Figure 5d. We have noticed that SIP discussions are not limited to cybersecurity-related topics. They also discuss seasonal (for example, Christmas) and general trending topics (e.g. Covid). People are worried about Omicron and Vaccination in November and December of 2021, they also exchange greetings and wished during Christmas. A closer look into Topic 7 shows, the topic weights suddenly started

[9]https://nvd.nist.gov/vuln/detail/CVE-2021-4104
[10]https://nvd.nist.gov/vuln/detail/CVE-2021-45046

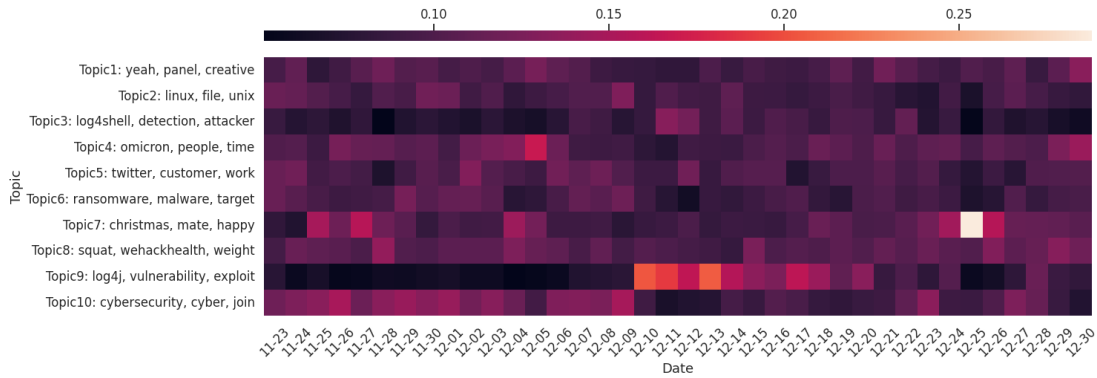


Fig. 4: Heat map generated from CTM

to sear at 23rd Dec and sharply decay at 26th Dec. This trend is in line with people's general behaviors during Christmas. Different from the log4shell topic, such seasonal topics are likely to happen recurrently. Finally, Figure5d showing the dynamic trend of Topic 8 which is related to a fitness campagin named "wehackhealth"

D. Discussions

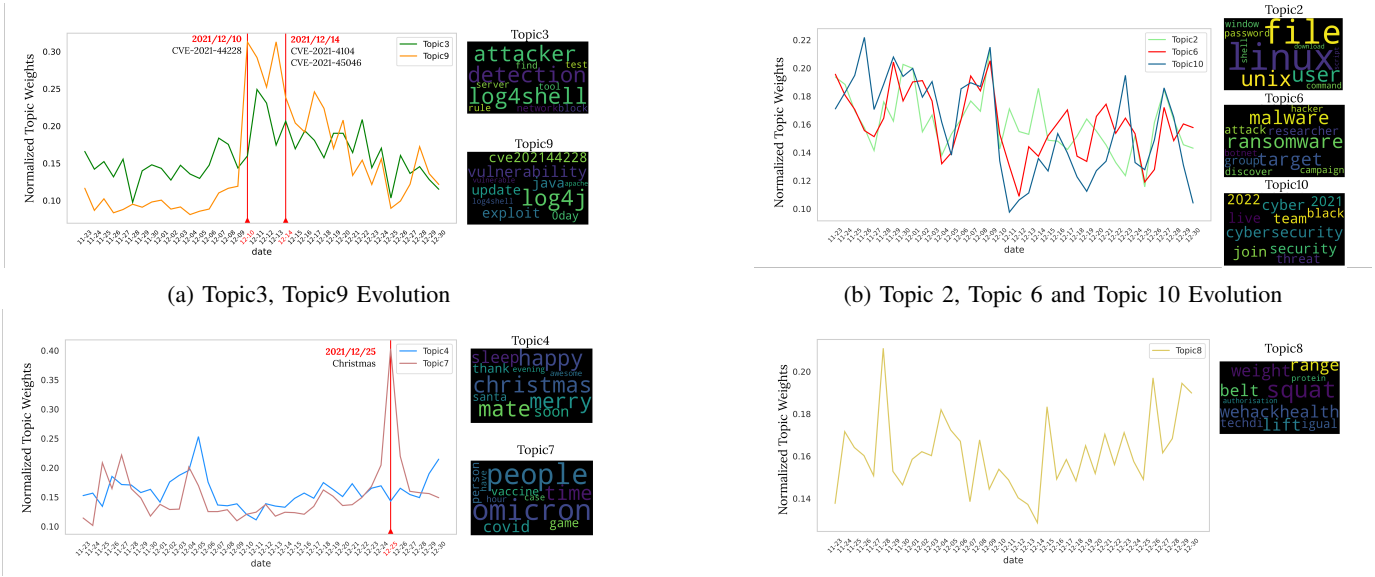
To summarise, we discovered that the coherence score C_V best reflects topic coherence in the given dataset, which is consistent with human judgement. The number of topics K is a deterministic factor that affects both topic diversity and runtime. CTM with contextualized document representation improve the produced topic coherence and diversity which is in line with [36]'s finding. Furthermore, CTM's ability to separate the embedding and topic modelling processes is an appealing feature that allows users to use different pre-trained language models that are more appropriate for the training dataset.

We also discovered that the STP community discusses a wide range of cybersecurity and non-cybersecurity topics, and that they respond actively to critical cyber attacks and vulnerability incidents such as the log4shell zero-day.log4shell, in particular, is a critical vulnerabilities incident that was ranked as the second most critical exploit in 2021 [55]. Because of its high impact factor, log4shell received a lot of attention and a lot of discussion on Twitter in the SIP community. Through the dynamic topic analysis,we can identify a set of interesting patterns that are in line with the real-word events.These findings are crucial to understanding the risk exposure of cyber events at various stages, as well as people's perceptions and behaviours.

V. Conclusion and Future Work

In this study, we proposed a vulnerability discovery and monitoring framework to help cybersecurity professionals collect, process, and organise unstructured tweets into interpretable topics with dynamic patterns and trends. The proposed framework compares various topic modelling techniques (NMF, LDA, and CTM) on a slef-collected dataset to understand the log4shell vulnerabilities incident. Experiments reveal that CTM with BERT-based text representation produces more coherent and diverse topics that are suitable for unstructured, short, domain-dependent CTI text in Twitter. We found that CTM discovered dynamic topic patterns and trends that correspond to real-world events. log4shell-related topics, in particular, appeared on Twitter a few days before the NVD public disclosure, demonstrating that Twitter is a valuable data source for early detection of cyberthreats. When compared to traditional forensic analysis, which is primarily based on human investigation, the proposed framework significantly improves efficiency when dealing with large volumes of documents. In practise, this framework could be used to monitor Twitter conversations or to crowdsource information from Twitter as part of the digital forensic process.

This study does not come without limitations. While the proposed system have potential to discover critical zero-day vulnerability events on Twitter, it deploys a human-in-the-loop design that allows cybersecurity professionals to filter and investigate the selected topics based on their knowledge and expertise. As the volume of data grows, such human intervention may become impractical. In the future, we plan to propose a fully automatic system for discovering and detecting cybersecurity-related topics to replace human intervention. The neural topic model has the promising advantage of separating the embedding and modelling processes, allowing for greater flexibility in using cutting-edge pre-trained language models such as BERT. We investigated the SBERT



(a) Topic3, Topic9 Evolution

(b) Topic 2, Topic 6 and Topic 10 Evolution

(c) Topic 4 and Topic 7 Evolution

(d) Topic 8 Evolution

Fig. 5: Topic Evolution Over Time

and CyBERT pre-trained language models in our research. A thorough examination of how different language models influence topic modelling performance could be one of the future research directions.

REFERENCES

[1] S. Frei, D. Schatzmann, B. Plattner, and B. Trammell, "Modeling the security ecosystem - the dynamics of (in)security," in *Economics of Information Security and Privacy*, T. Moore, D. Pym, and C. Ioannidis, Eds. Boston, MA: Springer US, 2010, pp. 79–106.

[2] C. Sauerwein, C. Sillaber, M. M. Huber, A. Mussmann, and R. Breu, "The tweet advantage: An empirical analysis of 0-day vulnerability information shared on twitter," in *ICT Systems Security and Privacy Protection*, L. J. Janczewski and M. Kutyłowski, Eds. Cham: Springer International Publishing, 2018, pp. 201–215.

[3] M. Almukaynizi, E. Nunes, K. Dharaiya, M. Senguttuvan, J. Shakarian, and P. Shakarian, "Proactive identification of exploits in the wild through vulnerability mentions online," in *2017 International Conference on Cyber Conflict (CyCon US)*. IEEE, 2017, pp. 82–88.

[4] H. Yang, S. Park, K. Yim, and M. Lee, "Better not to use vulnerability's reference for exploitability prediction," *Applied Sciences (Switzerland)*, vol. 10, 4 2020.

[5] N. Tavabi, P. Goyal, M. Almukaynizi, P. Shakarian, and K. Lerman, "Darkembed: Exploit prediction with neural language models," 2018. [Online]. Available: https://helpx.adobe.com/security/severity-ratings.html

[6] X. Liu, J. Fu, and Y. Chen, "Event evolution model for cybersecurity event mining in tweet streams," *Information Sciences*, vol. 524, pp. 254–276, 7 2020.

[7] V. Behzadan, C. Aguirre, A. Bose, and W. Hsu, "Corpus and deep learning classifier for collection of cyber threat indicators in twitter stream." Institute of Electrical and Electronics Engineers Inc., 1 2018, pp. 5002–5007.

[8] Z. Long, L. Tan, S. Zhou, and X. Liu, "Collecting indicators of compromise from unstructured text of cybersecurity articles using neural-based sequence labelling," 2019. [Online]. Available: http://www.ieee.org/publications

[9] A. D. Householder, J. Chrabaszcz, T. Novelly, D. Warren, and J. M. Spring, "Historical analysis of exploit availability timelines," in *13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20)*. USENIX Association, Aug. 2020. [Online]. Available: https://www.usenix.org/conference/cset20/presentation/householder

[10] H. Zhao, D. Phung, V. Huynh, Y. Jin, L. Du, and W. Buntine, "Topic modelling meets deep neural networks: A survey," *arXiv preprint arXiv:2103.00498*, 2021.

[11] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv*, 2018. [Online]. Available: https://github.com/tensorflow/tensor2tensor

[12] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Computational Intelligence*, vol. 31, no. 1, pp. 132–164, 2015.

[13] H. S. Dutta and T. Chakraborty, "Blackmarket-driven collusion among retweeters-analysis, detection, and characterization," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1935–1944, 2020.

[14] Y. Zhang, X. Ruan, H. Wang, H. Wang, and S. He, "Twitter trends manipulation: A first look inside the security of twitter trending," *IEEE Transactions on Information Forensics and Security*, vol. 12, pp. 144–156, 1 2017.

[15] H. Jo, J. Kim, P. Porras, V. Yegneswaran, and S. Shin, "Gapfinder: Finding inconsistency of security information from unstructured text," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 86–99, 2021.

[16] W. A. Arbaugh, W. L. Fithen, and J. McHugh, "Windows of vulnerability: A case study analysis," *Computer*, vol. 33, no. 12, pp. 52–59, 2000.

[17] S. Frei, M. May, U. Fiedler, and B. Plattner, "Large-scale vulnerability analysis," in *Proceedings of the 2006 SIGCOMM Workshop on Large-Scale Attack Defense*, ser. LSAD '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 131–138. [Online]. Available: https://doi.org/10.1145/1162666.1162671

[18] S. Frei, B. Tellenbach, and B. Plattner, "0-day patch exposing vendors (in)security performance," 2008. [Online]. Available: http://www.techzoom.net/risk/

[19] P. Shrestha, A. Sathanur, S. Maharjan, E. Saldanha, D. Arendt, and S. Volkova, "Multiple social platforms reveal actionable signals for software vulnerability awareness: A study of github, twitter and reddit," *PLoS ONE*, vol. 15, 2020.

[20] F. Alves, A. Bettini, P. M. Ferreira, and A. Bessani, "Processing tweets for cybersecurity threat awareness," *Information Systems*, vol. 95, 1 2021.

[21] J. Li, X. Hu, J. Tang, and H. Liu, "Unsupervised streaming feature selection in social media," vol. 19-23-Oct-2015. Association for Computing Machinery, 10 2015, pp. 1041–1050.

[22] E. Sobiesk, D. Bennett, P. Maxwell, W. P. A. C. Institute, N. C. C. D. C. of Excellence, I. C. Society, I. of Electrical, and E. Engineers, "Proactive identification of exploits in the wildthrough vulnerability mentions online," 2017.

[23] H. Chen, R. Liu, N. Park, and V. S. Subrahmanian, "Using twitter to predict when vulnerabilities will be exploited." Association for Computing Machinery, 7 2019, pp. 3143–3152.

[24] N. Dionísio, F. Alves, P. M. Ferreira, and A. Bessani, *Cyberthreat Detection from Twitter using Deep Neural Networks*, 2019. [Online]. Available: http://www.ieee.org/publications

[25] T. Sutanto and R. Nayak, "Fine-grained document clustering via ranking and its application to social media analytics," *Social Network Analysis and Mining*, vol. 8, no. 1, pp. 1–19, 2018.

[26] W. A. Mohotti and R. Nayak, "An efficient ranking-centered density-based document clustering method," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 439–451.

[27] L. F. D. C. Nassif and E. R. Hruschka, "Document clustering for forensic analysis: An approach for improving computer inspection," *IEEE Transactions on Information Forensics and Security*, vol. 8, pp. 46–54, 2013.

[28] A. Bose, V. Behzadan, C. Aguirre, and W. H. Hsu, "A novel approach for detection and ranking of trendy and emerging cyber threat events in twitter streams." Association for Computing Machinery, Inc, 8 2019, pp. 871–878.

[29] S.-Y. Huang and T. Ban, "Monitoring social media for vulnerability-threat prediction and topic analysis," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2020, pp. 1771–1776.

[30] Y. Fang, Y. Liu, C. Huang, and L. Liu, "Fastembed: Predicting vulnerability exploitation possibility based on ensemble machine learning algorithm," *PLoS ONE*, vol. 15, 2 2020.

[31] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, and Y. Xiang, "Data-driven cybersecurity incident prediction: A survey," *IEEE Communications Surveys and Tutorials*, vol. 21, pp. 1744–1772, 4 2019.

[32] Z. Zhu and T. Dumitras, "Chainsmith: Automatically learning the semantics of malicious campaigns by mining threat intelligence reports." Institute of Electrical and Electronics Engineers Inc., 7 2018, pp. 458–472.

[33] Z. Liu, Y. Lin, and M. Sun, *Representation Learning and NLP*. Springer, Singapore, 2020. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-15-5573-2_1

[34] D. Angelov, "Top2vec: Distributed representations of topics," 2020.

[35] S. Stieglitz, M. Mirbabaie, B. Ross, and C. Neuberger, "Social media analytics – challenges in topic discovery, data collection, and data preparation," *International Journal of Information Management*, vol. 39, pp. 156–168, 4 2018.

[36] F. Bianchi, S. Terragni, and D. Hovy, "Pre-training is a hot topic: Contextualized document embeddings improve topic coherence," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 759–766. [Online]. Available: https://aclanthology.org/2021.acl-short.96

[37] F. Bianchi, S. Terragni, D. Hovy, D. Nozza, and E. Fersini, "Cross-lingual contextualized topic models with zero-shot learning," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 1676–1683. [Online]. Available: https://www.aclweb.org/anthology/2021.eacl-main.143

[38] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[39] P. Ranade, A. Piplai, A. Joshi, and T. Finin, "Cybert: Contextualized embeddings for the cybersecurity domain," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 3334–3342.
[40] W. A. Mohotti and R. Nayak, "Discovering communities with sgns modelling-based network connections and text communications clustering," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 1770–1777.
[41] R. Zhao and V. Y. Tan, "Online nonnegative matrix factorization with outliers," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 555–570, 2016.
[42] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.
[43] M. Hoffman, F. Bach, and D. Blei, "Online learning for latent dirichlet allocation," *advances in neural information processing systems*, vol. 23, 2010.
[44] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.
[45] A. Srivastava and C. Sutton, "Autoencoding variational inference for topic models," *arXiv preprint arXiv:1703.01488*, 2017.
[46] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures." Association for Computing Machinery, 2 2015, pp. 399–408.
[47] S. Syed and M. Spruit, in *2017 IEEE International conference on data science and advanced analytics (DSAA)*. IEEE, 2017, pp. 165–174.
[48]
[49] S. Mumtaz, C. Rodriguez, B. Benatallah, M. Al-Banna, and S. Zamanirad, "Learning word representation for the cyber security vulnerability domain," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.
[50] A. B. Dieng, F. J. Ruiz, and D. M. Blei, "Topic modeling in embedding spaces," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 439–453, 2020.
[51] D. GOODIN, "Patch fixing critical Log4J 0-day has its own vulnerability that's under exploit," https://arstechnica.com/information-technology/2021/12/patch-fixing-critical-log4j-0-day-has-its-own-vulnerability-thats-under-exploit/, 2021, [Online; accessed 16-December-2021].
[52] N. Pankov, "Critical vulnerability in Apache Log4j library," https://www.kaspersky.com.au/blog/log4shell-critical-vulnerability-in-apache-log4j/30102/, 2021-12-11, [Online; accessed 7-March-2022].
[53] F. Wortley, F. Allison, and C. Thompson, "Log4Shell: RCE 0-day exploit found in log4j 2, a popular Java logging package," https://www.lunasec.io/docs/blog/log4j-zero-day/, 2021-12-19, [Online; accessed 7-March-2022].
[54] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.
[55] I. Security, "IBM Security X-Force Threat Intelligence Index2022," https://www.ibm.com/downloads/cas/ADLMYLAZ, 2022, [Online; accessed 6-July-2022].