Extracting Semantic Information from Online Conversations to Enhance Cyber Defense

by

Kazuaki Kashihara

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2022 by the
Graduate Supervisory Committee:

Chitta Baral, Chair
Adam Doupé
Eduardo Blanco
Ruoyu Wang

ARIZONA STATE UNIVERSITY

December 2022

ABSTRACT

Recent advances in techniques allow the extraction of Cyber Threat Information (CTI) from online content, such as social media, blog articles, and posts in discussion forums. Most research work focuses on social media and blog posts since their content is often contributed by cybersecurity experts and is usually of cleaner formats. While posts in online forums are noisier and less structured, online forums attract more users than other sources and contain much valuable information that may help predict cyber threats. Therefore, effectively extracting CTI from online forum posts is an important task in today's data-driven cybersecurity defenses. Many Natural Language Processing (NLP) techniques are applied to the cybersecurity domains to extract the useful information, however, there is still space to improve. In this dissertation, a new Named Entity Recognition framework for cybersecurity domains and thread structure construction methods for unstructured forums are proposed to support the extraction of CTI. Then, extend them to filter the posts in the forums to eliminate non cybersecurity related topics with Cyber Attack Relevance Scale (CARS), extract the cybersecurity knowledgeable users to enhance more information for enhancing cybersecurity, and extract trending topic phrases related to cyber attacks in the hackers forums to find the clues for potential future attacks to predict them.

To my beloved and supportive family.

ACKNOWLEDGEMENTS

I would like to thank my graduate supervisory committee chair and Ph.D. advisor, Dr. Chitta Baral, for his supervision throughout the dissertation. I appreciate his advice during the development of my research. In addition, I would like to thank my graduate supervisory committee members; Dr. Ruoyu Wang, Dr. Adam Doupè, and Dr. Eduardo Blanco, who carefully invested their attention and time to review my work. Especially, I could receive many feedback for cybersecurity side of aspects from Dr. Wang and Dr. Doupè , and for Natural Language Processing (NLP) side of aspects from Dr. Blanco.

I appreciate my lab mates from Cognition & Intelligence Lab for their support and collaboration with several projects. I have enjoyed discussion with research ideas, and worked together for the projects in day and night, week and weekend, and semester and holidays. Kuntal Pal, Swaroop Mishra, and Dr. Pratyay Banerjee gave me lots of feedback and suggestions of my work, and collaborated on some research papers. A former lab mate, Dr. Arpit Sharma, is a good friend and I enjoyed discussion about research and also shared some of our common hobbies, hiking.

I also want to thank Dr. Xuerong Feng, Dr. Janaka Balasooriya, and Lecturer Steven Osburn for the TA or GSA positions and moments of learning and teaching that I enjoyed helping students. It was a challenge to teach in huge class rooms in online and in person, however, it was a great opportunity to grow not only students but also myself during the classes.

I truly appreciate Dr. Yoshihiko Kobayashi for giving the opportunity to work his study abroad program in Japan. It was a great for me not only visiting Japan through the class but also visiting many state-of-the-art technology companies and museums to know how the technology helps our daily life. In addition, he is a great

mentor to live in the United States as Japanese.

Finally, I need to say thank you to my beloved family; My supportive wife, Heather, my parents, Hideaki and Kazuko Kashihara, and my parents in law, Graham and Sheila Wheeler. I appreciate my wife and father in law to check my papers numerous times for proof reading, and my parents to support to move to the United States to pursue my degree. They are supportive and cheered me to accomplish this long journey. It is sad that my father is not able to see how far I achieved, however, I believe he looks after me from above.

This dissertation would not have been possible without the support of the data from CYR3CON, Inc. (Cyber Security Works)

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Cybersecurity is a rapidly developing field in today's ever increasing technology era. It is concerned with protection of computer systems and internet services from damage and disruption caused by unauthorized third-party players. This field is growing in importance due to increasing reliance on computer systems, the internet, wireless networks such as Bluetooth and WiFi, and due to the growth of smart devices including smart phones, televisions, and various small devices that constitute the Internet of Things (IoT). With the rise of technology, all these systems are posed with an unbounded array of threats and attacks, that is becoming a challenge for security experts to build a system for timely risk management and prevention. Social media and magazines are platforms where we are at least notified of the various security incidents happening on a day-to-day basis all around the world. These reports are the first step in accumulating information about the various upcoming threats, their sources and target systems, and the technologies they incorporate. This can be useful in countering the incidents in future. In addition, National Science Foundation's program "National Artificial Intelligence (AI) Research Institutes" [106] has one of the themes "Intelligent Agents for Next-Generation Cybersecurity" and the theme mentioned an example that Artificial Intelligence (AI) will be able to analysis across multiple kinds of data for modeling cybersecurity threats including natural language intelligence reports from threat reports and dark web chatter.

According to Benjamin et al. [21], signals of impending attacks are more likely to be visible in the open public data source such as social media and discussion forums due to the growth of cyber threats. Cyber attackers exploit vulnerabilities

using tools, techniques, and tradecrafts through five steps according to [135]: (i) identify vulnerabilities, (ii) acquire the necessary expertise and tools to use, (iii) choose targets, (iv) recruit participants, and (v) plan and execute the attack. Other actors such as system administrators, security analysts, and even potential victims may discuss or share vulnerabilities, threats, or coordinate defense against exploits and various cyber attacks. These discussions are mainly conducted in online forums, blogs, and social media, thereby creating potential signals to identify an upcoming attack or a new vulnerability [130]. There are several previous works that focus on using web source as a signal for predicting high risk vulnerabilities or exploits [107, 10, 14, 12, 108] while other recent works have used social media or blogs [61, 20, 24, 130, 15].

The three data sources: social media, blogs, and discussion forums, are very different in nature. Each of them has a unique type of signal. For instance, content from social media such as Twitter and cybersecurity blogs is cleaner than discussion forums. The former is usually written by security experts; it is highly topical and rich in technical terms. On the other hand, the latter is a collection of information from cybersecurity related discussion forums on diverse topics. The discussion forum posts may include detailed information such as tutorials on exploits or vulnerabilities and data dumps of Personally Identifiable Information (PII), and non cybersecurity topics such as drug trade and selling pirate products. According to Nunes et al. [107], the writing style within the forums is often intentionally difficult to parse since they use words concatenated into new terms, multiple languages used in a post, and inaccurate

grammar.

1.1 Motivation

Some systems predicting indicators of attacks or vulnerability exploitation [11, 13] are using discussion forum posts, however, they select the posts containing vulnerability IDs (Common Vulnerabilities and Exposures (CVE) ID such as CVE-2017-0144 or Microsoft Security Bulletin (MSSB) ID such as MS17-010). They do not use the posts containing cybersecurity related information if the posts do not have vulnerability IDs. Zero-day vulnerabilities are unknown security flaws or bugs in software, firmware, or hardware which the vendor does not recognize, or does not have an official patch or update to address the vulnerability. Zero-day vulnerabilities are usually not assigned CVE ID unless the vulnerability is reported by a researcher or discovered as a result of an attack. Thus, their approaches miss the posts about Zero-day vulnerabilities without their CVE IDs. Moreover, the work [107] uses a classifier that is a machine learning technique using an expert-labeled dataset to detect relevant topics or not. However, they did not mention the way of labeling. Therefore, we need to filter the posts in the forums with clear rules to select cybersecurity related posts for providing the posts missed in the previous approaches. Then, our proposed approaches extract semantic information from the filtered posts for named entity recognition, forum structure construction, and detecting trending topics.

1.2 Contributions of the Research

In this research, we focus on how we can extract semantic information from hackers conversations in the discussion forums to enhance cyber defense by finding key users who are highly skilled and extracting the cyber attack trending topics in forums. To achieve this task, we propose new human-machine interaction corpus generation

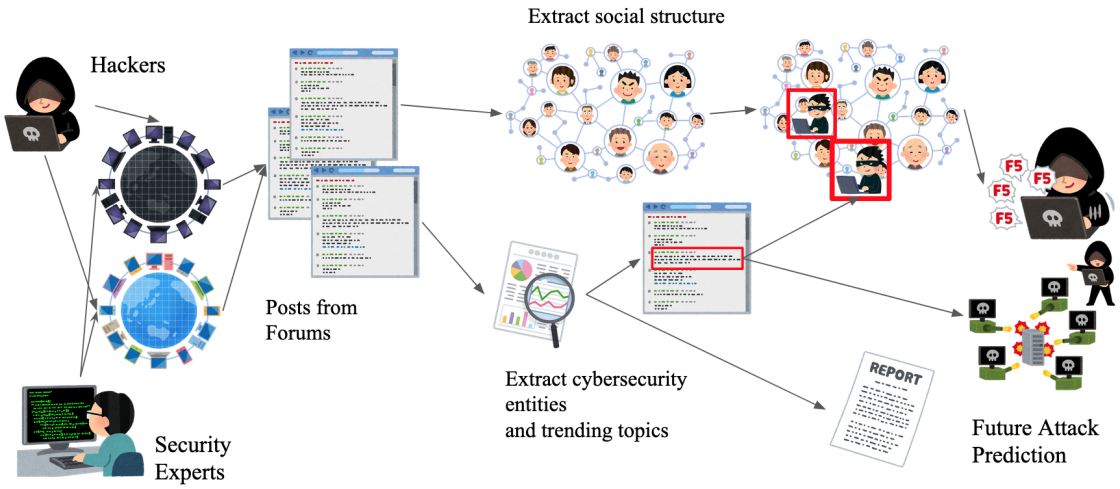

Figure 1.1: Overview of the Research

method for Named Entity Recognition (NER) and a unified model for multi-task approach in the cybersecurity domain, thread structure prediction methods to build social structure of hacker forums for social network analysis, and Cyber Attack Relevance Score (CARS) to scale the posts of forums to filter cyber attack related posts. Then, we use them to find key users with CARS and user interaction, and detect cybersecurity trending topic phrases considering cyber attack relevance.

The details of each sub-task is described as follows. Figure 1.1 provides an overview of different sub-tasks under this research.

1.2.1 Human-Machine Interaction for Improved Cybersecurity Named Entity Recognition Considering Semantic Similarity with Small Keyword Dictionary

The automated and timely conversion or extraction of cybersecurity information from unstructured text from online sources is important and required for many applications. Named Entity Recognition (NER) is used to detect the relevant domain entities such as product, attack name, malware name, and hacker group name. To train a new NER model for cybersecurity, traditional NER requires a training cor-

pus annotated with cybersecurity entities and state-of-the-art methods require time-consuming and labor intensive feature engineering. We propose a Human-Machine Interaction method for semi-automatic labeling and corpus generation for cybersecurity entities. Our method evaluates the learned NER model with the sentences that we collected in the training process, and the user selects only the correct pair of the named entity and its category for next iteration training. Thus, each iteration gets better training corpora to train the NER model. Some entities are ambiguous since the word or phrase has multiple meanings. We introduce a new semantic similarity measure and determine which category the word belongs to based on this semantic similarity of the entire sentence. The experimental evaluation result shows that our method is better than existing methods in finding undiscovered keywords of given categories.

However, the semantic similarity measurement in the method to solve the ambiguous keywords requires the specific category names even if non cybersecurity related categories. Thus, we also introduce another semantic similarity measurement using a text category classifier which does not require to give the specific non cybersecurity related category name. The performance of the two semantic similarity measurements are compared, and the new measurement performs better. The experimental evaluation result shows that our method with the training data that is annotated by a small dictionary provides almost the same level of performance as the models that are trained with fully annotated data. If we use 10% of the original keyword dictionary for generating the annotated dataset, we iterate our method three times and the model reaches nearly 80 F1 score. Then, if we use 70% of the original keyword dictionary for generating the annotated dataset, we iterate our method three times and the model reaches nearly 89 F1 score which is similar score of the existing methods with the fully annotated dataset. (See details at Chapter 2 and Table 2.4)

We also introduced an unified text-to-text multi-task model in cybersecurity domain (UTS), combining ten publicly available cybersecurity datasets involved in eight NLP tasks. We compared the performance of the same dataset we used in the previous proposed methods. The results show that T5 and UTS with T5 perform the highest F1 scores (95.97 and 98.81 respectively) for the full size of training dataset. Thus, at least, NER task improves the performance with the multi-task and cross dataset trained model. (See details at Chapter 2 and Table 2.3.)

1.2.2 Social Structure Construction from the Forums using Interaction Coherence

Social network analysis is one of the important tasks for the cybersecurity field to identify some potentially important members in the hackers' forums and communities. To create the social network of the forums and communities, we need to understand the social structure of them. Once the metadata from each forum is collected, the social structure is constructed based on the users' interactions such as who replies to whose post and when the response is posted. However, most of the hackers forums are unstructured, and there is no way to find the interaction between the users. We introduce the Next Paragraph Prediction (NPP) method that returns true if a response is a direct response of the previous post. This method helps to create the social structure from an unstructured forum. We also apply instructional prompts to the training process of NPP method to improve the performance (NPP-IP). The experimental result shows that NPP and NPP-IP performs 4 - 50 times better in F1 score than the existing methods. (See details at Chapter 3, Table 3.3, and Table 3.4.)

In addition, following procedural texts written in natural languages is challenging. we must read the whole text to identify the relevant information or identify the instruction flows to complete a task, which is prone to failures. If such texts are structured, we can readily visualize instruction-flows, reason or infer a particular

step, or even build automated systems to help novice agents achieve a goal. However, this structure recovery task is a challenge because of such texts' diverse nature. This new approach proposes to identify relevant information from such texts and generate information flows between sentences. This method is a new method for building flow graphs for procedural texts using Graph Neural Networks (GNNs). However, this can transfer to building social structure. We evaluate this method (FS) to the social structure evaluation dataset to compare our two methods (NPP and NPP-IP). FS method shows the highest F1 score (0.53 F1 score, 2.6 - 53 times better F1 score than the existing methods) in the Reddit dataset, however, FS does not perform as high as NPP-IP in the Hacker Forums dataset. (See details at Chapter 3, Table 3.3, and Table 3.4.)

1.2.3 Finding Key Users Considering User Interactions and Cyber Attack Relevance of the Hacker Forums' Posts

Recent advances in techniques allow the extraction of Cyber Threat Information (CTI) from online content, such as social media, blog articles, and posts in discussion forums. Most research work focuses on social media and blog posts since their content is often contributed by cybersecurity experts and is usually of cleaner formats. While posts in online forums are noisier and less structured, online forums attract more users than other sources and contain much valuable information that may help predict cyber threats. Therefore, effectively extracting CTI from online forum posts is an important task in today's data-driven cybersecurity defenses.

We introduce a new measurement of online posts, called Cyber Attack Relevance Scale (CARS), where posts with higher CARS scores contain more detailed cybersecurity information. We then develop a machine-learning-based solution to rate online posts by their relevance to cybersecurity. Finally, we create a human annotated

dataset that comprises posts from cybersecurity-related subreddits. We evaluate our measurement and solution using Random Forest, Linear Support Vector Classification, Multinomial Naive Bayes, Logistic Regression, Convolutional Neural Network, and Bidirectional Long Short Term Memory network for the classifiers to train on the dataset. Results show that our models can predict CARS score with an average accuracy of 84.6%. We also evaluate the top three classifiers with posts from other online forums that are cybersecurity-related. Overall, CARS scores can be used to effectively find cybersecurity-related posts in online discussion forums. Defenders can use CTI extracted from online discussion forums to predict the risk of cyber threats.

Compared to the existing methods, our proposed approach with Next Paragraph Prediction and CARS can extract more users who have more knowledge and information about exploitation or cyber attacks. Since there are many topics discussed in hacker forums including non cybersecurity related topics, the existing methods extracted some users who do not discuss any cybersecurity related topics. In contrast, our proposed approach predicts the thread structure based on the context and weight of each post's relations with CARS, and this helps extract more cybersecurity knowledgeable users from the forums. For instance, a user whom only our method found in a forum aggressively recruits some skilled programmers (hackers) to breach specific banks. We can assume that these banks are targeted. Another user whom only our method found in another forum posted about zero-day vulnerability topics, and claimed the exploitation of some of the vulnerabilities. These users are useful users to understand the potential attacks and trending cybersecurity related topics. (See details at Chapter 4, especially Analysis and Discussion Section, and Figure 4.3 and Figure 4.4.)

1.2.4 Detecting Cybersecurity Trending Topic Phrases Considering Cyber Attack Relevance

Cybersecurity experts are finding new approaches to mitigating cyber threats against the computational infrastructure of companies and society. One of these approaches is detecting the trending topics that the forums are discussing in the specific time frame. Topic modeling is used for discovering latent structure as topics in a large collection of documents. We propose a new method for early cyber threat detection. This method combines topic modeling using distributed representations of forum posts and words due to their ability to capture semantics of words and documents. Then, it clusters the posts and calculates the representing phrases of each cluster through Cluster-Phrase-TF-IDF (cp-TF-IDF) considering the cyber attack relevance of posts. The experimental evaluation shows that our clustering part provides similar results to the other related approaches, and extracts topic phrases which are significantly more informative than the other approaches' topic words such as tool names with specific version number, and attack type and the target names. In addition, we discover that some of the extracted topic phrases linked to discussion of cyber incidents or attacks prior to the events happened.

Compared to the existing methods, our methods can extract many cyber attack or incident related phrases that are more informative than the words from the existing methods. Especially, some of the words and phrases we extracted are related to the cyber attacks such as tool names, cyber criminal group names, and target names. Many of the words and phrases are mentioned prior to the attacks. One of our methods, cp-TF-IDF with CARS, could detect topic phrases that led to the malware attacks in 2020 1.44 - 1.86 times more than existing methods, and lead to the Phishing attack in 2020 2.37 times more than existing methods. For instance, only

our cp-TF-IDF method found a game name and hacking keywords a week prior to the source code of the game that was leaked online in April 2020. In addition, only our cp-TF-IDF with CARS method extracted 'ransomware attacks', 'K-12 educational institutions', and 'security hole' in a cluster of topics over a week before a school district in the U.S. hit cyber attack and leaked sensitive data in December 2020. Thus, our methods will be able to expand for predicting future cyber attacks based on the hacker conversations and this will improve the cyber defense. (See details at Chapter 5, especially Discussion section and Figure 5.11 and Figure 5.13.)

1.3 Summary of Applications and Impact

Deep analysis and understanding of hacker forums (communities) can be an important tool for building better cyber attack prediction systems. From the perspective of defenders, knowing the trends of the hacker forums allows effective action to be taken, enabling protection before the attack happens. For instance, finding who are the highly skilled and influential hackers in hacker forums, and filtering cyber attack related posts from hacker forums and finding topics of the filtered posts, can effectively observe the hacker forums instead of checking every conversations in the forums, and help cyber attack prediction systems identify potential threats. We believe this capability affords a risk reduction of the possible targets such as organizations, platforms, and products, and enhance the performance of these prediction systems while forecasting future cyber attacks.

1.4 Organization and Summary of Research Contribution

The remainder of this dissertation is organized as follows:

- **Chapter 2: Human-Machine Interaction for Improved Cybersecurity Named Entity Recognition Considering Semantic Similarity with**

Small Keyword Dictionary. In this chapter, we describe the semi-automated corpus generation method with small keyword dictionary for cybersecurity NER and a unified model for multi-task in cybersecurity domain. Our NER methods are reported to [70, 69].

- **Chapter 3: Social Structure Construction from the Forums using Interaction Coherence**. In this chapter, we propose three different methods, Next Paragraph Prediction (NPP), NPP with Instructional Prompts (NPP-IP), and Flow Structure (FS), to build the social structure from unstructured forums for social network analysis. NPP is reported to [71] and the original idea of FS is reported to [111].

- **Chapter 4: Finding Key Hackers with Cyber Attack Relevance Scale considering User Interaction**. In this chapter, we introduce Cyber Attack Relevance Scale (CARS) and develop a machine-learning-based solution to rate online posts by their relevance to cybersecurity. Then, we develop a system to identify key hackers in the hacker forums with combining CARS model and Social Structure Construction method.

- **Chapter 5: Detecting Cybersecurity Trending Topic Phrases Considering Cyber Attack Relevance**. In this chapter, We introduce a new method, TrendTopicExtractor, to detect cybersecurity trending topics through clustering the posts and extracting the topic phrases of each cluster considering cyber attack relevance scale of each post.

- **Chapter 6: Conclusion**. In this chapter, we review the main ideas and results presented in the dissertation and considering some directions for future work.

Table 1.1 summarizes the contributions presented in each chapter of this disser-

Chapter	Contribution
2	Semi-automated corpus generation method for cybersecurity NER task. Multi-task model in cybersecurity NER task.
3	Building social structure from unstructured forums.
4	Identification of key users on hacker forums.
5	Detecting cyber attack related topics from hacker forums.

Table 1.1: Contributions of This Dissertation.

tation.

Chapter 2

HUMAN-MACHINE INTERACTION FOR IMPROVED CYBERSECURITY NAMED ENTITY RECOGNITION CONSIDERING SEMANTIC SIMILARITY WITH SMALL KEYWORD DICTIONARY

2.1 Introduction

In many cybersecurity applications, Named Entity Recognition (NER) has been used to identify the entities of the interest such as the name and versions of the vulnerable software, those of vulnerable components, and those of underlying software systems that vulnerable software depends upon [44, 137]. A NER model pinpoints entities based on the structure and semantics of input text, and tracks down entities that have never been observed in the training data. In those works, in general, the training data for a NER model is created by manual annotation. To minimize the manual annotation efforts, the automated labeling method [25], feature engineering methods [36, 104], deep learning (DL) methods [25, 128, 150], and a transfer learning method [44] are introduced. The automated labeling method uses the database matching, heuristic rules, and relevant terms gazetteer. Any of the above methods using feature engineering, DL, or transfer learning methods requires some annotated training dataset to train the model. There are specific terms in this domain which in general English have different meanings and may not be an entity. For instance, "Wine" has meanings of a software name and a drink. The automated labeling method does not support any ambiguous keywords such as "Wine" to label correctly. In this paper, we address the problem of automated labeling method by taking a different approach. We introduce a new semantic similarity measurement that helps

to determine the suitable category of an ambiguous keyword. Our method requires small dictionary that has the pairs of keywords and their categories, and raw text data. Then, automatically generates the training data for an NER model.

The current NER tools that show state-of-the-art performance in the cybersecurity field are based on feature engineering or the Deep Learning. In addition, they require ample training data, which is generally unavailable for specialized applications, such as detecting cybersecurity related entities. The major issues are: it relies heavily on the experience of the person, the lengthy trial and error process that accompanies that, and it also relies on look-ups or dictionaries to identify known entities [36, 104]. These dictionaries are hard to build and harder to maintain especially with highly dynamic fields, such as cybersecurity. For instance, the Common Vulnerabilities and Exposures (CVE) ID is easily extracted by the regular expression: "CVE-\d{4}-\d{4,7}". However, software names, filenames, version information, and OS names are unique names and they are hard to identify through pattern matching methods. Thus, it requires human experts' annotations. These activities constitute the majority of the time needed to construct these NER tools. In addition, these tools are domain specific and do not achieve good accuracy when applied to other domains. However, the requirement of the available features to the training and test data will not only slow down the annotation process, but also diminish the quality of results.

Our first work [70] introduces a semantic similarity measurement and generate a new NER corpora for cybersecurity entities with human-machine interactions. The NER model with this corpora performs better than the existing methods in finding undiscovered keywords of given categories. However, the proposed semantic similarity measurement needs to have not only the cybersecurity related category for an ambiguous keyword but also specific category name for the other categories such as "wine" as "software" or "drink".

We introduce a new semantic similarity measurement and determine which category the word belongs to based on the semantic similarity of the entire sentence. This measurement does not require to give any specific category name for non cybersecurity related categories. This improves to preprocess for the semantic similarity measure algorithm. We apply this measurement to our previous method. The learning part of the method requires only the list of the pairs of the cybersecurity entities and their categories. This method generates the high quality training dataset from the small number of keywords of the target categories in cybersecurity field. The evaluation with two cybersecurity NER corpus shows that our approach with new semantic similarity measurement and the given small dictionary performs almost same performance of the manually annotated datasets.

In addition, we also introduce unified text-to-text multi-task approach in cybersecurity domain. Unlike other domains, in Cybersecurity domain the nature of texts is quite diverse (natural language text, URLs, malware reports, system calls, source code, binaries, decompiled code, network traffic, software logs [123, 76, 126, 87, 138, 25, 34, 164, 118]). This led to the introduction of specific models capable of performing individual tasks like cyber-bullying detection CyberBERT [93], cybersecurity claim classification CyBERT [16]. Apart from this, there is a scarcity of large-scale publicly available annotated datasets. These challenges demand the need of developing *robust* models capable of performing *multiple tasks* by *learning from many datasets together*. Hence, we introduce an ***U****nified (multi-task)*, ***T****ext-to-Text* Cyber**S**ecurity (*UTS*) model.

We train two transformer-based generative model models, BART [81] and T5 [27], in a multi-task setting on *eight* fine-grained NLP tasks involving *eight* datasets in the cybersecurity domain. We used task based prompt prefixes to help the models learn the task instead of learning specific datasets. We make the model more robust by

training on variety of texts. We experiment in two few-shot settings to evaluate our UTS approach.

The main contributions of this chapter include:

- We present a bootstrapping method to train an NER system for cybersecurity domain entity with small number of initial dictionary.
- We introduce a new semantic similarity measurement for solving ambiguous entities case. The semantic similarity measurement helps to determine which category an ambiguous entity should belong to.
- We empirically perform experiment. The result shows that our approach with the small number of keyword coverage in each category performs almost similar performance of the other DL methods with full annotated data.

2.2 Related Works

Aproaches to NER are mainly three types: rule-based, machine learning/statistical-based [35], or mixed [113]. The rule-based methods are a combination of gazette-based look ups and pattern matching rules that are hand-coded by a domain expert in most of the cases. These rules consider the contextual information of the entity to determine whether candidate entities from the Gazette are valid or not. There are a variety of models used in the statistical-based NER approaches such as Maximum Entropy Models [32], Hidden Markov Models (HMMs) [90], Support Vector Machine approach [65], Perceptrons approach [29], CRFs approach [91], or neural networks approach [36]. CRFs approach is one of the most successful NER approaches. It is because CRFs use the conditional probability property instead of the independence assumption mainly used in HMMs and also avoid label bias problems and weaknesses of other Markov models derived from Maximum Entropy Markov Models and graphic

models. In this model, the label of any entity is modeled as dependent on the labels of the preceding and following entities in a specified window and the size of the window varies by task. Stanford NER [46] is an example of CRF-based NER system. In our context, it would be insufficient to assume a relationship between individual posts and doing so would hurt accuracy.

Various methods have been applied to extract entities and their relations in the cybersecurity related domains. For example, Jones et al. [67] implemented a bootstrapping algorithm that requires little input data to extract security entities and the relationship between them from the text. A SVM classifier has been used by Mulwad et al. [103] to separate cybersecurity vulnerability descriptions from non-relevant ones. They require pre-process or annotated corpus. The automatic labeling method for cybersecurity [25] uses Database Matching (string pattern matching), Heuristic Rules (rule based matching), and Relevant Terms Gazetteer (extended string matching that if a phrase contains a keyword in the database, the phrase is annotated with the label in the database). However, there are specific terms in cybersecurity domain which in general English have different meanings and may not be an entity. For instance, "Windows" and "Wine" are an OS name and an application name in cybersecurity field, but they have different meanings in general English. The above methods do not support any ambiguous keywords to label correctly. Sirotina and Loukachevich [142] provide the corpora of 10 cybersecurity related categories in Russian and the corpora is manually annotated by human experts.

Recently, the Deep Learning (DL) methods are used for NER. DL is an enhanced classical neural network model with naturally learning non-linear combinations. For instance, the Conditional Random Fields (CRFs) can just learn linear combinations of the defined features. This reduces the human work of tedious feature engineering [25, 128, 150]. The recent work by Gasmi et al. [50] relies on Long Short-Term Mem-

ory (LSTM) and the Conditional Random Fields (CRFs) method for cybersecurity NER that applies the LSTM-CRF architecture suggested by Lampal et al. [78]. The architecture combines LSTM, word2Vec [98] models, and CRFs. The input for this method is an annotated corpus in the same format as the CoNLL-2000 dataset [134]. In the recent days, many applications of DL have been leverage in the field of cybersecurity [152, 151, 127]. However, any of the above methods using feature engineering or DL methods requires some annotated training dataset to train the model. There are two challenges. First, it requires some certain number of annotated sentences to make the decent performance model. Second, the sentences are annotated by experts (human in many cases) and the human makes the incorrect annotation or miss to annotate some words or phrases.

2.2.1 spaCy

spaCy [60] provides an exceptionally efficient statistical system for named entity recognition in python, which can assign labels to groups of tokens which are contiguous. It provides a default model which can recognize a wide range of named or numerical entities, which include: company names, locations, organizations, and product names. Apart from these default entities, spaCy enables the addition of arbitrary classes to the entity-recognition model, by training the model to update it with newer trained examples. spaCy uses a convolutional neural network (CNN) to train the model. The statistical models in spaCy are custom-designed and provide an exceptional performance mixture of both speed, as well as accuracy.

2.2.2 BERT

BERT (Bidirectional Encoder Representations from Transformer) [41] has two steps: pre-training with large raw corpus, and fine-tuning the model for each task.

BERT is based on Transformer [148], which can catch the long distance dependency relations, because it is based on self-attention, and does not use RNN or CNN.

The input for BERT is a sentence, pair of sentences, or document, and it represents the sequence of tokens in each case. Each token is the summation of token embedding, segment embedding, and position embedding.

Each word is divided into sub-words, and the non-head part in the subwords will be assigned "##". For instance, "playing" is divided into "play" and "##ing" as subwords. If the input is two sentences, segment embedding gets the first sentence token as sentence A embedding, and the second sentence token as sentence B embedding (put "[SEP]" token between two sentences). In addition, the location of each token is learned as position embedding. The head of each sentence is marked with the "[CLS]" token. In the document classification task or two sentences classification task, the final layer of embedding of the token is the representation of the sentence or the two-sentences-set.

For text classification tasks, BERT takes the final hidden state $\mathbf{h}$ of the first token [CLS] as the representation of the whole sequence. A simple softmax classifier is added to the top of BERT to predict the probability of label c:

$$p(c|\mathbf{h}) = \text{softmax}(W\mathbf{h}),$$

where W is the task-specific parameter matrix. We fine-tune all the parameters from BERT as well as W jointly by maximizing the log-probability of the correct label.

2.2.3 Multitask Learning in Diverse domains:

In natural language domain, DecaNLP [92] introduced the approach of converting multiple task into single QA format to train and evaluate ten tasks. With the gradual introduction of stronger NLP generative models like GPT, T5 and BART,

the text-to-text unified models gained prominence. The multi-task approach have been shown to perform well in various domains like SciFive [122] in the biomedical domain, CodeT5 [155] in the source code domain, LEGAL-BERT [31] in legal domain and FinBERT [84] in financial service domain. Using "teacher forcing" for all tasks for training with a maximum likelihood objective, SciFive enables multitask learning. CodeT5 is a unified pre-trained encoder-decoder Transformer model and it can handle various tasks across various directions between program languages and natural languages.

2.2.4 Task-Based Unified Models:

Apart from these, there are individual task based unified models like Instruction-NER which expands the existing methods for sentence-level tasks to a instruction-based generative framework for low-resource named entity recognition [154]. In biomedical domain, KGNER [19] formulated the NER task as a multi-answer knowledge guided question-answer task and experimented with 18 datasets.

UnifiedNER [159] works on unifying span-based, nested and discontinuous NER tasks. UnifiedQA [72] showed that an unified training of QA tasks help in improvement of other QA tasks. Similar results are shown in common-sense reasoning tasks by Unicorn [85].

2.3 Methodology

In this section, we will provide an overview of our Human-Machine Interaction method that has two sub-parts which are inter-dependent (1) Learning part and (2) Evaluation part. The architecture of the proposed method is shown in Figure 2.1.

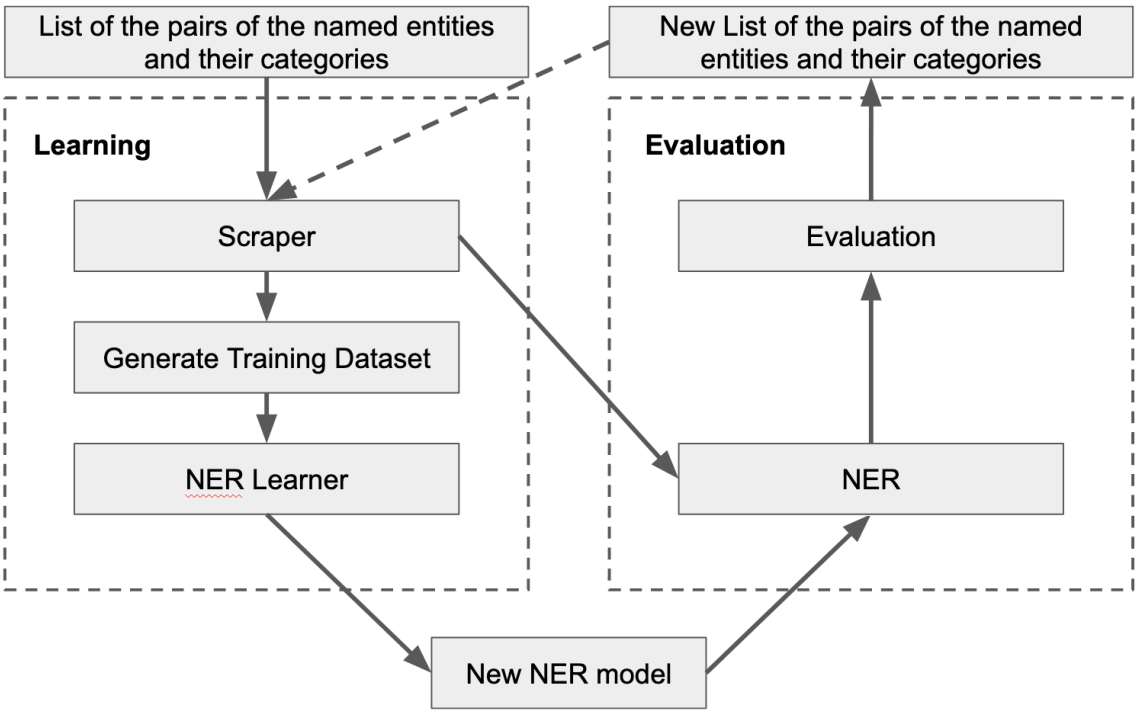

Figure 2.1: The Architecture of the Proposed Method

2.3.1 Learning part

The learning part is fully automated to generate the training data of the cybersecurity related tags for the customized NER model. The algorithm of Learning Part is shown in Algorithm 1. The learning part requires the list of the pairs of keyword (named entity) and its category as input. Cybersecurity incident reports and professionals' articles published online containing the keywords are labeled and the paired categories are assigned. Then, the Scraper function searches and extracts the incident reports that include one of the keywords, and returns the list of the sentences that contain the keyword from the reports.

The scraper algorithm is shown in Algorithm 2. Some keywords have multiple meanings and they are ambiguous since they belong to multiple categories. In Algo-

Algorithm 1 LearningProcess(*TrainList*)

```
1: TrainSentList = {}
2: for keywordPair ∈ TrainList do
3:     SentenceList = {}
4:     keywordPair is a pair of keyword and its categoryList (Category List).
5:     SentenceList add Scraper(keyword)
6:     for sentence ∈ SentenceList do
7:         if ‖categoryList‖ ≥ 2 and keyword appears in sentence then
8:             highestCat = SentCat(sentence, keyword, categoryList)
9:             if highestCat is one of the categories we annotate then
10:                TrainSentList add (sentence, keyword, highestCat)
11:            end if
12:        else if keyword appears in sentence then
13:            category = categoryList
14:            TrainSentList add (sentence, keyword, category)
15:        end if
16:    end for
17: end for
18: Train NERModel with TrainSentList
```

rithm 1, we introduce SentCat to decide to which category the ambiguous keyword is assigned in the given sentence based on semantic similarity of the category and the context. SentCat is described in greater detail in the next subsection.

Handling Ambiguous Meaning

Many keywords' meaning changes with the context. For instance, "Microsoft has released a security update to address an elevation of privilege vulnerability (CVE-

Algorithm 2 Scraper(*keyword*)

SentenceList = {}
siteList is the list of cybersecurity professionals' sites
for *site* ∈ *siteList* **do**
    reportLinks = the incident report links in *site* that contain *Keyword*
    for *link* ∈ *reportLinks* **do**
        Extract all sentences in the report from *link*
        SentenceList add the extracted sentences
    end for
end for
return *SentenceList*

2019-1162) in **windows**" and "an inventory of the network analysis classes for which you can set time **windows**". The "windows" in the first sentence means the operating system but the second one means the window of time. To avoid mislabeling, we introduce the semantic similarity of the sentence between ambiguous categories.

Let $S = w_1 w_2 \dots w_n$ be a sentence that has n words (w_i is ith word in the sentence where $1 \leq i \leq n$), and Nouns $= (n_1, \dots, n_k)$ be a set of nouns in the sentence S (k is the number of nouns in the sentence S and $k \leq n$). We are given a set P that has the pairs of ambiguous keywords and their categories $P = ((x_1, C_1), \dots, (x_m, C_m)$, where x_i is ith keyword and C_j is the set of jth keyword's categories $C_j = (c_1, \dots, c_l)$ where $1 \leq j \leq l$.

We define the similarity score of a word w_i and the category c_j as $\mathrm{Sim}(w_i, c_j)$ and its range is $[0, 1]$. Then, the semantic similarity score of the sentence S that contains an ambiguous keyword x_i with the category $c_j \in C_i$ is defined as

$$\mathrm{SemSim}(S, x_i, c_j) = \frac{\sum_{a=1}^{k} \mathrm{Sim}(n_a, c_j)}{k} \tag{2.1}$$

Algorithm 3 SentCat(*sentence*, *keyword*, *categoryList*)

highestCategory =“”
highestSimScore = 0
for *category* ∈ *categoryList* **do**
    nounList is the list of all nouns and noun phrases in the *sentence*
    simScore = 0
    for *noun* ∈ *nounList* **do**
        simScore+ = Sim(*noun*, *category*)
    end for
    $simScore = \frac{simScore}{\|nounList\|}$
    if *simScore* ≥ *highestSimScore* **then**
        highestSimScore = *simScore*
        highestCategory = *category*
    end if
end for
return *highestCategory*

If the ambiguous keyword x_i appears in the sentence S, the NER category $c \in C_i$ is determined by SentCat as follows:

$$\text{SentCat}(S, x_i, c) = \max_{c \in C_i} \text{SemSim}(S, x_i, c) \tag{2.2}$$

. The steps of SentCat are described in Algorithm 3.

2.3.2 Category Classification for Ambiguous Meaning Keywords

In our first work [70], we introduced a human-machine interaction framework for semi-automatic labeling and corpus generation for cybersecurity entities. The framework has the Training module and the Evaluation module. The Training module

collects sentences, annotates the keywords from the given dictionary and passes the generated corpora to the NER system to train the model. We introduced a semantic similarity measurement named *SentCat* to judge the suitable category for ambiguous keywords that can be annotated to multiple categories. This measurement requires all of the ambiguous categories since the measurement calculates the similarity of the sentence against each category and determines the highest similarity score's category as the suitable category.

Many keywords' meaning changes within the context. For instance, "Microsoft has released a security update to address an elevation of privilege vulnerability (CVE-2019-1162) in **windows**" and "an inventory of the network analysis classes for which you can set time **windows**". The "windows" in the first sentence means the operating system but the second one means the window of time. To avoid mislabeling, we introduce new approach: text category classification using BERT fine-tuning (CategoryClassifier method).

In the CategoryClassifier method, we build the text category classifier using BERT fine-tuning. The training data for this text category classifier is the pair of the sentences that contain the known ambiguous keywords and the category of each sentence (it must be one of the ambiguous keyword categories). For instance, let's assume an ambiguous keyword "wine" and it has two categories, "software" and "non-software". We use the two sentences and labeled them as follows: "Only if you drink French wine, if it's radiated Californian wine that makes you an alcoholic mutt." is labeled as "non-software", where as "Wine is not a virtual machine, just an api converter, it can also directly call Linux programs." is labeled as "software". These sentences and their labels are given to BERT fine-tuning for building the text category classifier for the ambiguous keyword categories. The steps of CategoryClassifier are described in Algorithm 4.

Algorithm 4 CategoryClassifier(*sentence*, *categoryList*)

Load the fine-tuned BERT Text category classifier model *classifier*
category = *classifier*(*sentence*)
if *category* ∈ *categoryList* **then**
    finalCategory = *category*
else
    finalCategory = *NONE*
end if
return *finalCategory*

2.3.3 Unified Text-to-Text CyberSecurity (UTS) model

We formulate this multi-task problem in a generative text-to-text approach. Given an input text $I = \{i_1, i_2, ..., i_n\}$ and a task T, the model should generate a stream of output tokens $O = \{o_1|o_2|...|o_n\}$ defined by the task. For classification and regression tasks $O = \{o_1\}$ which represents the class-name and floating-point value respectively. For named entity recognition and event-extraction tasks, each o_i represents entity and entity-type separated by a pre-defined marker i.e. $o_i = \{e_i * t_i\}$. The task (T) is formulated as an instruction to help the models to learn individual tasks in this setting.

Multi-Task Training: All the training datasets of these four broad NLP tasks - Classification, Named Entity Recognition, Event Extraction and Regression - are grouped together for joint training. Under classification tasks, there are four fine-grained tasks : Text, Sentence, Relation, and Token Classification. In event extraction, we added two tasks like Event Nugget Extraction and Event argument Extraction. We parse the textual output generated by the models and evaluate the UTS models on test data of each of the corresponding datasets. To avoid confusion of the

model in identifying similar yet textually different categories, we use unique mapping of the entity types for all extraction tasks.

Prompt-Based Approach: We use task control codes as prompt-prefix for training the models in a multi-task setting so that it learns to perform each task instead of learning for any particular dataset. This helps the model to learn from more examples for each task. We pretend task acronyms CLS, NER, EVNT, REG with the input for classification, named-entity recognition, event-extraction and regression tasks respectively.

In Figure 2.2 shows, we propose a **U**nified (multi-task) **T**ext-to-Text Cyber**S**ecurity (*UTS*) approach to fine-tune the model on multiple tasks at a time using a task control code as the source prompt.

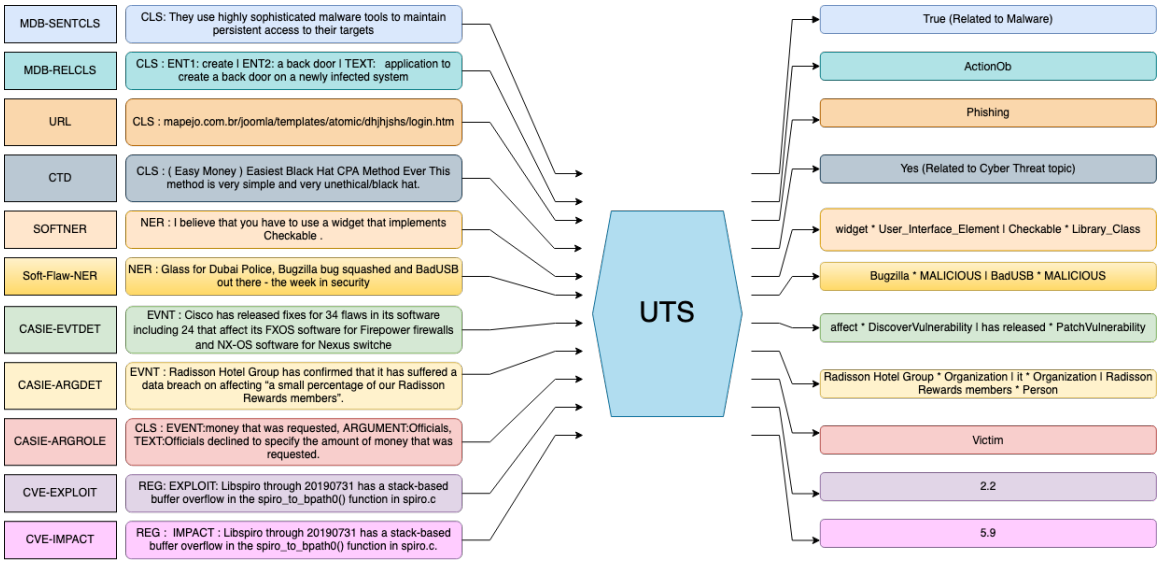

Figure 2.2: Illustration of UTS (Unified Text-to-Text CyberSecurtiy) Model

2.4 Evaluation

2.4.1 Data

We evaluate our method with Auto-labeled Cyber Security domain text corpus (we call Auto-labeled data) provided by Bridges et al. [25] comprising of around 15 categories was used in this work, and Russian Sec_col collection (We call Sec_col data) by Sirotina and Loukachevich [142] comparing of 10 categories was used in this work. We use spaCy in the NER model training part.

In Auto-labeled data, each word in the corpus is auto-annotated with an entity type. We joint each word in a sentence in a separate line into a sentence in order to feed the data into our method. The total number of the sentences is 15,781. For the evaluation our method, we convert the entities in each word into the categories, for instance, we merge "buffer: B-Relevant_Term" and "overflow: I-Relevant_Term" into "buffer overflow: Relevant_Term". Table 2.1 shows the statistics of the number of unique keywords in the dataset. We call the dictionary that contains these unique keywords of each category the unique full dictionary. In addition, the number of ambiguous keywords that have multiple categories is 153. The dataset is divided into three subsets that is training, validation, and testing consisting of 70%, 10%, and 20% sentences respectively.

Sec_col data consists of 855 texts (posts and forum publications and each text has multiple sentences) from SecurityLab.ru website. Table 2.2 shows the statistics of the number of unique keywords in the dataset. We also call the dictionary that contains these unique keywords of each category the unique full dictionary. In addition, the number of ambiguous keywords that have multiple categories is 224. We follow the evaluation way of [142] and do 4-fold cross-validation.

For preprocessing to use CategoryClassifier, we fine-tuned the BERT model. In

Category	# of unique keywords
Application	4335
Relevant_Term	193
Vendor	605
Version	7733
Update	222
OS	74
Function	1283
File	2426
Hardware	275
Method	107
CVE_ID	447
Parameter	270
Edition	58
Programming_Language	3
Language	2

Table 2.1: The Statistics of Unique Keywords in the 15 Categories of Auto-labeled Data

Auto-labeled data, we fine-tuned the model with the top 10% frequent ambiguous keywords (15 keywords from 153 ambiguous keywords from the original corpus) and 1,819 sentences that contain at least one ambiguous keyword with the ambiguous keyword's category as the sentence label from the training dataset (70% of the original corpus). This 1,819 sentences are divided into three subsets that is training, validation, and testing consisting of 70%, 10%, and 20% sentences respectively. Figure 2.3 shows the loss and performance curves of training and validation of ambiguous Auto-labeled sentences. After the 10 epoch, the accuracy of Training and Validation is as

Category	# of unique keywords
Org	1328
Loc_Term	420
Person	781
Tech	1029
Program	1884
Device	318
Virus	328
Event	187
Hacker_Group	35
Hacker	11

Table 2.2: The Statistics of Unique Keywords in the 10 Categories of Sec_col Data

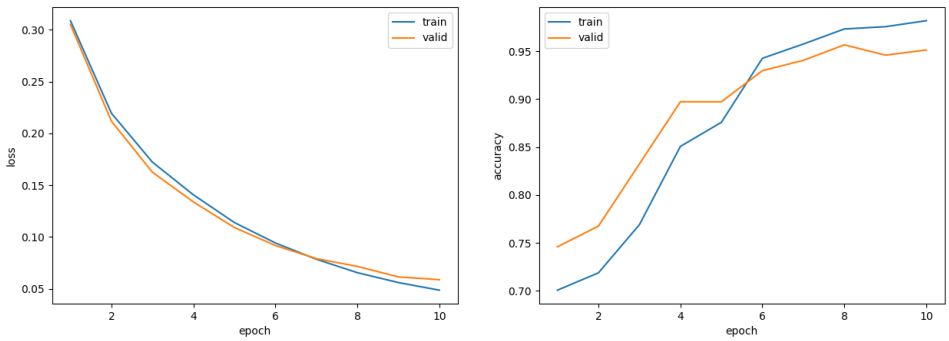

Figure 2.3: The Loss and Performance Graphs of CategoryClassifier's Training and Validation with the Ambiguous Keyword Sentences from Auto-labeled Data.

follows.

- Training: 98.2%
- Validation: 95.1%

Then, we compared with the accuracy of Testing data with CategoryClassifier and

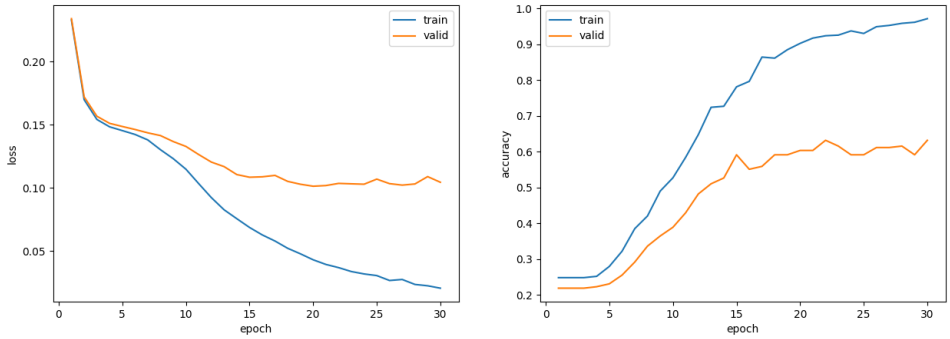

Figure 2.4: The Loss and Performance Graphs of CategoryClassifier's Training and Validation with the Ambiguous Keyword Sentences from Sec_col Data.

SentCat [70]. The result is as follows:

- SentCat: 82.8%
- CategoryClassifier: 88.4%

CategoryClassifier performs better than SentCat.

In Sec_col data, we fine-tuned the BERT model with all ambiguous keywords (224 keywords from the original corpus) and 2,425 sentences that contain at least one ambiguous keyword with the ambiguous keyword's category as the sentence label from the training dataset (70% of the original corpus). This 2,425 sentences are divided into three subsets that is training, validation, and testing consisting of 70%, 10%, and 20% sentences respectively. Figure 2.4 shows the loss and performance curves of training and validation of ambiguous Sec_col sentences. After the 30 epoch, the accuracy of Training and Validation is as follows.

- Training: 97.1%
- Validation: 63.2%

Then, we compared with the accuracy of Testing data with CategoryClassifier and SentCat. The result is as follows:

- SentCat: 59.3%
- CategoryClassifier: 61.1%

CategoryClassifier performs better than SentCat as well.

In our method's evaluation, we pick the most frequent $X\%$ of the original unique keywords of each category where X is 10, 20, 30, 40, 50, 60, 70, 80 and 90, and we fix the number of the ambiguous keywords as 10% of the original ambiguous keywords. In the evaluation part, we evaluate the learned model with the validation dataset and add the new keywords that are not listed in the dictionary but they are listed in the full dictionary with the right category to the next iteration dictionary. We use pre-trained models for spaCy; an English model "en_core_web_lg" for Auto-labeled data since all the posts are written in English, and a multi-language model "xx_ent_wiki_sm" for Sec_col data since Russian posts are written in not only Russian but also multiple languages including English, and this model is the only model supports Russian.

2.4.2 UTS model dataset

We prepare *ten* datasets involved in *eight* NLP tasks. For each of the datasets, we used the original train-test splits if mentioned in the paper otherwise, we split in 80:20 ratio respectively.

Classification

MalwareTextDB-V2: This dataset [123] is constructed from 83 APT reports each containing multiple cybersecurity-related natural language statements often mentioning about activities of malwares. We consider two tasks from this dataset for UTC.

They are : (1) *Sentence Classification* - classifying whether individual sentences are relevant to cybersecurity applications, and (2) *Relation Classification* - classifying the relation between two given entities.We take 68 documents as train and 15 documents test datasets. Each document has multiple sentences which we pre-process as each input sample.

SMS-SPAM: Another classification subtask is to classify the spam messages. This benchmark dataset [9] is for detecting SMS spam messages. The SMS-SPAM dataset is a combination of several publicly available SMS corpus and websites.

CyberThreatDetection: This dataset [126] was constructed from three hacker forums, Twitter, and the Dream Marker forum. Short forum posts were collected and labeled by humans into three categories. *Yes*, for posts that appear as malicious posts. *No*, for posts not related to hacker activity. *Undecided*, for posts where the annotator did not have enough information. For our experiment we counted the *Undecided* labels as *Yes* labels just like the original authors.

PhishStorm: The paper [87] introduced the PhishStorm dataset which included around 96k URLs. These URLs are labeled as normal or phishing, and were collected through PhishTank [1] , which is a crowd sourced project where people submit phishing URLs and were later confirmed by several people.

Event Detection

CASIE: This is the first cybersecurity Event Detection dataset [138] with five main types of events. We consider three tasks from this dataset: (1) *Event Extraction* (2) *Event Argument Detection* and (3) *Event Argument Role Detection.* Event Extraction is a task to extract event nuggets that are words or phrases that best express the event occurrence clearly. Event Argument Detection is a task to detect event arguments

[1] http://www.phishtank.com

that are event participants or property values. They can be tangible entities involved in the event such as person or organization, or attributes that specify important information such as time or amount. Event Argument Role Detection is a task to find roles between given event nuggets and event arguments. A role is a semantic relation between an event nugget and an argument. Thus, each event type specifies the roles it can have and constraints on the arguments that can fill them.

Named Entity Recognition

Stucco-Autolabeled: This dataset [25] is constructed from Common Vulnerabilities and Exposure (CVE) databases containing descriptions of information security issues from Jan, 2010 to Mar 2013. In Stucco-Autolabeled dataset, each word in the corpus is auto-annotated with an entity type. This dataset has 15 entity types.

softNER: This dataset [145] has 20 annotated entity types from 1237 StackOverflow QA pairs. The text is embedded with source codes constructs from many programming languages.

Soft-Flaw NER: Cybersecurity NER corpus 2019 corpus [131] consists of 1000 annotated tweets. The entities marked are usually the name of the software, system, device, or company with a security related issue, or the name of a malware. We use this dataset for our zero-shot evaluation.

Regression

NVD CVE metrics: The NIST National Vulnerability Dataset uses vulnerabilities found through the CVE (Common Vulnerabilities and Exposure) system. Human security experts assign a corresponding CVSS (Common Vulnerability Scoring System) vector, and from that, the exploitability and impact score for the vulnerability is calculated. We split the data from 2002 onward into train and test in a 1:1 proportion

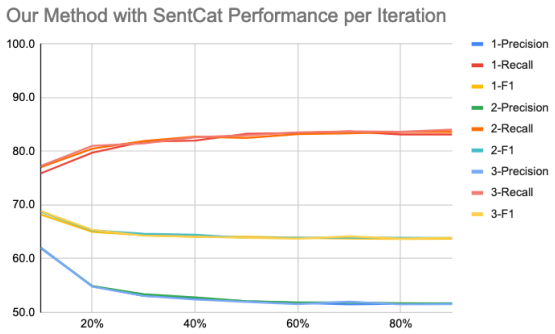

Figure 2.5: The Graph of the Performance of Our Method with SentCat in the Original Test Data.

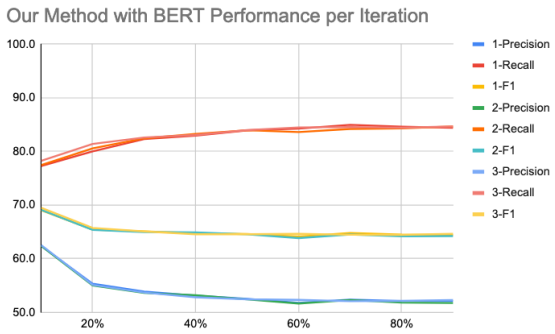

Figure 2.6: The Graph of the Performance of Our Method with CategoryClassifier (BERT) in the Original Test Data.

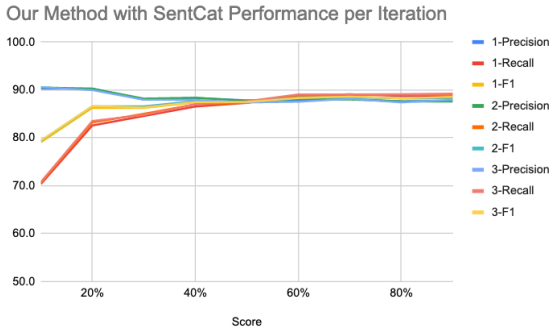

Figure 2.7: The Graph of the Performance of Our Method with SentCat in the Fully Annotated Test Data.

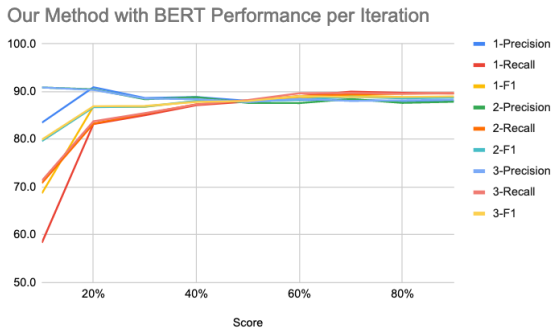

Figure 2.8: The Graph of the Performance of Our Method with CategoryClassifier (BERT) in the Fully Annotated Test Data.

as per the previous work [139] and directly generate the scores from the descriptions.

2.4.3 Results

In Auto-labeled data, we iterated three times in the experimental evaluation of our method. First, we evaluate the learned models with SentCat and CategoryClassifier (BERT) approaches for solving the ambiguous keywords through the original annotation. We compare the eight different approaches: LSTM-CRF, CRF [50], CNN-

CRF, RNN-CRF, GRU-CRF, Bidirectional GRU-CRF, Bidirectional GRU+CNN-CRF [141], and spaCy. The results are shown in Figure 2.5, Figure 2.6, and Table 2.3. In both SentCat and CategoryClassifier (BERT) cases, the dictionary size 10% gets the highest precision score in the dictionary size range between 10% and 90%, and the dictionary size 70% gets the highest recall score. The recall performance is higher than CRF method with full annotation, however, the precision performance is not as high as we expected. When we check the original annotation, we found some annotation issues. For instance, some categories like "Version" has "(" and ")" as the part of the keywords (phrase) like "4.0 before 4.0(16)", however, some cases are missing ")" such as "4.1 before 4.1(7". This incomplete paired cases are not accepted to annotate by spaCy and our model learned only the paired cases. In addition, the original annotation has many unnecessary characters that are included to the annotation such as comma, quote(s), and double quote(s). Since the performance is calculated by exact matches, our trained models can detect the part of the original annotated entities but they did not count correctly. Moreover, many unique keywords are not annotated in the original annotation, and our models detect them. On the other hand, the performances of our UTS models that train BART and T5 with multiple cybersecurity datasets of various tasks show that our UTS models can improve the performance from BART and T5 models trained with only Auto-labeled dataset. The UTS model with T5 (UTS-T5) reached the highest F1 score as well.

Since the original annotation accuracy has some doubt, we add the additional annotation from the full unique dictionary if a sentence has missed an annotation from the original. We call this new annotated test data as the fully annotated Test data on Auto-labeled dataset, and we evaluate our learned models with this fully annotated Test data. The results are shown in Figure 2.7, Figure 2.8, and Table 2.4. In both SentCat and CategoryClassifier (BERT) cases, the dictionary size 10% gets

Method	P	R	F1
LSTM-CRF	85.3	94.1	89.5
CRF	82.4	83.3	82.8
CNN-CRF	83.1	93.9	88.2
RNN-CRF	83.5	85.6	84.5
GRU-CRF	86.5	95.7	90.9
Bidirectional GRU-CRF	88.7	95.4	91.9
Bidirectional GRU+CNN-CRF	90.8	**96.2**	**93.4**
spaCy	**92.3**	90.7	91.5
SentCat (10%, 1st)	62.0	75.9	68.2
SentCat (10%, 2nd)	62.1	77.1	68.8
SentCat (10%, 3rd)	62.1	77.3	68.9
SentCat (70%, 1st)	51.5	83.7	63.8
SentCat (70%, 2nd)	51.9	83.4	63.9
SentCat (70%, 3rd)	52.0	83.7	64.2
CategoryClassifier (10%, 1st)	62.4	77.3	69.1
CategoryClassifier (10%, 2nd)	62.4	77.5	69.2
CategoryClassifier (10%, 3rd)	62.6	78.2	69.5
CategoryClassifier (70%, 1st)	52.4	85.0	64.8
CategoryClassifier (70%, 2nd)	52.3	84.2	64.5
CategoryClassifier (70%, 3rd)	52.1	84.6	64.5
BART	99.9	40.3	54.8
UTS-BART	100.0	41.0	55.47
T5	96.5	95.4	**95.97**
UTS-T5	100.0	97.7	**98.81**

Table 2.3: The Comparison of the Recent NER Methods with the Average Weighted Performance Metrics. P, R and F1 are the Represent Precision, Recall and F1 Score Respectively.

Method	P	R	F1
SentCat (10%, 1st)	90.3	70.3	79.1
SentCat (10%, 2nd)	90.5	70.7	79.4
SentCat (10%, 3rd)	90.6	70.8	79.5
SentCat (70%, 1st)	88.1	89.1	88.6
SentCat (70%, 2nd)	88.0	88.9	88.4
SentCat (70%, 3rd)	88.2	89.1	88.6
CategoryClassifier (10%, 1st)	83.5	58.4	68.7
CategoryClassifier (10%, 2nd)	90.9	70.9	79.6
CategoryClassifier (10%, 3rd)	**90.9**	71.4	79.9
CategoryClassifier (70%, 1st)	88.3	**90.0**	89.1
CategoryClassifier (70%, 2nd)	88.5	89.4	88.9
CategoryClassifier (70%, 3rd)	88.1	89.7	**88.9**

Table 2.4: The Average Weighted Performance Metrics of Our Method with SentCat and CategoryClassifier (BERT) for All Entity Types on the Full Annotation by the Full Dictionary. P, R, F1 are the Represent Precision, Recall and F1 Score Respectively.

the highest precision score in the dictionary size range between 10% and 90%, and the dictionary size 70% gets the highest recall score. The recall performance is higher than CRF method with full annotation, however, the precision performance is not as high as we expected. Thus, our method can create the high quality train corpus with the smaller dictionary than the full unique dictionary.

In Sec_col data, we compare the ten different approaches as follows:

(A) CRF

(B) BiDirectional LSTM

(C) BiDirectional LSTM with a CRF-classifier as an output layer

(D) BiDirectional LSTM with BiDirectional LSTM embeddings

(E) BiDirectional LSTM with BiDirectional LSTM embeddings and a CRF-classifier as an output layer

(F) BiDirectional LSTM with CNN embeddings

(G) BiDirectional LSTM with CNN embeddings and a CRF-classifier as an output layer

(H) spaCy (CNN)

(I) SentCat (number is % of the dictionary size)

(J) Our method with BERT (number is % of the dictionary size)

and the core layer in (B)-(G) is Bidirectional Long-Short Term Memory (BiLSTM) Neural Network (NN) [53, 63]. Models (C), (E) and (G) use a CRF-classifier as an output layer [63, 78, 86]. Models (D)-(G) also have special layers that build character embeddings [78, 86]. While models (D) and (E) use BiLSTM-layer to build character embeddings, models (F) and (G) use CNN-layer for the same purpose. The result data of (A)-(G) are from [142]. Table 2.5 shows the result.

In both SentCat and CategoryClassifier (BERT) cases, the dictionary size 30% and 70% performs better in many categories.

Since the original annotation accuracy has some doubt on Sec_col data as well, we add the additional annotation from the full unique dictionary if a sentence has missed annotation from the original. We call this new annotated test data as the fully annotated Test data on Sec_col dataset, and we evaluate our learned models with this fully annotated Test data. Table 2.6 shows the result with this fully annotated Test data. The performance of each case is better than the original annotation since the

Category		(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)-30	(I)-70	(J)-30	(J)-70
	P	85.4	28.9	61.2	79.1	**85.7**	72.8	79.2	67.4	56.4	50.1	55.0	47.1
Person	R	57.8	8.9	30	46.9	54.7	35	49.1	**66.3**	52.1	**59.1**	52.9	54.5
	F1	**68.9**	13.5	40.3	58.9	66.8	47.2	60.6	66.7	54.1	54.1	53.8	50.4
	P	**96.7**	90.2	88.1	92.7	92.9	95.5	94.6	79.8	78.2	79.4	80.2	80.1
Loc	R	81.9	39.4	53.5	70	82.3	52.5	73.5	**83.7**	**84.7**	**83.8**	**83.2**	**81.4**
	F1	88.6	54.8	66.6	79.8	**87.3**	67.6	82.7	81.7	81.3	81.5	81.6	80.7
	P	**85.9**	68.7	73	75.3	78.1	78.3	76.4	66.6	47.2	43.8	66.0	56.4
Org	R	65.5	30.3	38.3	62.1	**69.1**	48.6	67.5	60.4	45.6	53.0	47.4	52.0
	F1	**74.3**	42	50.2	68.1	73.3	59.9	71.6	63.2	46.3	47.9	55.2	54.1
	P	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Hacker	R	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	F1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	P	**87.5**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HackerGroup	R	14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	F1	24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	P	82.1	56.6	65.1	77.6	**85.8**	71.4	78.5	49.8	40.9	42.3	38.2	39.2
Program	R	**61.2**	29	40.4	51.3	60	57.1	58.2	55.6	32.6	41.2	43.9	49.3
	F1	70	38.4	49.9	61.8	**70.6**	63.4	66.6	52.5	36.2	41.6	40.8	43.4
	P	**65.4**	0.0	0.0	0.0	11.1	18.8	11.9	20.4	6.9	7.4	0.0	0.0
Device	R	**21.9**	0.0	0.0	0.0	0.8	2.5	0.8	3.5	6.9	6.8	0.0	0.0
	F1	**32.5**	0.0	0.0	0.0	1.5	4.3	1.3	5.8	6.7	6.8	0.0	0.0
	P	71.3	63	67.2	71.8	**77.4**	70.2	76.6	58.9	46.4	42.4	40.4	38.1
Tech	R	53.6	4.1	16.8	55.5	41.9	48	53.7	**66.9**	44.7	49.3	55.3	**59.3**
	F1	61.1	13.3	26.9	62.6	54.4	57	**63.1**	62.4	45.5	45.5	46.6	46.4
	P	68.5	0.0	0.0	0.0	37.5	3	23.8	50.0	59.0	**70.5**	0.0	0.0
Virus	R	**28.3**	0.0	0.0	0.0	5.1	0.4	3.8	17.9	12.3	8.0	0.0	0.0
	F1	**39.6**	0.0	0.0	0.0	9	0.7	6.6	24.5	19.8	14.2	0.0	0.0
	P	67.8	0.0	0.0	0.0	**71.4**	0.0	37.6	56.2	50.0	25.0	25.0	25.0
Event	R	**27.2**	0.0	0.0	0.0	5.9	0.0	7.2	9.6	2.9	0.7	0.9	0.4
	F1	**38.5**	0.0	0.0	0.0	10.9	0.0	12	16.0	5.5	1.4	1.7	0.7

Table 2.5: The Result of Test Data on Sec_col Data. P, R, F1 are the Represent Precision, Recall and F1 Score Respectively.

original annotation missed some keywords and annotated incorrectly. We got the best performance in Precision or Recall in four out of ten categories: "Person", "Location", "Tech", and "Virus". Thus, our method can perform better in some cases even if the dictionary size is smaller than original.

Category		(I)-40	(I)-70	(J)-30	(J)-70
Person	P	70.9	71.2	69.6	69.5
	R	**63.6**	**67.5**	53.9	**64.0**
	F1	67.0	**69.2**	60.6	66.4
Loc	P	75.3	79.1	80.5	80.5
	R	**85.3**	**83.2**	**82.7**	80.8
	F1	79.9	81.0	81.5	80.5
Org	P	47.8	47.8	66.0	60.1
	R	49.9	53.4	44.3	52.1
	F1	48.9	50.5	53.0	55.8
Hacker	P	0.0	0.0	0.0	0.0
	R	0.0	0.0	0.0	0.0
	F1	0.0	0.0	0.0	0.0
HackerGroup	P	0.0	0.0	0.0	0.0
	R	0.0	0.0	0.0	0.0
	F1	0.0	0.0	0.0	0.0
Program	P	45.8	44.3	39.8	41.0
	R	35.1	40.5	43.3	48.5
	F1	39.6	42.1	41.4	44.2
Device	P	13.3	14.1	0.0	0.0
	R	12.6	11.6	0.0	0.0
	F1	12.8	12.6	0.0	0.0
Tech	P	61.5	60.1	53.2	50.9
	R	54.5	**56.8**	**58.9**	**64.0**
	F1	57.8	58.3	55.9	56.7
Virus	P	**84.2**	**72.6**	0.0	0.0
	R	7.7	6.5	0.0	0.0
	F1	13.6	11.9	0.0	0.0
Event	P	8.3	25.0	25.0	25.0
	R	0.5	0.8	0.9	0.4
	F1	0.9	1.5	1.7	0.8

Table 2.6: The Result of the Fully Annotated Test Data on Sec_col Data. P, R, F1 are the Represent Precision, Recall and F1 Score Respectively.

2.5 Discussion

2.5.1 Analysis: Auto-labeled data

We checked the new keywords from the validation data in each model, and noticed that there are so many keywords that are easily identified in the specific category, but they are not annotated in the original corpus [25]. For instance, Some CVE IDs are special and unique such as CVE-2008-2565.1 and CVE-2009-4083.1, but the models learned from our annotated training data can detect them with the correct "CVE_ID" category. Since the original annotation uses the simple regular expression to detect CVE IDs, they missed these unique cases in the original annotation.

In addition, "Programming_Language" category has one keyword "JavaScript" originally, and this "JavaScript" is an ambiguous keywords that also belongs to "Function" and "Method" categories. We could find "C++" and "C#" as "Programming_Language" from the learned models but they are not annotated in the original dataset. Furthermore, some cyber attack related phrases ("Relevant_term" category phrases) such as "Cross-application scripting" and "Cross-zone scripting" are detected from our trained models but they are not annotated in the original dataset as well.

Moreover, many typos or extended version of application names are detected by our models. For instance, "OpenSSL", "VLC Media Player", "Enterprise Manager Grid Control" are in "Application" category of the original dictionary, and our trained models with both SentCat and CategoryClassifier train datasets can detect "openSSL", "VLC", "VLC 1.1.8", and "Enterprise Manager Grid Control EM Base Platform". These detected words are not listed in the original dictionary, and they are not annotated by the original work.

On the other hand, the learned models detects file paths such as "apps/admin/

handlers/" and "admin/action/" as "File" category, and file names with unnecessary characters such as "Admin/frmSite.aspx , (" and "admin/OptionsPostsList.php in". The issue of file paths has come from the frequent patterns of the file names. The frequent substrings of the file paths such as "/Admin/" and "apps/" are considered one of the features in the trained NER models and the phrases that contain the above patterns are extracted. The issue of the additional character is the problem of annotation or original text. For instance, the original text does not have the proper spacing between file name and other words or characters, the chunking the sentence in the learning NER model process affected the inaccurate chunking words from these sentences.

The original annotation has some issues such as using regular expressions and annotate wrong words and phrases with wrong categories like "7.50/7.53" as "File" category. However, our models with both SentCat and CategoryClassifier train datasets can annotate them as "Version" category.

In evaluation part, on average, about 2,794 entities are newly detected by a model with SentCat and about 438 of them are in the original keyword list, and about 2,960 entities are newly detected by a model with CategoryClassifier and about 422 of them are in the original dictionary. CategoryClassifier can detect more entities but SentCat can detect more entities in the original keyword list. The best ratio of the entities in the original dictionary is 10% of the original dictionary size case for both SentCat and CategoryClassifier, and 33.4% (778 out of 2,327 entities are in the original dictionary) with SentCat and 29.1% (615 out of 2,113 entities are in the original dictionary) CategoryClassifier respectively.

2.5.2 Analysis: Sec_col data

As the results of Sec_col data show in Table 2.5 and Table 2.6, our models with both SentCat and CategoryClassifier cannot learn and detect entities of "Hacker" and "Hacker_Group" categories. We checked the original dictionary and found the following points: "Hacker" and "Hacker_Group" categories have very small numbers of their entities (12 entities for "Hacker" and 37 entities for "Hacker_Group" respectively), and half of the hacker names in the original dictionary starts and ends with double quotes. The spaCy's system could not handle many cases of entities starting and ending with double quotes or parentheses. Thus, these original annotation issues and spaCy's issue may cause the low performance of "Hacker" and "Hacker_Group" categories.

In addition, spaCy has only one pre-trained model to support Russian, and the model covers multiple languages widely but not deeply. Since Sec_col data has the posts and forum publications from a Russian cybersecurity forum, it has many technical keywords in English and Russian. We suspect that the pre-trained model does not have the vector representation of many of these technical words and could not learn the semantic relations of the entities.

However, our models with both SentCat and CategoryClassifier can detect some useful but original annotation missed entities. For instance, "Taiwan", "Korea" and "Province of China" are all some geological words (locations) and categorized as "Loc" category by our models but the original annotation did not have these entities. In addition, the models detected some person's name and usernames as "Person" category such as "Carlos Almedia" and "Xaker45reg ***kov" but they are also not in the original annotated entities. The Sec_col data is manually annotated. Thus, we suspect there is some human mistakes during the annotation process and our models

can detect the missed entities.

After we carefully checked the original annotations, we found some potential and serious mistakes in the original annotation. For instance, "APT" (Advanced Persistent Threat) is only annotated as "Virus" category in the original annotation. An APT is a stealthy thread actor, so it should be annotated as "Hacker" or "Hacker_Group" category. In addition, under "Virus" category, there are so many non virus entities are annotated such as "DDoS", "0-day" and some CVE IDs. These inaccurate annotations may cause to the models' performance lower.

In evaluation part, on average, about 1,702 entities are newly detected by a model with SentCat and about 188 of them are in the original keyword list, and about 1,559 entities are newly detected by a model with CategoryClassifier and about 200 of them are in the original dictionary. SentCat can detect more entities but CategoryClassifier can detect more entities in the original keyword list. The best ratio of the entities in the original dictionary is 10% of the original dictionary size case for SentCat and 20% of the original dictionary size case for CategoryClassifier, and 31.0% (196 out of 632 entities are in the original dictionary) with SentCat and 31.6% (392 out of 1,242 entities are in the original dictionary) with CategoryClassifier respectively.

2.5.3 Discussion

Both of the experimental results with Auto-labeled data and Sec_col data show that our method can generate some high quality training data for a NER system with smaller dictionary size comparing to the original annotated datasets.

In Auto-labeled data, both SentCat and CategoryClassifier methods perform low precision score in the original annotation. However, when we use our annotation for the evaluation data as well, our method performs almost same performances of the most of the other NER methods. Since the original Auto-labeled data is annotated

by the automated labeling method, this result shows that our method can generate the higher quality training dataset for a NER system, and our method can annotate more accurately than the original automated method.

In addition, we train our UTS model with BART and T5 with various cybersecurity datasets. The UTS models are trained with nine datasets; Classification (four datasets), Event Detection (one dataset), Named Entity Recognition (three datasets), and Regression (one dataset) tasks respectively. The result shows that the UTS models improve the performance comparing to the models trained with only Auto-labeled dataset. The result supports that the UTS approach with task-based control codes has the potential to perform better than training individually.

In Sec_col data, both SentCat and CategoryClassifier methods perform highest precision or recall scores in some categories in the original annotation such as "Person", "Location", and "Tech". In addition, when we use our annotation for the evaluation data as well, our performance increased most of the categories. However, the performances of "Organization", "Hacker", "Hacker_Group", "Program", "Device" and "Event" categories are lower than the other methods. The "Virus" category got the highest precision score in SentCat method but 0 score in CategoryClassifier method. This means that CategoryClassifier may not annotate the "Virus" category keywords correctly than SentCat.

This preliminary experiment shows the advantage of our method but there is some space to improve. However, our method can annotate more accurately than the automated labeling method in Auto-labeled data, and our method is able to support multiple languages in Sec_col data. In addition, we found many issues from the original datasets and original annotations. We suspect that some low performance in some categories may be caused by these inaccurate keywords and these categories. The comparison of the original annotation and our method's annotation with the

carefully picked dictionary that has the keywords which the experts carefully evaluate and classify the right category will be needed to reinforce the benefit of our method. We also need to compare with the combinations of our method with other state-of-the-art NER systems to see some of the issues in the above can be solved with the different NER systems.

2.6 Conclusion

We propose a Human-Machine Interaction method for automatic annotation and corpus generation and **U**nified **T**ext-to-Text Cyber**S**ecurity (UTS) for multi-task model in the cybersecurity domain. We introduced SentCat to calculate the semantic similarity of the given keyword's category and the sentence that include the keyword to minimize the wrong annotation of ambiguous keywords. The experimental evaluation with three different corpora shows that our method performs well after iterating the process, and our method with SentCat can find more undiscovered keywords and useful training sentences that contain keywords. However, we find some issue with SentCat if the sentence is short or just noun phrase case. Thus, we introduced CategoryClassifier to calculate the semantic similarity of the given keyword's category and the sentence that includes the keyword to minimize the wrong annotation of ambiguous keywords. The initial experiment shows that CategoryClassifier performs slightly better than our previous measurement: SentCat. The experimental evaluation shows that our method performs well after iterating the process, and reached almost same performance of the state-of-the-art methods that use the fully annotated corpus with about $30 - 70\%$ of the keywords to annotate. In addition, the trained NER models with our method can detect many phrases that are not annotated originally. Furthermore, UTS models trained with various cybersecurity datasets improves the performance of the model that trained with a single dataset.

UTS-T5 model performs the highest F1 score in Auto-labeled dataset. Thus, our corpus generation method can generate the high quality training data with the small number of keywords comparing to the original full annotated data. Our method can help to create high quality training data for new cybersecurity domains if users need to create a new model to detect the phrases of new categories. Our UTS approach will increase the performance of individual task with a unified model trained with multiple datasets with various tasks. This is the first time to apply multi-task model in the cybersecurity field, and the result supports that the UTS approach works in this field.

For future work, we will extend the current keyword matching algorithm to find the noun phrase in the given sentence that includes the keyword since some keyword appears as a part of the noun phrase but the current method annotates the keyword itself and not the phrase. This change will increase the quality of annotation. Then, we will extend the NER model from spaCy to the other state-of-the-art NER models [18, 7, 42], and evaluate the difference of the performance by each NER method. Finally, we will apply this trained NER model for other cybersecurity related tasks such as detecting new malware names, analysis of malware families, and APT Groups supported by more detailed information on such actors.

Chapter 3

SOCIAL STRUCTURE CONSTRUCTION FROM THE FORUMS USING INTERACTION COHERENCE

3.1 Introduction

Extracting social structure from forums and communities is an important task, especially in the cybersecurity field. Researchers have used Social Network Analysis (SNA) to identify key individuals within the hackers forums and communities in the Deepweb and Darkweb [51, 102, 4, 163, 45, 132, 89]. To build the social network, the member's interaction must be taken into consideration [48]. In the forum, members' activity is followed according to its participation on the forum [124]. In addition, SNA is used for many applications and methods as a part of their features to predict cyber threats and enterprise cyber incidents from Deepweb and DarkWeb forums [10, 136].

There are several structured forums and communities such as Reddit [1] and Stack Exchange [2] . Reddit is a platform for discussions on a variety of topics on the web. There are many threads under a specific topic, and the responses are shown in tree structure. Stack Exchange is a network of question-and-answer (Q&A) websites on topics in diverse fields, each site covering a specific topic. Each thread has a tree structure to see the replies of the posted question. However, most of the communities and forums in the Deepweb and Darkweb are unstructured, and it is hard to build the social structure from unstructured threads. There are two user network (social network) representations according to the reply schema of members [82]: Creator-

[1] https://www.reddit.com/

[2] https://stackexchange.com/

oriented Network and Last Reply-oriented Network. Many of the SNA methods are using one of the representations, or extended representation of them [132, 10, 136, 89].

The recent work [56] proposed the approach to use a neural network based model to analyze the posts to judge whether the post is useful for the thread. Neural network based models have outperformed existing classifiers in many text classification tasks. They have been widely adopted as they induce useful features on their own, given sufficient data. All posts in a thread are similar to the original post to an extent. Helpful posts are not easily identified through similarity as a single source solution. Their recurrent neural network based architecture is to model the relevance of a post regarding the initial post that starts the thread, and the novelty it brings to the discussion, compared to the previous posts in the thread.

BERT (Bidirectional Encoder Representations from Transformer) [41] is a neural network-based technique for Natural Language Processing (NLP) pre-training. BERT helps better understand the nuances and context of words in searches and better match those queries with more relevant results. BERT pre-trains the two tasks: Masked Language Modeling (LM), and Next Sentence Prediction (NSP) with raw corpus. The second task is NSP, where BERT learns to model relationships between sentences. In the training process, the model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original document.

Inspired by all these previous works, we propose a new Social Structure Construction method. Our new method uses the Next Paragraph Prediction that we introduce the extension of BERT's Next Sentence Prediction since the posts in a thread usually have more than one sentence and the number of responses of a post is not only one but also multiple. We compare Next Paragraph Prediction model with traditional network models. The result shows our Next Paragraph Prediction model performs

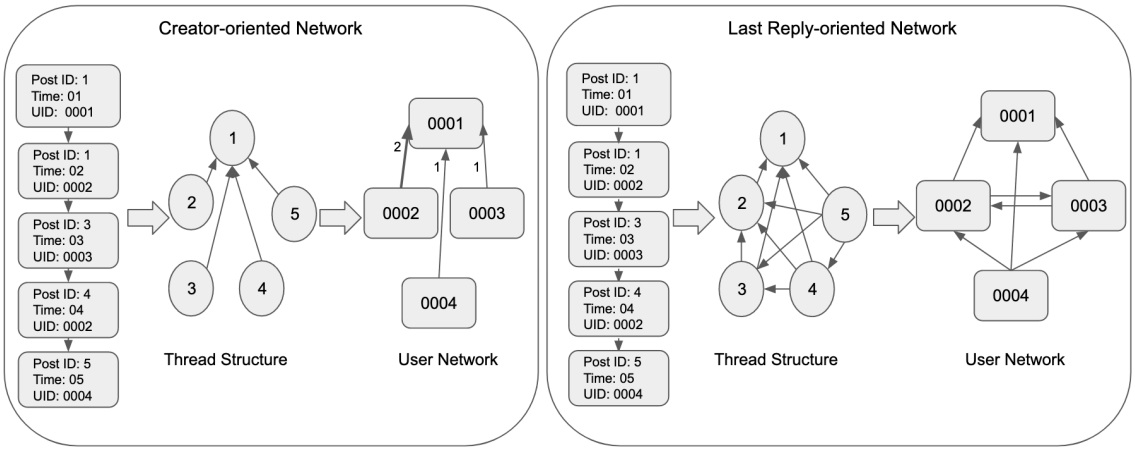

Figure 3.1: Two Different Networks Models: Creator-oriented Network and Last Reply-oriented Network to Represent a Given Unstructured Thread Interaction in a Forum.

on average over 80% in the third iteration after fine-tuning on unbalanced dataset which is the largest training data. This initial evaluation shows that our approach can construct very accurate thread structures from unstructured threads in a forum, and build an accurate social structures from the thread structures. We also perform the ablation experiment over the difference between Next Sentence Prediction and Next Paragraph Prediction. The result shows that Next Paragraph Prediction performs better when the posts contain multiple sentences, and it can consider the semantic meanings of the whole posts.

In addition, following procedural texts written in natural languages is challenging. We must read the whole text to identify the relevant information or identify the instruction flows to complete a task, which is prone to failures. If such texts are structured, we can readily visualize instruction-flows, reason or infer a particular step, or even build automated systems to help novice agents achieve a goal. However, this structure recovery task is a challenge because of such texts' diverse nature. This paper [112] proposes to identify relevant information from such texts and generate

information flows between sentences.

The main contributions of this paper include:

- We propose new method that predicts the possible direct response of a post to build the social structure of a thread. This method can build a social structure from unstructured thread.

- We introduce our new Next Phrase Prediction approach which is extended from BERT's Next Sentence Prediction. Next Phrase Prediction returns true when a response post is the direct reply of a post.

- We empirically perform experiments on ten different topics under Reddit's cybersecurity field. The experimental results demonstrate our method performs better than the traditional approaches.

- We also compared the performance between BERT's Next Sentence Prediction and our Next Phrase Prediction. If the response is not a single sentence, our method performs better since the replies can be considered thematically related. In other words, Next Phrase Prediction can use the information that is more loosely related (e.g. question and response) than two subsequent sentences.

The rest of the paper is organized as follows: we introduce several terms and applications related to extracting social structures and BERT in Section 2, the general framework of our proposed method including Next Paragraph Prediction in Section 3, and finally the experimental evaluations in Section 4.

3.2 Related Works

3.2.1 Extracting Social Structure and Network

A Social Network (SN) is defined as the representation of communication networks where the nodes are people and the edges (arcs) correspond to the relationships between. Social Network Analysis (SNA) [156] helps to understand the relationships in a given community through analyzing its graph representation. Users in the community are seen as nodes and relations among users are seen as arcs. Through this way, there are several techniques have been researched such as extract important (key) members [105, 89], classify users according to his or her relevance within the community [121], and discovering and describing resulting sub-communities [77]. However, all these approaches leave aside the meaning of relationships among users. Therefore, analysis based only on reply of posts to measure relationships' strangeness or weakness is not a good indicator.

To build the social network, the members' interaction must be considered. In general, the activities of members is followed according to its participation on the forum such as posting or responding in the threads on the forum. There are two network representations introduced [82]:

- Creator-oriented Network: When a member creates a thread, every reply will be related to him or her. This network representation is the less dense network (density is measured in terms of the number of arcs that the network has).

- Last Reply-oriented Network: Every reply of a thread is assumed to be a response to the last post. This network representation has a medium density.

In Figure 3.1, these two approaches of network conversion of an unstructured thread of a forum are presented. The arcs represent members' replies and nodes

represent the authors of the posts. In Creator-oriented network approach, the weight of arcs in User Network (social network) will be a counter of how many times a given member replies to posts. The two approaches create very different thread structures and user networks. The Last Reply-oriented Network is widely used for the social network analysis in the recent works [124, 10, 89, 136].

Since these two traditional network conversion approaches are based on preliminary assumptions, we suspect that the social structures of the networks are not accurate representations of social structure. Thus, we consider the users' interaction in the thread to reconstruct the thread structures from unstructured threads, then build the social structure based on the thread structures. The recent work predicted helpful posts in the forums [56] uses a neural network based model that determines whether the post is useful or not. In addition, BERT has Next Sentence Prediction to judge that a sentence is the next sentence of a given sentence. We assume that BERT's Next Sentence Prediction can extend to predict the response post from the previous post.

3.2.2 BERT

BERT (Bidirectional Encoder Representations from Transformer) [41] has two steps: pre-training with large raw corpus, and fine-tuning the model for each task.

BERT is based on Transformer [148], which can catch the long distance dependency relations, because it is based on self-attention, and does not use RNN or CNN.

The input for BERT is a sentence, pair of sentences, or document, and it represents the sequence of tokens in each case. Each token is the summation of token embedding, segment embedding, and position embedding.

Each word is divided into sub-words, and the non-head part in the subwords will be assigned "##". For instance, "playing" is divided into "play" and "##ing" as

subwords. If the input is two sentences, segment embedding gets the first sentence token as sentence A embedding, and the second sentence token as sentence B embedding (put "[SEP]" token between two sentences). In addition, the location of each token is learned as position embedding. The head of each sentence is marked with the "[CLS]" token. In the document classification task or two sentences classification task, the final layer of embedding of the token is the representation of the sentence or the two-sentences-set.

BERT pre-trains the following two tasks with the raw corpus: Task 1: Masked Language Modeling (LM), and Task 2: Next Sentence Prediction.

BERT sets Masked LM as a task, it can use Transformer in bidirection which read the text input sequentially both left-to-right and right-to-left. For instance, lets assume the following sentence.

1. the men went to the store

The randomly selected word "'went' from the above sentence is masked and the following sentence is created.

2. the men [MASK] to the store

Then, this sentence is applied Transformer and the model is trained to predict [MASK] part's token correctly.

It is important to capture the relationship between two sentences in the tasks such as Question Answering and Textual Entailment Recognition. Then, Next Sentence Prediction task pre-trains the model. The model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original document. During training, 50% of the inputs are a pair in which the second sentence is the subsequent sentence in the original document (The following (3)), while in the other 50% a random sentence from the corpus is chosen as the

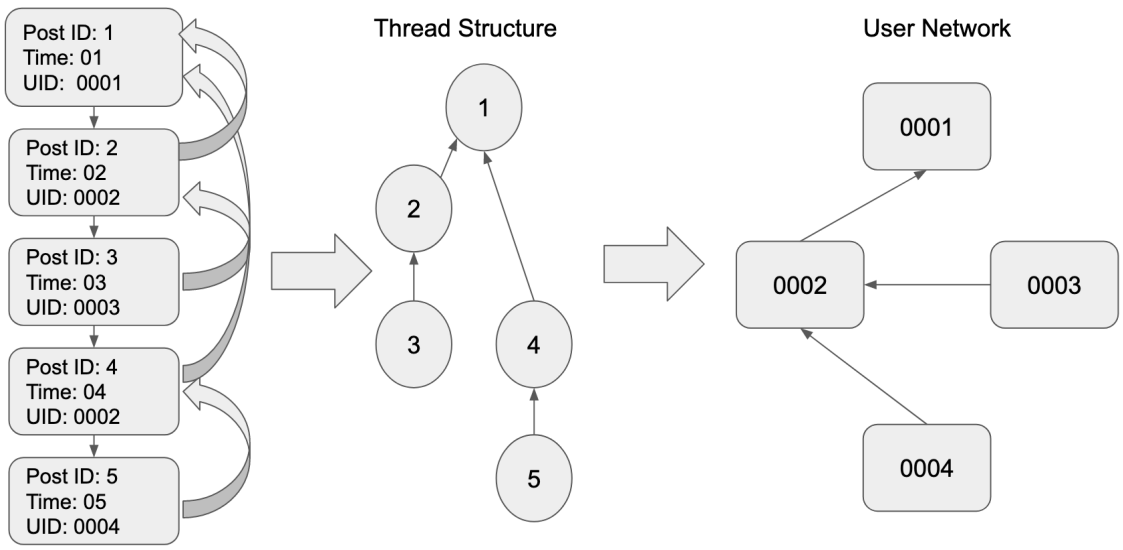

Figure 3.2: Sample Thread Structure and Its User Network.

second sentence (The following (4)). The assumption is that the random sentence will be disconnected from the first sentence.

3. [CLS] the man went to the [MASK] [SEP] he bought a gallon of milk [SEP]

4. [CLS] the man [MASK] to the store [SEP] penguin [MASK] are flight ##less birds [SEP]

While only adding a small layer to the core model, BERT can be used for a wide variety of language tasks such as Classification tasks, Question Answering tasks (e.g. SQuAD), and Named Entity Recognition (NER) tasks. For instance, lets consider sentence pair classification task or sentence classification task. This task is to calculate the probability of each class through $P = \text{softmax}(CW^T)$ where C is the final layer's embedding corresponding to [CLS] and additional parameter $W \in \mathbb{R}^{K \times H}$ (K is the number of classes).

3.3 Proposed Method

Since both of the traditional networks do not consider the user interaction of the thread correctly if the forum is unstructured, the social networks do not represent the users' interaction accurately. Thus, the new approach to build the thread structures from the unstructured forum to generate more accurate social network is required. To achieve this goal, it is promising to determine user interaction more clearly through identifying who responds to whose post. For instance, Figure 3.2 shows, if the relationship between posts is figured out by understanding the likelihood of each post, the thread structure is constructed even if the thread is unstructured, and the accurate user network is constructed for social network analysis. We consider each post in the forum's thread as one paragraph, and extend BERT to predict direct response of the post or reply as next paragraph.

We propose the following three methods; Next Paragraph Prediction (NPP), Flow Structure, and Next Paragraph Prediction with Instructional Prompting (NPP-IP).

3.3.1 Next Paragraph Prediction

We introduce Next Paragraph Prediction that returns true if a response post is a direct response of the previous post in a thread using BERT's Next Sentence Prediction idea. To extend Next Sentence Prediction in BERT to Next Paragraph Prediction, we need to consider the following differences between sentence and paragraph.

- The next sentence is usually unique. However, the next paragraph (in this case, a responding post to the previous post) may be not unique and multiple responses may exist against a post. Although, in this approach, the replies can be considered thematically related, it could be argued that they are more

Algorithm 5 NextParagraphPredictionTraining

Input: Structured threads in a forum *Forum*

Output: Fine-tuned model for Next Paragraph Prediction

TrainTripletList = []
for all *Thread* ∈ *Forum* **do**
  parentDict = {}
  postList =list of all posts in *Thread*
  posCount = 0 # count the positive example number per thread
  for all *post* ∈ *postList* **do**
    if *parentPost* of *post* is not ROOT **then**
      parentDict[*post*] = *parentPost*
      TrainTripletList add (True, *parentPost*, *post*)
      posCount+ = 1
    end if
  end for
  for $i = 0$; $i <$ *posCount*; $i++$ **do**
    Randomly picks *post1* and *post2* from *postList* where *post1* ≠ *parentDict*[*post2*] and *post1* ≠ *post2*
    TrainTripletList add (False, *post1*, *post2*)
  end for
end for
Fine-tuning the BERT model with *TrainTripletList* for training the model for Next Paragraph Prediction

loosely related (e.g. question and response) than two subsequent sentences. In this regard, the case at hand is semantically closer to two paragraphs.

- Next Sentence Prediction creates same number of negative case from the positive case by randomly picking the next sentence from the training corpus. However, this approach may pick another positive paragraph as a negative sample.

Considering the above differences, the training process of Next Paragraph Prediction is shown in Algorithm 5. NextParagraphPredictionTraining algorithm generates the training corpus from the given structured forum data and using the labeled pairs of paragraphs are used for fine-tuning BERT model for Next Paragraph Prediction. The examples of the positive paragraphs pair and negative paragraphs pair are shown in (5) and (6) respectively.

5. [CLS] Just bought a subscription . Thank you for the use ##ful service . We find it very value ##able for aware ##ness [MASK] [SEP] Thank you for the support and kind words [SEP]

6. [CLS] Ok . [MASK] . [SEP] I really [MASK] not know what I am looking honestly . [SEP]

3.3.2 Flow Structure

We map each post of a thread as a node in a graph, and the direct replies as edges. The task is then simplified into an edge prediction task: Given a pair of nodes, find if there is an edge between them. We learn feature representations of nodes using language models like BERT/RoBERTa [43, 83]. Then, to make the nodes aware of their neighboring sentences, we use Graph Neural Network (GNN) to update the node representations. We check for the edge between every pair of nodes in a graph and

reduce the task to a binary classification during inference. This formulation enables us to predict any kind of structure from a document.

Our goal is to find paths or traces of actions or information between posts. This needs an understanding of each post's interconnection. Hence, we modeled the problem into an edge prediction task in a graph using GNNs. We represent each post as a node and directed edges as information flows. Since this is procedural text (unidirectional nature) of instructions, we consider only the directed edges from one sentence S_n to any of its next sentences S_{n+i}. The node representations are learned using language models (LM) and GNNs.

Each threat (D_i) is converted into a series of posts (S_j) where n is the number of valid posts in a threat.

$$D_i = \{S_0, S_1, S_2 ... S_{n-1}\}$$

Document to Graph Representation

A graph ($G = (V, E)$) is formally represented as a set of nodes ($V = \{v_0, v_1, ..\}$) connected by edges ($E = \{e_0, e_1, ..\}$ where $e_i = \{v_m, v_n\}$). We consider the posts (S_j) of any thread (D_i) as nodes of a directed graph (G_i). We experiment with two graph structure types for learning better node representation using GNN. First, we form local windows (W_N, where $N = 3, 4, 5, all$ posts) for each post and allow the model to learn from all of the previous post in that window.

We form the document graph by connecting each post with every other post in that window, with directed edges only from S_i to S_j where $i < j$. We do this since procedural languages are directional. We call this configuration *Semi-Complete*. Second, we consider connecting the nodes linearly where every S_i is connected to S_{i+1} except the last node. We call this *Linear* setting.

We use LMs like BERT and RoBERTa to generate initial post representations. For each sentence (S_i), we extract the pooled post representation (CLS_{S_i}) of contextual BERT/RoBERTa embeddings (h_{S_i}). We use CLS_{S_i} as node features for the graph (G_i).

$$h_{S_i} = BERT\,(\,[CLS]s_0s_1...s_{n-1}\,[SEP]\,)$$

Neighbor Aware Node Feature Learning

Since the LM post vectors are generated individually for each post in the threat, they are not aware of other local post. So, through the *semi-complete* graph connection, the model can learn a global understanding of the thread. However, the *linear* connection helps it learn better node representation conditioned selectively on its predecessor. We call the connected nodes as the neighbor nodes. We use Graph Convolutional Network (GCN) [75] and Graph Attention Network (GAT) [149] to aggregate the neighbor information for each node following the generic graph learning function (3.1)

$$\mathbf{H}^{l+1} = f(\mathbf{H}^l, \mathbf{A}) \tag{3.1}$$

where $\mathbf{A}$ is the adjacency matrix of the graph, $\mathbf{H}^l$ and $\mathbf{H}^{(l+1)}$ are the node representations at lth and $(l+1)$th layer of the network and f is the message aggregation function. In GCN, each node i, aggregates the representations of all of its neighbors $N(i)$ based on $\mathbf{A}$ and itself at layer l and computes the enriched representation $\mathbf{h}_i^{l+1}$ based on the weight matrix $\mathbf{\Theta}$ of the layer normalized by degrees of source $d(i)$ and its connected node $d(j)$ as per (3.2). In GAT, messages are aggregated based on multi-headed attention weights (α) learned from the neighbor node representations

$\mathbf{h}_j^l$ following (3.3).

$$\mathbf{h}_i^{l+1} = \mathbf{\Theta} \sum_{j \in N(i) \cup \{i\}} \frac{1}{\sqrt{d(i)d(j)}} \mathbf{h}_j^l \tag{3.2}$$

$$\mathbf{h}_i^{l+1} = \alpha_{ii} \mathbf{\Theta} \mathbf{h}_i^l \quad + \sum_{j \in N(i)} \alpha_{ij} \mathbf{\Theta} \mathbf{h}_j^l \tag{3.3}$$

Projection

We concatenate the neighbor aware node representations of each pair of nodes ($\mathbf{h}_i$;$\mathbf{h}_j$) from a graph and pass it through two projection layers with a GELU [57] non-linearity in between. We use the same non-linearity functions used by the BERT layers for consistency. We steadily decrease the parameters of each projection layer by half. During testing, given a thread, we are unaware of which two posts are connected. So, we compare each pair of nodes. This leads to an unbalanced number of existing (1) and non-existing (0) edge labels. Hence, we use weighted cross-entropy loss function as in equation (3.4) and (3.5), where L is the weighted cross-entropy loss, w_c is the weight for class c, i is the data in each mini-batch.

$$L(x, c) = w_c \Big(- x_c + log \Big(\sum_j exp(x_j) \Big) \Big) \tag{3.4}$$

$$L = \frac{\sum_{i=1}^{N} L(i, c_i)}{\sum_{i=1}^{N} w_{c_i}} \tag{3.5}$$

3.3.3 Training and Inference

Our training data comprises a set of posts and the connections as an adjacency matrix for each document. Batching is done based on the number of graphs. GCN/GAT updates the post representations. A pair of node representations are assigned a label of 1 if there is an edge between them; otherwise, we assign them 0. Thus, we model it as a binary classification task as in equation (3.6) where f is the projection function, g is the softmax function, and y is the binary class output. Depending on the weighted

cross-entropy loss, the node representations get updated after each epoch. During inference, the model generates node representations of each post in a test thread, and we predict whether an edge exists between any two nodes in a given thread graph.

$$y_c = \arg\max_k g(f(\mathbf{h}_i; \mathbf{h}_j), k) \quad c \in \{0, 1\} \tag{3.6}$$

3.3.4 NPP with Instructional Prompts

Our proposed NPP-IP model is based on infusing the original dataset with specific task instructions using an instruction prompting function. Formally, the instruction prompting function $f_{prompt}(\cdot)$ is defined as

$$f_{prompt}(x) = I||x, \tag{3.7}$$

where —— represents concatenation of instruction prompt I with training sample x. Instruction prompt I is formally defined as:

Task Description:

ΨYou are given two posts and you need to generate True if they are the direct reply relation, otherwise generate False.

Ψ**Positive Example:**

post1: Windows Defender Gets a New Name: Microsoft Defender

post2: Bring back MSE and its ui even logo looks cool...

Ψ

output: True

Negative Example:

post1: Windows Defender Gets a New Name: Microsoft Defender

post2: Title says it

output: False"

Training sample x is formally defined as

$$x = \text{Post } k \text{ —— [sep] —— Post } k + i, \tag{3.8}$$

which represents a pair of concatenated posts at index k and $k+i$ with a separation key [*sep*], such that $i \neq 0$.

The NPP-IP model leverages five framing techniques defined in [100] for framing the instruction prompting information I. First, the **Use Low Level Patterns** technique is accomplished by providing a simple task descriptor to correctly output a value of True or False if a reply relationship exists between posts without including any cybersecurity jargon. Second, **Itemized Instructions** are provided via the positive and negative examples with the corresponding output in bulleted list format for thread structure prediction. The positive and negative examples also fulfill the **Break It Down** technique by defining simpler sub-tasks corresponding to identifying negative and positive examples. This is also where cybersecurity information is introduced into the instructional prompt. Next, **Enforce Constraints** is accomplished by constraining the examples to their respective outputs of True or False. Lastly, the **Specialize Instructions** technique is accomplished by specifically stating the expected output in both task description and examples.

Figure 3.3 shows the BERT-based neural network structure used by an NPP model as well as the resultant NPP-IP model after introducing instructional prompting information. The original dataset gives two posts as its input, where the label space is defined as {True, False}, defining whether posts share a direct response relation or not. Including instructional prompting provides critical task information for both positive and negative cases, which are then used in the embedding and subsequent prediction task during training.

3.3.5 Social Structure Construction

Using the fine-tuned model for NPP, Flow Structure, or NPP-IP as a thread prediction model, Social Structure Construction algorithm builds the social structure

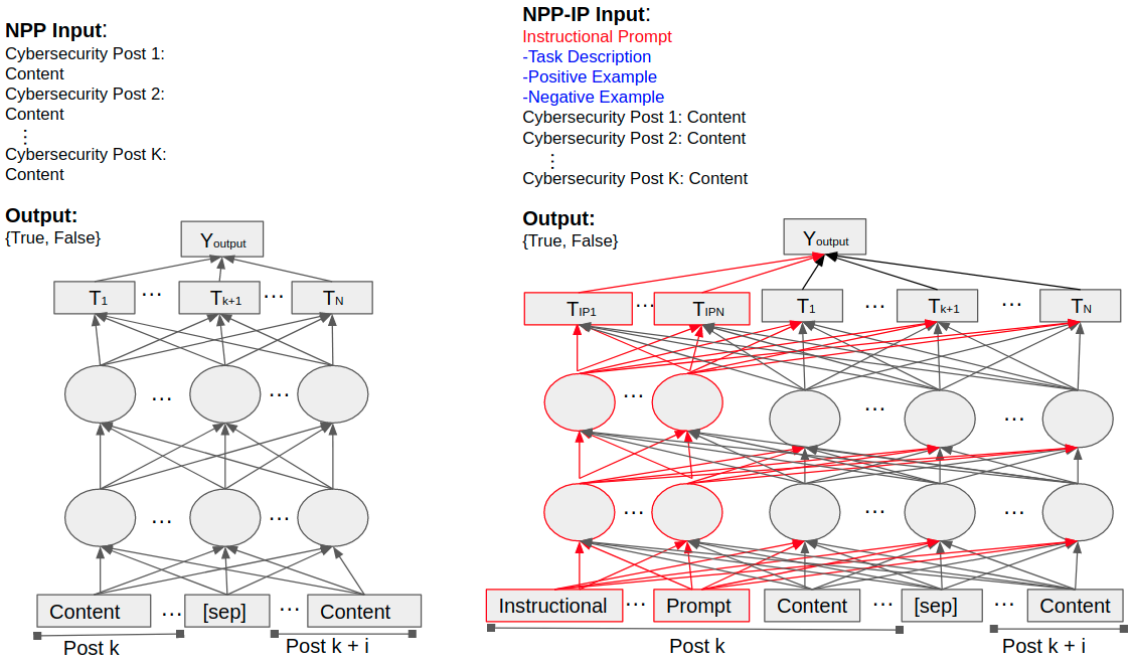

Figure 3.3: The Original NPP Model (Left) Combines a Pair of Posts to Predict Whether One Post is a Response to the Other. Our NPP-IP Model (Right) Incorporates Instructional Prompt Information into the NPP Structure Allowing for Task Information to be Leveraged.

of unstructured forum to generate the social network of users therein. Algorithm 6 shows the process to generate the social structure of the given unstructured forum. If one of our thread structure prediction models *TPM* in Algorithm 6) returns "true" for given two individual posts from same thread, the thread structure puts the edge between the two posts' nodes.

Once the social structure of unstructured forum is built, the social network (user network) is easily extracted for the social network (user network) from the social structure for Social Network Analysis. This approach will build the accurate social network for unstructured forums compared to the traditional approaches: Creator-oriented Network and Last Reply-oriented Network.

Algorithm 6 SocialStructureConstruction

Input: Unstructured threads in a forum *Forum*, Thread Prediction Model *TPM*

Output: *SocialStructure*

ForumStructure
for all *Thread* $\in$ *Forum* **do**
  ThreadStructure
  postList =list of all posts in *Thread*
  for $1 \leq i \leq |postList|$ **do**
    for $1 \leq j \leq |postList|$ **do**
      if $i \neq j$ and *postList*[j] posted after *postList*[i] **then**
        post1 = *postList*[i]
        post2 = *postList*[j]
        if *TPM*(*post1*, *post2*) returns True **then**
          ThreadStructure add the edge from *post2* to *post1*
        end if
      end if
    end for
  end for
  ThreadStructure is added to *ForumStructure*
end for
Generate *SocialStructure* of the *Forum* based on *ForumStructure*

3.4 Evaluation

We evaluated our proposed method with ten different Reddit topics related to the cybersecurity field, and compared with the traditional approaches: Creator-oriented Network and Last Reply-oriented Network. The evaluation performance is measured

to return the accuracy of the prediction of the correct pairs of paragraphs (post and reply) in the structured threads. We generate the training corpus for fine-tuning the Next Paragraph Prediction model and we use the corpus for the evaluation as well.

3.4.1 Data

Reddit is a popular platform for discussing a wide-variety of topics on the web. This discussion platform presents each thread in the forum of a tree structure, so that it is clear to see the users' interactions such as who replies to whose post and when the response is posted. We picked the following ten topics from "cybersecurity" field in Reddit and extracted the threads of these topics: "cyber_security", "AskNetsec", "ComputerSecurity", "cyberpunk", "cybersecurity", "Hacking", "Malware", "Malwarebytes", and "security".

Each post or response under a forum in a topic is considered a paragraph, and the positive pair of the paragraphs is created if a paragraph's ID appears in the response's children list. The statistic of our collected ten Reddit topics is shown in Table 3.1.

Our proposed models were also evaluated using 20 hacker forum threads from three English hacker forums annotated by human experts, which is referred to as the "Hacker Forums" dataset. The forum thread data is from CYR3CON [3] . The average posts per thread is 15.4. Four cybersecurity experts checked posts in each thread, and annotated a relation of two posts in a thread which the two posts are direct response relations or not. The site names and usernames are anonymized. The topic, thread, and post information from the Hacker Forums dataset are provided at Table 3.2.

[3]https://www.cyr3con.ai

Topic Name	TH	Posts
cyber_security	8	48
AskNetsec	14	338
ComputerSecurity	12	110
cyberpunk	11	176
cybersecurity	11	158
Hacking	12	370
Hacking_Tutorial	12	110
Malware	9	82
Malwarebytes	8	72
security	8	184

Table 3.1: The Statistics of the Evaluated Data from Reddit Ten Topics. TH Means the Number of Threads in Each Topic, Posts Means the Number of Posts

Forum #	TH	Posts
Forum1	7	169
Forum2	7	80
Forum3	6	58

Table 3.2: The Hacker Forums Dataset Consisted of 20 Threads from Three Hacker Forums. "TH" is Defined as the Number of Threads in Each Topic While "Posts" is Defined as the Number of Posts Across the Different Threads.

3.4.2 Metrics and Task

Our proposed NPP, NPP-IP, and FS methods were evaluated against several different methods for thread structure prediction using cybersecurity related posts. Two language models, BERT (BE) and RoBERTa (RB), were explored when training the NPP and proposed NPP-IP models, where -B and -L represent base and large models for each LM respectively.

We compared performance with well known methods, Creator-Oriented Network (CO) and Last Reply-Oriented Network (LR) using Precision (P), Recall (R), and F1 score (F1) metrics reported in Tables 3.3 and 3.4 for Reddit and Hacker Forums datasets, respectively.

3.4.3 Experimental Results

Tables 3.3 and 3.4 show the results of Reddit and Hacker Forums datasets, respectively. NPP-IP and FS methods outperformed all other methods for thread structure prediction across all. Most of the cases, NPP-IP improves the performance from NPP. FS method reached the highest performance in Reddit dataset, however, it did not perform well in hacker forum dataset.

3.5 Discussion

Our NPP approach out performed to the existing methods, CO and LR in both Reddit and Hacker forums datasets. This means that our approach can consider the context of each posts and determine the user interaction to predict the thread structure. However, the highest F1 scores of NPP in Reddit and Hacker Forums datasets are 0.4 range.

We extended NPP training process with instructional prompts (NPP-IP). As the

Method	Model	P	R	F1
CO	-	0.00	1.00	0.01
LR	-	0.72	0.12	0.20
NPP	BE-B	0.42	0.46	0.44
	BE-L	0.36	0.51	0.42
	RB-B	0.59	0.33	0.43
	RB-L	0.41	0.57	0.48
NPP-IP	BE-B	0.48	0.46	**0.47**
	BE-L	0.64	0.41	**0.50**
	RB-B	0.62	0.43	**0.51**
	RB-L	0.39	0.56	0.46
FS	BE-GCN	0.36	0.79	0.50
	RB-GCN	0.60	0.47	**0.53**
	BE-GAT	0.65	0.40	0.49
	RB-GAT	0.45	0.66	**0.53**

Table 3.3: Results from the Reddit Test Data Show that the NPP-IP and FS Methods Outperformed All Other Methods for Thread Structure Prediction Across All but One of the Different BERT Language Models Analyzed.

results show that the NPP-IP improved the performance than original NPP in most of the cases. This is the first time to apply instructional prompts to the cybersecurity domain, and the results prove that instructional prompts work in the cybersecurity domain as well. The F1 scores of NPP-IP in both Reddit and Hacker Forums datasets are 4-50 times better than the existing CO and LR methods.

Our FS approach performed best in Reddit dataset, however, it did not perform well in the Hacker Forums dataset. NPP and NPP-IP are determining the pairs of posts to find the given posts are direct reply or not. On the other hand, FS

Method	Model	P	R	F1
CO	-	0.03	0.44	0.05
LR	-	0.50	0.06	0.11
NPP	BE-B	0.39	0.43	0.41
	BE-L	0.84	0.27	0.41
	RB-B	0.50	0.35	0.41
	RB-L	1.00	0.28	0.44
NPP-IP	BE-B	0.80	0.38	**0.52**
	BE-L	0.61	0.44	**0.51**
	RB-B	0.74	0.31	0.44
	RB-L	0.47	0.35	0.41
FS	BE-GCN	0.31	0.86	0.46
	RB-GCN	0.31	1.00	0.47
	BE-GAT	0.32	0.79	0.46
	RB-GAT	0.32	0.73	0.45

Table 3.4: Results from Each of the Anonymous Hacker Forums Demonstrated that the NPP-IP Outperformed All Other Models. The NPP, NPP-IP, FS Models were both Trained with Reditt Data Further Demonstrating NPP-IP Inference Performance Robustness on Unrelated Cyber Forums.

approach receive all posts in a thread and predict the thread structure itself. The Hacker Forums dataset has real hacker forums posts that have many grammatical mistakes and mixed with multiple languages and codes. These noise may affected the performance.

A common issue between NPP, NPP-IP, and FS approaches is predicting some non direct response posts as direct response. This is because many reply posts quoted the previous posts. We suspect that our models may catch these quoted part to judge the direct reply or not. There are several error cases that are not easy to solve.

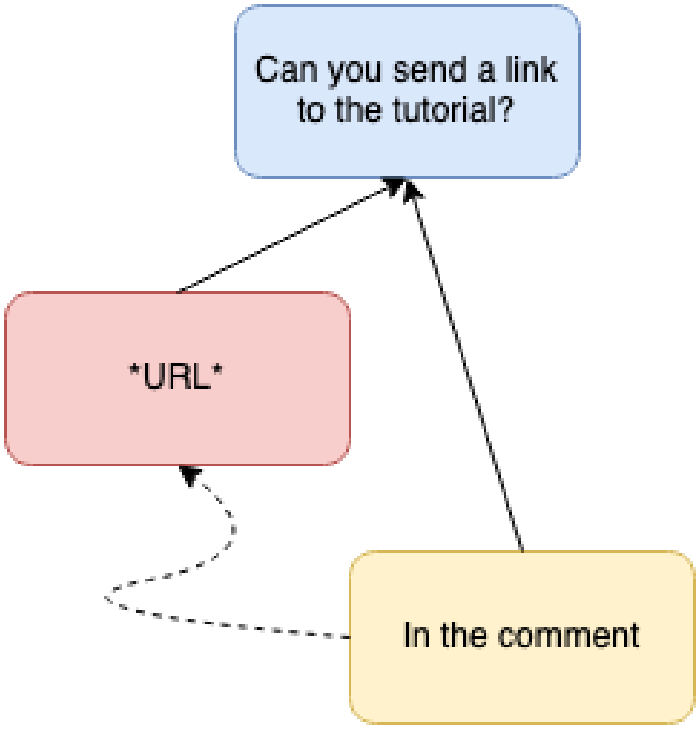

Figure 3.4: An Interesting Case in Reddit Dataset. Since the Actual URL is Harmful Site URL, We Replaced It as *URL*.

Figure 3.4 shows an interesting case we found in the Reddit dataset. For the question post "Can you send a link to the tutorial?", a user responded "*URL*" and "in the comment". In the ground truth, "in the comment" is the response of "*URL*", however, both of our models predicted "in the comment" is the response of "Can you send a link to the tutorial?". We think that "in the comment" reinforces the post "*URL*" and also answers the original question. Since the ground truth is based on the thread tree structure, it only has one interaction even if it can interact with multiple posts or users. However, due to the tree structure in Reddit, the ground truth from the subreddit structure is assigned to only one of them. We found some cases that our methods predicted a post replied to multiple posts, and only one of them is correct as we mentioned before. Thus, these cases may decrease the performance of our methods.

In the Hacker Forums dataset, two hacker forums have a feature to quote the

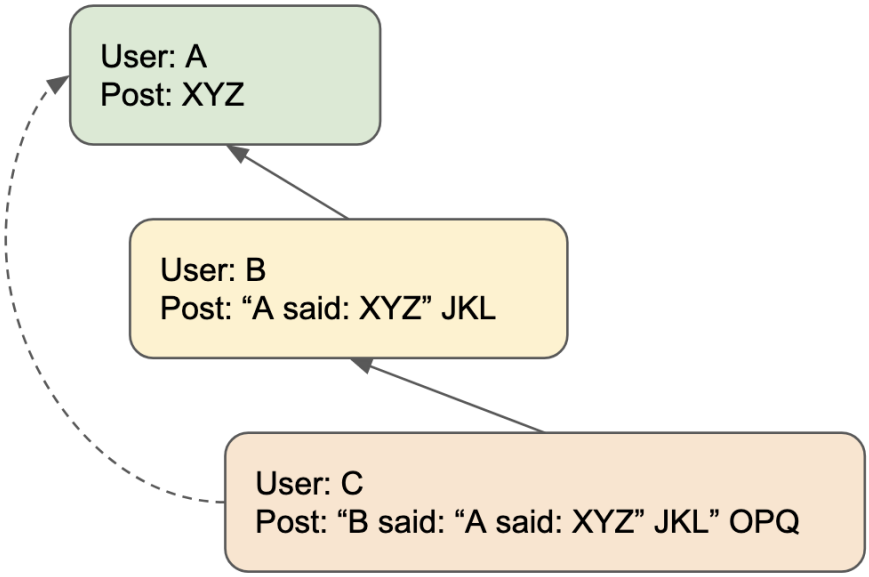

Figure 3.5: An Interesting Case in Hacker Forums Dataset.

referencing post in the same thread. However, the quote feature catches not only the referencing post but also the referencing post's referencing post if it has. Figure 3.5 shows the example of this case. User C replied (referenced) User B post, however, the User B referenced User A post as well. Then, both NPP and NPP-IP models predicted the pair of User C post and User A post are direct reply relationship since User C post contains User A post content through referencing User B post. We observed many this false positive cases, and this type of error effected some performance in the Hacker Forums dataset. Some pre-process to remove reference's reference post content will be needed to solve this issue.

3.6 Conclusion

Predicting thread structures within cybersecurity forums is a crucial component in defining key social networks used to identify prominent users who provide useful information. Identifying these users can facilitate prediction and prevention of future

cyber incidents and attacks.

We proposed three models for predicting thread structures. A A prompt-based learning model called Next Paragraph Prediction with Instructional Prompting (NPP-IP) for predicting thread structures across different cybersecruity topics was introduced. The three methods were evaluated using two different datasets and compared against several well known methods. The results show that the NPP-IP method had considerable improvement over existing methods, achieving the highest F1 score across different real world hacker forum datasets.

Chapter 4

IDENTIFYING KEY USERS CONSIDERING USER INTERACTIONS AND CYBER ATTACK RELEVANCE OF THE HACKER FORUMS' POSTS

4.1 Introduction

From national defense to private industry, cybercrimes cost trillions of dollars in damages worldwide impacting different sectors of society each year [47]. Current trends of cybercrimes indicate a considerable rise in the future as hacker tools become more sophisticated and ubiquitous [101]. This trend is partially caused by the advent of the dark web which has given hackers the opportunity to interact, profit, and exchange information on the dark web forums [52]. However, there are users with different levels of knowledge in the hacker forums, and the cybersecurity researchers who want to identify emerging cyber threats need to scrutinize these individuals to find key hackers (users) [163]. Thus, identifying key users is an important task to predict cyber attacks since highly-skilled hackers are usually more successful in their goals [102]. For instance, "WannaCry" ransomware attack directed against not only hospitals in the U.K. but also numerous other worldwide targets was discussed on a dark web forum a few weeks prior [143]. In addition, "Anna-Senpai" released the source code of the Mirai Botnet on a popular hacker forum in 2016, and then numerous copycat botnets have been made since then [144]. Identifying the key users is a complex problem for cybersecurity since they are a small percentage of the hacker forum users.

There are several works using social network analysis (SNA) to identify key users in hacker forums [162, 133, 89, 115]. There are challenges for key users. Since some

of these forums discuss a variety of topics such as hacking tools, drugs, and firearms, some works use the filters to remove the unrelated posts or topics [10]. In addition, since most of the forums are unstructured, many works use some assumption to create the thread structures such as Creator-oriented network and Last Reply-oriented Network based upon temporal interaction assumptions [82] for social networks [132, 10, 136, 89, 120]. The users' interactions must be considered to build the social network [48], however, these works did not consider the user interactions since there was no context considered method until the Next Paragraph Prediction (NPP) method [71] was proposed.

In our approach, we improve the quality of the extracting key (influential or knowledgeable) users in the hacker forums who provide cyber incidents and attacks information using our proposed three points. (i) Building the thread structure considering the user interaction using the Next Paragraph Prediction (NPP) method [71]. Instead of the existing two assumption network methods, this approach considers the user interactions based on the post contents. Thus, the accuracy of the built thread structure will increase since most of the (hacker) forums are unstructured. (ii) Since our interest is to identify the influential users who provide cyber incidents and attacks information, we do not want to consider the users who post unrelated topics or put less weight on them. We introduce Cyber Attack Relevance Scale (CARS) and the model predicts the scale of posts to filter the forum posts. Since many previous works manually analyze the results to remove these unrelated posts or users, our approach will automatically eliminate or give less weighting for the posts of unrelated topics. (iii) Combining the above two methods to create a new social network analysis tool, NPP-CARS, to identify the influential users in the hacker forums who provide cyber incidents and attacks information.

We evaluate the CARS dataset with eight classifiers, and social network analysis

with the combination of the best CARS model and NPP method with six English hacker forums. The instructional prompting BiLSTM model shows average F1 score 0.81 as the best performed CARS model. The comparison of the extracted top 10 users from the traditional approaches and our new approach, NPP-CARS, shows that the NPP-CARS approach can extract more potentially useful users who post many cybersecurity related topics than the other approaches. Thus, our NPP-CARS approach will assist in extracting cyber threat intelligence from hacker forums, and predict future cyber incidents or attacks from the posts in the hacker forums.

The contributions of this work are as follows:

- Introducing Cyber Attack Relevance Scale (CARS) for scaling posts based on how much exploited cyber incident or attack related information in the posts.
- We apply Prompt-based learning to the Cybersecurity domain for the first time.
- We apply the Next Paragraph Prediction (NPP) method to predict thread structure for social network analysis for considering user interactions.
- The evaluation results show that our approach, NPP-CARS, can extract more users who mainly discuss cyber incidents and attack related topics than the traditional methods.

The rest of the paper is organized as follows: we introduce several terms and applications related to our proposed approach in Section 5.2, proposed our approach and methodology in Section 4.3, training performance of CARS in Section 4.4, then the experimental evaluations of social network analysis in Section 4.5, finally the analysis and discussion of the experimental evaluation in Section 4.6.

4.2 Related Work

In this section, we describe the background and related works: (i) social network analysis in hacker forums for identifying key users, and extended works using social network analysis, (ii) extracting social structure and networks from unstructured forums, and (iii) automatic scoring of posts with several features for evaluating the post content.

4.2.1 Social Network Analysis in Hacker Forums

Currently, there are several works using SNA techniques to analyze hacker forums. For instance, the recent work [120] conducted SNA over six dark web forums. Their findings only help to understand the communities of the six dark web forums, and some analysis of the central nodes in these networks. However, they did not extract the key individuals who contribute to these forums. Another work [66] combined text analysis with Latent Dirichlet Allocation (LDA) and SNA with centrality measures. LDA is a topic modeling algorithm and it has an ability to find unobserved groups (i.e. identify latent topics). The combination is able to identify proficient criminals i.e. key hackers in a real-world hacker forum. However, their approach needs manual efforts to find the cybersecurity related topics or not, once their model has processed the forum posts. The other work [115] analyzed the characteristics and pathways of key actors (forum users who have been linked to criminal activities such as providing services and tools to disrupt systems and networks or using these tools to perform attacks). They proposed tools to automatically identify likely key actors. The combination of the results of a logistic regression model with k-means clustering and social network analysis, can verify the findings using topic analysis. They identified variables relating to forum activity that predict the likelihood a user will become an

actor of interest to law enforcement, and would therefore benefit the most from intervention. In addition, some works use the results of SNA as a feature to predict cyber threats. For instance, Almukaynizi et al. [10] filtered posts in the hacker forums by existence of CVE (Common Vulnerabilities and Exposures) numbers in the post or not. Thus, they only use the posts containing their needed information in the forums instead of the whole forum posts.

To build the social network, the members' interactions must be considered [48]. However, these works use some network representations for building the social networks in forums. The recent work, HackerRank [62], combines social network analysis and topic modeling to cluster the forum posts into the topics, then extract key users of each topic. To create the social network of the forums, they use some assumptions to extract the graph structure of the threads.

In the next subsection, we explain the current social network generation methods and issues.

4.2.2 Extracting Social Structure and Network

In order to build social networks from forums, member interactions must be correctly identified via posts on threads. There are two network representations introduced [82] for building the social network in forums: Creator-oriented Network and Last Reply-oriented Network. The Last Reply-oriented Network is widely used for the social network analysis in the recent works [124, 10, 89, 136, 120, 66]. Since these two traditional network conversion approaches are based on limited information and considerable assumptions on interactions between users, the social structures of the networks are unlikely to be accurate representations. Other recent works have predicted helpful posts in the forums [56] using a neural network based model that determines whether the post is useful or not. However, the importance of a post has

very little utility when predicting interactions and social networks. More recently, Kashihara et al. [71] proposed the Next Paragraph Prediction (NPP) method which extended BERT's Next Sentence Prediction to predict the response post from the previous post. This method allows for the Reconstruction of social networks using thread structure prediction.

4.2.3 Automatic Scoring of Posts

Measuring the importance of a comment or post in online communities has been widely researched, and there are both manual and automatic ways of performing the task. Using a Japanese news site as an example, the constructiveness score [49] is introduced to label each comment on the site with a graded numeric score that represents the level of constructiveness for ranking comments. They defined the C-score as the number of crowdsourcing workers who judged a comment to be constructive as an answer to a yes-or-no question. For instance, a C-score of 8 in a comment means that eight workers judged the comment as constructive. However, their experimental result shows that C-scores are not always related to users' positive feedback.

The ability to automatically rate postings in online discussion forums, based on the value of their contribution, enhances the ability of users to find knowledge within this content. In general, Quality Dimensions (QDs) are some common features that are applied for enhancing information and the thread retrieval [58, 30, 8, 109]. Many QDs features were used for identifying the non-quality (irrelevant), low-quality (partially relevant) and high-quality (relevant) replies in the threads to their initial posts of the threads. In addition, the classification and the feature selection techniques were used for identifying appropriate features for the forum threads, which could help in achieving significant improvement in retrieval performance.

In [153], the authors applied the relevancy dimension and the popularity dimension

features for evaluating. The evaluation is to see if a post was related to the topic of discussion or if the post was quoted or answered by other users in the thread. They introduced five categories (22 features): (i) Relevance, (ii) Originality, (iii) Forum-specific features, (iv) Surface features, and (v) Posting-component features. Some studies applied four feature classes: the lexical syntactic, surface, forum specific, and similarity features for assessing the forum post quality [158, 157]. On the other hand, the appropriateness of the lexical dimension features is not confirmed, since the thread postings of the forum do not follow correct linguistic rules [157, 153]. The work by Osman et al. [110] used 28 different quality features in six quality dimensions: Relevancy dimension, Author Activeness dimension, Timeliness dimension, Ease-of-understanding dimension, Politeness dimension, and Amount-of-data dimension.

There is no measurement for the posts that mentioned about cyber incidents or attacks. Thus, we introduce the Cyber Attack Relevance Scale (CARS).

4.3 Our Approach and Methodology

4.3.1 Cyber Attack Relevance Scale (CARS)

The hacker forums have threads about not only the cyber incident or attack related topics but also other non related topics. Since we want to extract key users who provide cyber incident and attack information, these non related topics need to be filtered. Thus, we introduce the Cyber Attack Relevance Scale (CARS) to categorize the posts for filtering, and the definition is as follows:

- CARS Not Relevant (CARS-NR): a post is not relevant to cyber attack.
- CARS Low (CARS-L): a post is relevant to cyber attack and has low details e.g. containing few cyber attack related keywords in a post.

CARS	Sample Post
CARS-NR	Anybody Have A Career In Cyber Security?
CARS-NR	You're the head of security and your password is password.
CARS-L	Is it possible that my Iphone has been compromised with a **zero day exploit**?
CARS-L	MAC address conflict in **IP spoofing attack**
CARS-M	[**CVE-2019-14615**] **iGPU Leak**: An **Information leakage vulnerability** on Intel Integrated GPU.
CARS-M	**Rag** allows the user to generate there own personal custom reverse shells. Like a normal shell it allows the **hacker** to **control there computer** and then give custom computer commands.
CARS-H	**Invicta Group**, a French company specializing in wood heating is **down after a cyber attack**.
CARS-H	**California City** is **hit with a ransomware attack**.

Table 4.1: Examples of Annotation for Each CARS. Bold Keywords and Phrases are Related to Cyber Attacks.

- CARS Medium (CARS-M): a post is relevant to cyber attack and has details of attack e.g. containing cyber attack related keywords and some detailed information such as target and reference link, but the post is not clear that the attack is happened or ongoing.

- CARS High (CARS-H): a post is relevant to cyber attack and has the detail of how the attack/incident happened or is happening.

Higher CARS scale has more detailed information related to cybersecurity incidents.

4.3.2 Data and Annotation for CARS

For a supervised deep learning approach to build a CARS model, a corpora of over $7,000$ posts and comments from nine cybersecurity related subreddits curated directly from Reddit in the term from January 1st, 2020 to December 31st, 2020 and Hackmageddon by Paolo Passeri [114]. After the collection of subreddits, the annotation of all posts is performed by four cybersecurity experts. Each post is checked by at least three of them to avoid the expert's bias. If a post's scale is different from each expert, pick the highest voted scale. The site "Hackmageddon"

is the collection of the cyber attacks data from various security news sites. This data contains the date of when the attack was reported, hacker group name or malicious tool names that related to the attack, target of the attack, and the type of attack such as DDoS, Malware, and Vulnerability (Exploitation). The description of the cyber attacks and incidents are used as CARS-H data. We create the balanced dataset which has $1,867$ posts per each CARS, and the total number of posts is $7,468$.

Table 4.1 shows sample posts of each CARS. CARS-NR does not related to cyber incidents or attacks. CARS-L has some cyber attack or incident related keywords. CARS-M has several keywords but the post cannot determine that a cyber incident or attack is on going, happened or not. CARS-H has several keywords and they indicate that a cyber incident/attack happened or is on going.

4.3.3 CARS model

Since many new keywords related to cybersecurity are added or generated frequently, especially malicious software names and cybercriminal group names, the traditional models with some keyword dictionaries will not work for new keywords. Thus, we use several combinations of classification models and word embeddings. Especially, the recent works of cybersecurity NER models [70, 69] show that their models that are the combination of neural network models and word embeddings can predict new cybersecurity entities that do not appear in the training data. We expect that the combination of neural network based classifiers with word embeddings can understand new cybersecurity keywords that do not appear in the training and perform better.

To evaluate our CARS approach, we use seven different models. There are four traditional text feature classification models, and three combination models of neural network models and word embeddings to understand new keywords that do not appear

in the training data. The seven models are listed as follows; Random Forest (RF), Linear Support Vector Classification (LinearSVC), Multinomial Naive Bayse (MUltinomialNB), Logistic Regression, Convolutional Neural Network (CNN), Bidirectional Long Short Term Memory Network (BiLSTM), and T5 that is a transformer-based generative model [27]. We use several packages for implementing the models. The scikit-learn package [117] is used for RF, LinearSVC, MultinomialNB, and LR. PyTorch [116] and Keras [33] are used for CNN and BiLSTM.

We use TF-IDF vectors as a feature for RF, LinearSVC, MultinomialNB, and LR. For CNN, we use the implementation idea of using a CNN to classify text from TextCNN [74], and GloVe word embeddings [119] as the features for CNN and BiLSTM. T5 uses T5-base model for fine-tuning.

For CNN, BiLSTM and T5 models, we apply the instructional prompting [100] to improve the performance. As Figure 4.1 shows, the instructional prompting dataset added the task description and four CARS examples for each scale before giving the original post content. The task description we define in here is "You give a post and generate CARS-H if post contains exploited cyber attacks or incidents, CARS-M if post contains keywords of potential cyber attacks or incidents, CARS-L if post contains few of the cyber attack keywords, and CARS-NR if post is not related to cyber attack."

4.3.4 Social Structure Construction with CARS

Using the fine-tuned model for Next Paragraph Prediction, Social Structure Construction algorithm builds the social structure of an unstructured forum to generate the social network of users therein. Algorithm 7 shows the process to generate the social structure of the given unstructured forum. If the Prediction model (PM in Algorithm 7) returns "true" for given two individual posts from same thread, the

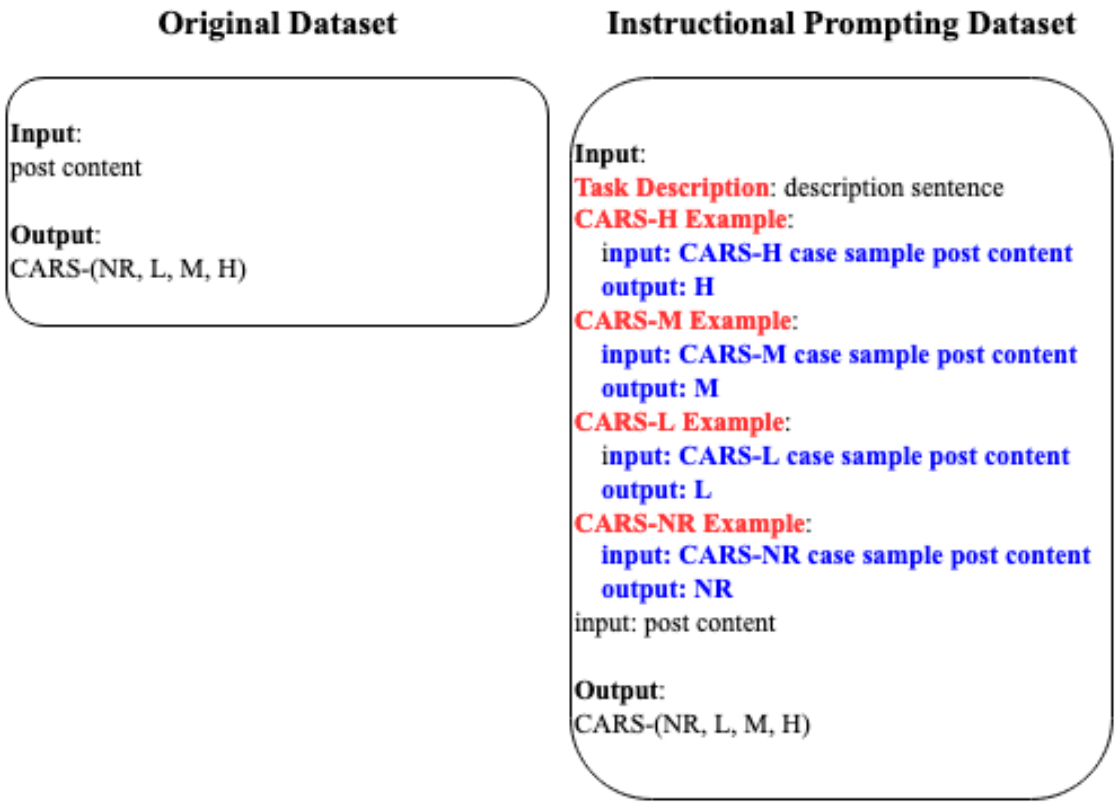

Figure 4.1: The Data Structure of the CARS Dataset and Instructional Prompting Dataset.

thread structure puts the edge between the two posts' nodes. The weight of the edge is calculated based on the CARS of post1 and post2. To assign the weight of each CARS, we define that CARS-NR has no weight and CARS-L has weight 1, then higher scale's weight increases by 100% each. Thus, we assign each weight as follows; CARS-NR = 0, CARS-L = 1, CARS-M = 2, and CARS-H = 3 respectively.

$$edgeW(post1, post2) = (CARS(post1)) * (CARS(post2))$$

, where $edgeW$ is a weight of edge, $CARS(post1)$ means the CARS of post1 respectively. Thus, if the CARS of post1 or post2 is CARS-NR, the weight of the edge is 0.

Once the social structure of an unstructured forum is built, the social network (user network) is extracted for the social network from the social structure for Social Network Analysis. This approach will build an accurate social network for unstructured forums compared to the traditional approaches: Creator-oriented Network and Last Reply-oriented Network.

4.4 CARS Model Performance

To evaluate each model with CARS dataset, we randomly split the human annotated CARS dataset into Train dataset and Test dataset. The split ratio is 80% for Train dataset and 20% for Test dataset.

Method	CARS	Precision	Recall	F1
RandomForest	NR	0	0	0
	L	0	0	0
	M	0.77	0.09	0.17
	H	0.25	1	0.4
LinearSVC	NR	0.76	0.67	0.71
	L	0.6	0.61	0.6
	M	0.75	0.84	0.79
	H	0.97	0.96	0.97
MultinomialNB	NR	0.79	0.62	0.69
	L	0.59	0.68	0.63
	M	0.72	0.68	0.7
	H	0.9	1	0.95
LogisticRegression	NR	0.93	0.27	0.42
	L	0.5	0.65	0.56
	M	0.66	0.88	0.75
	H	0.93	0.99	0.95
TextCNN	NR	0.81	0.76	**0.78**
	L	0.65	0.71	**0.68**
	M	0.86	0.81	0.83
	H	0.94	0.98	0.96

Method	CARS	Precision	Recall	F1
BiLSTM	NR	0.81	0.71	**0.76**
	L	0.59	0.74	0.66
	M	0.86	0.77	0.81
	H	0.97	0.97	**0.97**
T5	NR	0.87	0.69	0.77
	L	0.65	0.76	**0.7**
	M	0.84	0.87	**0.85**
	H	0.99	0.99	**0.99**
Prompt-TextCNN	NR	0.78	0.77	0.77
	L	0.61	0.77	0.66
	M	0.87	0.73	0.79
	H	0.93	0.95	0.94
Prompt-BiLSTM	NR	0.79	0.79	**0.79**
	L	0.64	0.7	0.67
	M	0.86	0.79	**0.82**
	H	0.97	0.97	**0.97**
Prompt-T5	NR	0.87	0.7	0.77
	L	0.64	0.73	**0.68**
	M	0.82	0.88	**0.85**
	H	0.99	0.99	**0.99**

Table 4.2: Result of CARS Models.

For evaluating the CARS dataset with the six existing method and two neural

Algorithm 7 SocialStructureConstruction

Input: Unstructured threads in a forum *Forum*, Prediction model *PM*

Output: *SocialStructure*

ForumStructure
for all *Thread* $\in$ *Forum* **do**
  ThreadStructure
  postList =list of all posts in *Thread*
  for $1 \leq i \leq |postList|$ **do**
    for $1 \leq j \leq |postList|$ **do**
      if $i \neq j$ and *postList*[j] posted after *postList*[i] **then**
        post1 = *postList*[i]
        post2 = *postList*[j]
        if $PM(post1, post2)$ returns True **then**
          ThreadStructure add the edge from *post2* to *post1*
          Add the weight of the edge based on the CARS scale of *post1* and *post2*.
        end if
      end if
    end for
  end for
  ThreadStructure is added to *ForumStructure*
end for
Generate *SocialStructure* of the *Forum* based on *ForumStructure*

network classifiers with instructional prompting, and compared performance using Precision, Recall, and F1 score metrics reported in Table 4.2. The combinations of word embeddings and neural network classifiers perform better than the traditional

classifiers. When we apply instructional prompting to CNN, BiLSTM, and T5 models, the prompt-CNN and prompt-T5 decreased the performance, however, the prompt-BiLSTM improved the performance. T5 models perform three best F1 scores out of four CARSs. However, T5 model's CARS-NR performance is not high. Thus, we use the second best performed model, the prompt-BiLSTM model, in social network analysis since the model got the highest CARS-NR F1 score that the model can filter non cyber attack related posts more accurately. Both T5 and BiLSTM models use word embeddings as features and consider semantics. Therefore, these models can capture new words or phrases about cyber attacks such as jargon and new tool names.

4.5 Social Network Analysis

To compare with the existing approaches for social network analysis (Creator-oriented network and Last Reply-oriented network), we compare the top 10 users of each analysis result with the six English (mainly speaking) hacker forums (collected the posts from January 1st, 2021 to December 31st, 2021). The basic statistics is in Table 4.3. These forum data are obtained from a cyber-threat reconnaissance firm (called CYR3CON [1]).

Forum	1	2	3	4	5	6
Users	15656	410	1197	749	1943	3163
Total Threads	6954	260	558	738	2438	856
Total Posts	66135	3115	4378	27345	35788	7246
Average Posts Per Thread	9.51	11.98	7.85	37.05	14.68	8.46

Table 4.3: The Basic Statistic of Six English Hacker Forums with the Number of Threads in 2021.

There are many centrality measurements to find the important users in social

[1]https://www.cyr3con.ai

network analysis. However, some of the centrality measurements have disadvantages for this hacker forums' social network analysis task. For instance, the two major centrality measurements, Degree centrality and Eigenvector centrality, have the following disadvantages. The degree centrality returns a high centrality score when a user (node) connects the most nodes. This score can be easily changed by minimal local operation i.e. adding many dummy users to connect a specific user. Eigenvector centrality has a disadvantage for directed graphs such as the thread structures we use. Eigenvector centrality considers that the neighbors of important nodes are important. Thus, the Eigenvector centrality of a node that does not have indegree such as the original post of a thread becomes 0, and the centrality of the neighbor nodes of the node that only get the edges from the node indegree 0 nodes also become 0. Thus, this centrality will not work well for the users who mainly start new threads. On the other hand, PageRank, Betweenness centrality and Closeness centrality are able to get over these disadvantages. PageRank can avoid the disadvantage of Eigenvector centrality. Betweenness centrality returns a high score on a node that appears often on any path of two nodes in the network. Therefore, a node with a low degree in a bridge of groups in the network can get a high score i.e. we can find the users who bridge multiple groups in the network. Closeness centrality returns a high score for nodes that have similar average distance from other nodes. Thus, we use the combination of PageRank, Betweenness centrality and Closeness centrality to rank the key users. Algorithm 8 shows how to rank the users.

In order to evaluate the effectiveness of our approach, we compare the average ratio of non CARS-NR over the top 10 users in each approach for metric reported in Table 4.4. Since most of the forums have more than 1000 users, we pick top 10 users for the comparison. When an approach has the average ratio higher than the others, the approach identified users who posts more cyber incidents and attacks

Algorithm 8 UserRanking

Input: List of Users containing the values of PageRank, Betweeness centrality and Closeness centrality

Output: *Top10UserList*

1: $PageRank - cr = 1$

2: $Betweenness - cr = 1$

3: $Closeness - cr = 1$

4: $Top10UserList = []$

5: **while** number of *Top10UserList* is less than 10 **do**

6: **if** $PageRank - cr == Betweenness - cr == Closeness - cr$ **then**

7: Compare the centrality values of them.

8: Add *user* of the highest centrality value to *Top10UserList*.

9: Increase the highest centrality value's $-cr$ +1.

10: **else**

11: Find the lowest $-cr$ centrality.

12: **if** There are multiple centrality having same lowest $-cr$. **then**

13: Compare the centrality values of them.

14: Add *user* of the highest centrality value to *Top10UserList*.

15: Increase the highest centrality value's $-cr$ +1.

16: **else**

17: Add *user* of the highest centrality value of the lowest $-cr$ centrality to *Top10UserList*.

18: Increase the centrality's $-cr$ +1.

19: **end if**

20: **end if**

21: **end while**

Forum	LR	CO	NPP	NPP-CARS
1	**0.37**	0.35	**0.37**	**0.37**
2	0.55	0.44	0.46	**0.61**
3	0.23	0.23	0.18	**0.35**
4	0.76	0.77	0.78	**0.80**
5	**0.81**	0.74	**0.81**	0.79
6	0.58	**0.67**	0.58	0.63

Table 4.4: The Average Ratio of Non CARS-NR (CARS-L or Higher) for Top 10 Users in Each Approach for Six English Forums.

related topics. Our approach, NPP-CARS, has the highest ratio in four out of six forums and the rest of two forums, NPP-CARS reached the second highest ratio. This result shows that the NPP-CARS approach can identify more users who post more cybersecurity related contents.

We checked the top 10 users from each approach for six forums, and compared the number of threads, the number of posts, the ratio of each CARS, and the topics of each user discussed. Since some users posted multiple topics, we use the shorter version of topics listed in Table 4.5. The top 10 users detailed results for six forums are in Table 4.6, 4.7, 4.8, 4.9, 4.10, 4.11 respectively. We highlighted some users who appeared in only one approach i.e. unique users than the others, and the ratio of CARS-H if the ratio is over 10%.

We found that the common topics in each forum are different from the other forums. For instance, one of the common topics in Forum 1 is "Data Leak", and one of the common topics in Forum 6 is "Game".

Topic	Shorten
Data Leak	DL
Software as a Service	SaaS
Crime as a Service	CaaS
Share Files	SF
Vulnerability info	VL
Malware	ML
Stolen or Pirate Software (Activation Key)	S/PS
(Stolen) Credit Card Information	CC
Suspicious IP addresses and URL	SIP&U
Stolen or Fake Bank Accounts	SBA
Fake Phone Numbers	FPH
Suspicious Tools and AddOns	STA
Crypto Currency (Mining)	CCM
Selling Code (malicious)	SC
Scam site information	SSI
Hijacking System	HS
Tools Dump	TD
Botnet	BN
Cyber incidents	CI
Market Places	MKP
Fake ID	FID
Zero-Day	ZD
Software Patch Info	SP
Asking Something	ASK
Jail Break Software	JBS

Table 4.5: The Major Topics and Their Shortened Versions.

LR	CO	NPP	NPP-CARS	# of T	# of P	CARS-NR Ratio	CARS-L Ratio	CARS-M Ratio	CARS-H Ratio	Main Topics
5521109	5521109	5521109	5521109	267	316	58.86%	30.70%	10.44%	0.00%	DL
2730819	2730819	2730819	2730819	351	544	70.77%	19.30%	9.93%	0.00%	SaaS, CaaS, DL
6181200	6181200	6181200	6181200	1356	1516	88.98%	9.56%	1.45%	0.00%	DL
6251104	6251104	6251104	6251104	1067	1158	82.21%	17.53%	0.26%	0.00%	DL
3919428	3919428	3919428	3919428	566	642	95.64%	4.21%	0.16%	0.00%	SF
4826551	4826551	4826551	4826551	498	833	29.05%	33.97%	36.73%	0.24%	DL, VL
6971481	6971481	6971481	6971481	330	385	80.78%	14.81%	4.42%	0.00%	ML
			5224406	222	290	64.83%	21.38%	13.79%	0.00%	DL, VL, S/PS
	3919713			180	243	72.02%	21.40%	6.58%	0.00%	DL
3919085	3919085	3919085	3919085	314	581	28.23%	33.22%	38.38%	0.17%	DL, ML
	3928263			119	162	45.06%	32.10%	22.84%	0.00%	DL
3921803		3921803	3921803	132	365	34.52%	42.74%	22.19%	0.55%	DL
7229652		7229652		40	269	63.57%	23.79%	12.64%	0.00%	DL

Table 4.6: The Top 10 Users of Each Approach in Forum 1.

LR	CO	NPP	NPP-CARS	# of T	# of P	CARS-NR Ratio	CARS-L Ratio	CARS-M Ratio	CARS-H Ratio	Main Topics
	3818634			2	3	33.33%	33.33%	33.33%	0.00%	CC
	9557676			1	3	66.67%	0.00%	33.33%	0.00%	DDoS, CaaS
	3817816			40	336	0.00%	0.00%	97.92%	2.08%	SIP&U
	5123794			7	22	77.27%	13.64%	9.09%	0.00%	CaaS, SBA
7849662		7849662		9	15	93.33%	6.67%	0.00%	0.00%	FPH
3807335		3807335	3807335	6	49	0.00%	12.24%	87.76%	0.00%	CCM, STA
	3809312			1	3	66.67%	33.33%	0.00%	0.00%	SBA
			3812558	8	11	72.73%	27.27%	0.00%	0.00%	SBA
	3811808		3811808	2	12	58.33%	25.00%	16.67%	0.00%	SBA
	3807185			4	5	80.00%	20.00%	0.00%	0.00%	CC
	4002424			13	16	93.75%	6.25%	0.00%	0.00%	CCM, CC
	9642624			1	2	50.00%	50.00%	0.00%	0.00%	SC
		3794648		3	9	88.89%	11.11%	0.00%	0.00%	CC
5132911		5132911	5132911	10	121	34.71%	7.44%	57.85%	0.00%	DL, CC
3813865		3813865	3813865	1	8	12.50%	25.00%	62.50%	0.00%	CC
3797299	3797299	3797299	3797299	2	14	35.71%	42.86%	21.43%	0.00%	CC, SSI
3818615		3818615	3818615	1	4	0.00%	75.00%	25.00%	0.00%	DL, CC
3813927		3813927	3813927	3	11	72.73%	9.09%	18.18%	0.00%	CC
3811306			3811306	18	171	0.00%	0.00%	100.00%	0.00%	SIP&U
4015977		4015977		4	9	100.00%	0.00%	0.00%	0.00%	-
8165015		8165015	8165015	1	2	100.00%	0.00%	0.00%	0.00%	-

Table 4.7: The Top 10 Users of Each Approach in Forum 2.

4.6 Analysis and Discussion

In this section, we analyze and discuss the results of CARS models and top 10 users from four approaches with social network analysis in the six hacker forums.

LR	CO	NPP	NPP-CARS	# of T	# of P	CARS-NR Ratio	CARS-L Ratio	CARS-M Ratio	CARS-H Ratio	Main Topics
8898707	8898707	8898707	8898707	3	30	100.00%	0.00%	0.00%	0.00%	SC, Russian
9629970		9629970	9629970	4	7	71.43%	0.00%	28.57%	0.00%	HS
	4042820			17	46	80.43%	6.52%	13.04%	0.00%	CC, SBA, Russian
4042690	4042690	4042690	4042690	30	116	73.28%	17.24%	8.62%	0.86%	SBA, ML, Russian
6004329	6004329	6004329	6004329	23	47	61.70%	27.66%	10.64%	0.00%	SBA, CC, Russian
	8898445			2	4	25.00%	0.00%	75.00%	0.00%	BN, DL, Russian
9516400		9516400	9516400	1	19	94.74%	0.00%	5.26%	0.00%	HS
4042354	4042354	4042354	4042354	127	135	80.74%	2.96%	16.30%	0.00%	MKP
	8898169			2	5	100.00%	0.00%	0.00%	0.00%	-
			9574975	1	10	70.00%	20.00%	10.00%	0.00%	VL, ZD
	6232318			11	25	84.00%	16.00%	0.00%	0.00%	CCM, Russian
6764323		6764323		7	10	100.00%	0.00%	0.00%	0.00%	-
	7735177			2	6	100.00%	0.00%	0.00%	0.00%	-
		7347129		5	43	90.70%	0.00%	9.30%	0.00%	DL, Russian
			6232509	4	73	4.11%	1.37%	58.90%	**35.62%**	DL, Russian
8235130			8235130	2	10	40.00%	20.00%	40.00%	0.00%	BN, Russian
5123754		5123754	5123754	1	18	55.56%	27.78%	11.11%	5.56%	FID
	7139616			1	3	66.67%	33.33%	0.00%	0.00%	ASK
8304698		8304698		2	14	92.86%	7.14%	0.00%	0.00%	FID, Russian

Table 4.8: The Top 10 Users of Each Approach in Forum 3.

LR	CO	NPP	NPP-CARS	# of T	# of P	CARS-NR Ratio	CARS-L Ratio	CARS-M Ratio	CARS-H Ratio	Main Topics
4369485		4369485	4369485	162	593	14.67%	57.34%	27.49%	0.51%	VL, ML
	10004192			114	364	65.38%	23.90%	6.87%	3.85%	CI
4370276	4370276	4370276	4370276	199	805	26.09%	62.24%	11.43%	0.25%	CI, ML, STA
8299692	8299692	8299692	8299692	138	592	20.10%	60.81%	18.58%	0.51%	CI, ML, STA
5288347	5288347	5288347	5288347	157	703	19.91%	58.89%	19.91%	1.28%	CI, ML, STA
	4369576			218	1137	13.98%	26.56%	57.34%	2.11%	CI
	4369440			111	371	34.23%	49.33%	16.44%	0.00%	VL, ML, SP
	4369566			96	459	15.03%	34.64%	50.33%	0.00%	SP
4369582		4369582	4369582	124	482	30.71%	55.60%	12.24%	1.45%	VL, ML
4369576		4369576	4369576	218	1137	13.98%	26.56%	57.34%	2.11%	CI, VL
			4369566	96	459	15.03%	34.64%	50.33%	0.00%	VL, SP
4369373	4369373	4369373	4369373	156	536	5.78%	29.66%	58.77%	5.78%	CI, SP
4386816				106	412	54.85%	38.59%	5.34%	1.21%	CI
4388079	4388079	4388079	4388079	82	900	9.11%	61.00%	29.78%	0.11%	VL, ML, SP
	4370303	4370303		123	427	25.29%	69.32%	4.92%	0.47%	VL, SP
4390969		4390969	4390969	123	550	49.45%	34.36%	15.64%	0.55%	DL, VL, ML, SIP&U

Table 4.9: The Top 10 Users of Each Approach in Forum 4.

4.6.1 CARS models

We train and evaluate our created CARS dataset with eight models. Four of them are the traditional classification models with word embeddings of the posts as features, and the rest of them are neural network classifiers with word embeddings.

LR	CO	NPP	NPP-CARS	# of T	# of P	CARS-NR Ratio	CARS-L Ratio	CARS-M Ratio	CARS-H Ratio	Main Topics
4370036	4370036	4370036	4370036	2344	6009	0.47%	2.88%	87.09%	9.57%	CI, VL, ML, SP
4481660	4481660	4481660		161	474	21.52%	54.85%	23.42%	0.21%	CI, VL, ML
4481431	4481431	4481431	4481431	334	886	27.99%	49.89%	19.86%	2.26%	VL, STA
4471619	4471619	4471619	4471619	160	757	31.70%	46.10%	20.21%	1.98%	VL, ML
4472084	4472084	4472084	4472084	426	1376	12.28%	54.87%	32.56%	0.29%	CI, VL, ML
4472080	4472080	4472080	4472080	155	722	42.52%	44.60%	12.33%	0.55%	CI, ML
	4472247			67	254	30.31%	56.30%	13.39%	0.00%	VL
4471581		4471581		58	248	17.34%	70.56%	11.69%	0.40%	SP, CCM, CI
	4471571			211	508	24.80%	37.40%	31.89%	5.91%	CI, VL, SP
	4471715			211	595	42.02%	48.07%	9.58%	0.34%	CI, VL, SP
4481771	4481771	4481771	4481771	191	632	23.89%	50.32%	25.63%	0.16%	CI, ML
			4370428	485	1369	5.62%	26.52%	38.50%	**29.36%**	CI, VL, MI, SP
4370661		4370661	4370661	53	331	6.65%	12.69%	77.04%	3.63%	CI, VL, SP
7992481		7992481	7992481	27	59	5.08%	23.73%	20.34%	**50.85%**	DL, CI, ML, MKP
			8337450	1	2	50.00%	0.00%	50.00%	0.00%	SP

Table 4.10: The Top 10 Users of Each Approach in Forum 5.

LR	CO	NPP	NPP-CARS	# of T	# of P	CARS-NR Ratio	CARS-L Ratio	CARS-M Ratio	CARS-H Ratio	Main Topics
7844415	7844415	7844415	7844415	38	75	26.67%	56.00%	17.33%	0.00%	CI
7844289	7844289	7844289	7844289	54	133	39.85%	54.89%	4.51%	0.75%	JBS, Game
	7331230			26	37	75.68%	24.32%	0.00%	0.00%	SP
	7844411			63	144	27.78%	59.72%	12.50%	0.00%	SP, JBS, Game
7844539	7844539	7844539	7844539	62	107	25.23%	67.29%	7.48%	0.00%	SP, Game
	7844652			21	33	12.12%	66.67%	21.21%	0.00%	Game
7844411		7844411	7844411	63	144	27.78%	59.72%	12.50%	0.00%	SP, STA, Game
7844395	7844395	7844395	7844395	36	68	13.24%	57.35%	29.41%	0.00%	SP, STA, Game
7243798	7243798	7243798	7243798	299	460	81.09%	14.57%	4.13%	0.22%	JBS, Game,
7851928		7851928	7851928	25	73	2.74%	35.62%	38.36%	**23.29%**	JBS, Game
	7844642			26	40	15.00%	50.00%	35.00%	0.00%	Game
	8287425			12	18	16.67%	50.00%	33.33%	0.00%	CI, Game
8287135		8287135	8287135	42	82	0.00%	0.00%	86.59%	**13.41%**	CI, SP
8850108		8850108		1	2	100.00%	0.00%	0.00%	0.00%	-
			8326560	1	2	50.00%	0.00%	0.00%	**50.00%**	Illegal Drugs
8646015		8646015	8646015	1	2	100.00%	0.00%	0.00%	0.00%	-

Table 4.11: The Top 10 Users of Each Approach in Forum 6.

For two of the neural network classifiers, we apply instructional prompting for the dataset to improve the performance.

As Table 4.2 shows, Prompt-BiLSTM model performs three best F1 scores out of four CARSs. In the traditional models, most of the models did not perform well, especially Random Forest could not predict CARS-NR and CARS-L posts correctly. On the other hand, the neural network classifiers (CNN and BiLSTM) performs bet-

ter. However, when we apply the instructional prompting to them, Prompt-BiLSTM improved the performance, and Prompt-CNN decreased the performance. Generally, the CNN is used to extract the local features of the text vector, and the BiLSTM is used to extract the global features related to the text context. The instructional prompting requires to understand not only the given training text features but also the additional task description and sample information. Thus, the Prompt-BiLSTM model could use the global features of the instructional prompting dataset, and the Prompt-CNN model could not use the global features.

When we apply the trained Prompt-BiLSTM model to the six hacker forums' posts, the model can predict CARS of posts even if some posts are in a mix of English and other languages. For instance, user ID 6232509 in Forum 3 (See Table 4.8) has many CARS-H posts about selling leaked data in English and Russian. The model can predict CARS-H for both English and Russian posts mentioned selling the leaked data. Since the BiLSTM model considers the global features related to the post context, we think the model can understand not only word features but also some semantic features, and it helps to understand the other languages as well.

4.6.2 Social Network Analysis

As Table 4.4 shows, the NPP-CARS approach can extract users whose average ratio of non CARS-NR are the highest for four forums and the second highest for the other two forums. The NPP-CARS approach considers user interactions and uses the CARS to weight the edges of the social networks. This combination helps to improve the quality of extracted top 10 users who post more cybersecurity related topics. Since the number of posts and the number of threads that each user posted are different, however, we carefully checked the top 10 users of each approach in six forums one by one through reviewing their posts. Table 4.6, 4.7, 4.8, 4.9, 4.10, 4.11

show the detailed statistics of each user respectively.

Each forum has its own characters, and the extracted top 10 users of each approach show the unique topics of each forum. In Forum 1, most of the users extracted by each approach discussed leaked data (DL in the table). In Forum 2, most of the users discussed financial crimes such as stolen credit card information (CC), bank accounts (SBA), and crypto currency mining (CCM). In Forum 3, most of the users posted in English and Russian, and they mainly discussed leaked data, stolen bank accounts, and credit card information. In Forum 4, many users discussed cyber incidents (CI), vulnerability information (VL), and Malware (ML) topics. In Forum 5, the major topics are similar to Forum 4, however, many users also discussed patch information (SP) as well. In Forum 6, most of the users discussed Game and Jailbreak.

In Forum 1, most of the users posted over 100 posts to over 100 different threads, and these users are extracted by all four approaches. Two unique users were extracted by the Creator-oriented network approach and they mainly posted about data leaks. In contrast, the unique user extracted by NPP-CARS approach posted about not only data leak but also vulnerabilities and stolen or pirate software. Figure 4.2 shows some posts of the user. Discussing more than three topics by one user is rare to this forum.

In Forum 2, the Creator-oriented network approach extracted many unique users, however, most of them posted less than 10 posts in the entire forum. One of them is user ID 3817816, and most of the posts by the user are about suspicious IP addresses. The CARS model predicts some of the suspicious IP addresses posted as CARS-H, however, we could not identify these IPs are actually used for any cybercrimes or cyber incidents. In contrast, the unique user extracted by NPP-CARS approach, user ID 3812558, posted the recruitment for skilled programmers to breach some banks aggressively. Figure 4.3 shows some posts of the user. Thus, we can assume

that these banks are targeted. The unique user extracted by NPP approach, user ID 3704648, posted several links to the stolen credit card information marketplaces.

In Forum 3, two of the unique users extracted by Creator-oriented network approach have few posts and they are not related to any cyber incident or attack. In contrast, the unique users from NPP and NPP-CARS approaches discussed data leak and vulnerability information. Especially, user ID 9574975 by NPP-CARS approach posted about zero-day vulnerability topic, and claimed that he/she tested exploitation of some of the zero-day vulnerabilities. Figure 4.4 shows some posts of the user.

In Forum 4, there are four unique users extracted by Creator-oriented network approach, and most of them shared the links of cyber incident news articles or software patch information. One unique user extracted by Last Reply-oriented network approach also shared the links of cyber incident news articles. In contrast, the unique user by NPP-CARS approach shared information and links of patch and update details of a variety of software. Figure 4.5 shows some posts of the user.

In Forum 5, three unique users were extracted by Creator-oriented network approach, and these users mainly discussed the vulnerability information, especially cited many vulnerability explained article links. There are two unique users extracted by the NPP-CARS approach, and the one of them that discussed cyber incidents, specific malware, vulnerabilities related to the cyber incidents and malware and patch information of them. This user has a high CARS-H ratio, 29.36%. The other user posted one software patch information and one non cybersecurity related topic. The user ID 7992481 has the highest CARS-H ratio, 50.58%, and is extracted by Last Reply-oriented network, NPP and NPP-CARS approaches. This user shared information and links of cyber incidents, leaked data dumps, malware information, and marketplaces. Figure 4.6 shows some posts of the user.

In Forum 6, there are five unique users extracted by Creator-oriented network

approach, and most of them discussed Game and software patch information (software related to Game). Some of them also discussed jailbreak software or the game itself. In contrast, only one unique user is extracted by NPP-CARS approach, and the user posted about the drug "Methamphetamine" , especially the history and penalty of illegal usage of it in several countries. CARS model predicted this post as CARS-H since some keywords such as "crimes", "committed", and "law enforcement" are in the post and they also appeared in cyber incident reports. We suspect that model could not judge the relations of these words, drug or cybersecurity incidents correctly. Figure 4.7 shows some posts of the user.

This is the first time that we apply NPP for considering user interactions to building social networks in forums. The results show that the top 10 users by Last Reply-oriented network and NPP approaches have the same or few user differences in most of the forums. However, there are some differences that come from either considering user interaction or not. In addition, the NPP-CARS approach added the weight of the post contents to the NPP based on how much cyber incident or attack related information was in the posts. This weight helps to filter out the users that mainly posted not related to cyber incidents or attacks or less ratio of the related contents. The benefit of the NPP-CARS approach is less manual effort to get the results. The existing works use manually prepared dictionaries or regular expressions to filter the posts in the forums to remove unrelated posts, or use topic clustering to categorize the posts, then manually find the topics by checking the extracted keywords. Thus, our proposed NPP-CARS approach is simple and easy to identify users who post cyber incidents and attack related information frequently. Instead of tracking all users in the forums, focusing on the identified users by NPP-CARS approach will help to understand the current trend of the cyber incidents and attacks, and be able to predict the future cyber incidents and attacks based on the

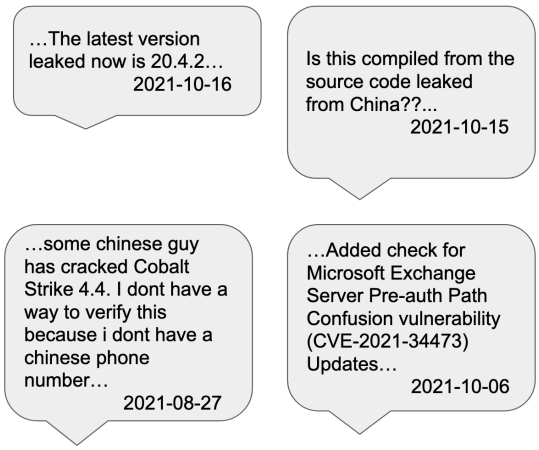

Figure 4.2: Some Posts of the Unique User of NPP-CARS Method in Forum 1.

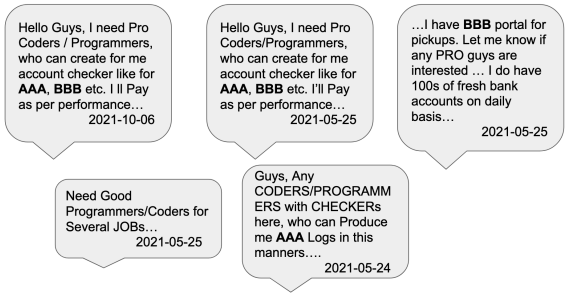

Figure 4.3: Some Posts of the Unique User of NPP-CARS Method in Forum 2. We Put AAA and BBB for the Actual Bank Names.

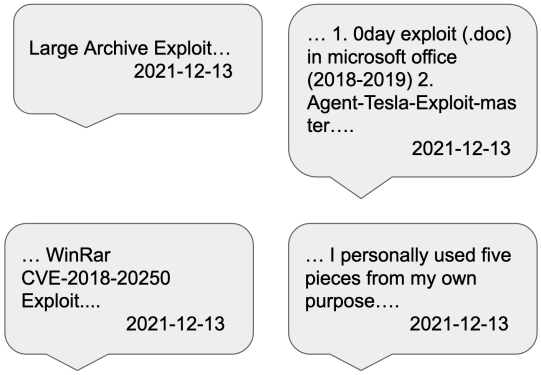

Figure 4.4: Some Posts of the Unique User of NPP-CARS Method in Forum 3.

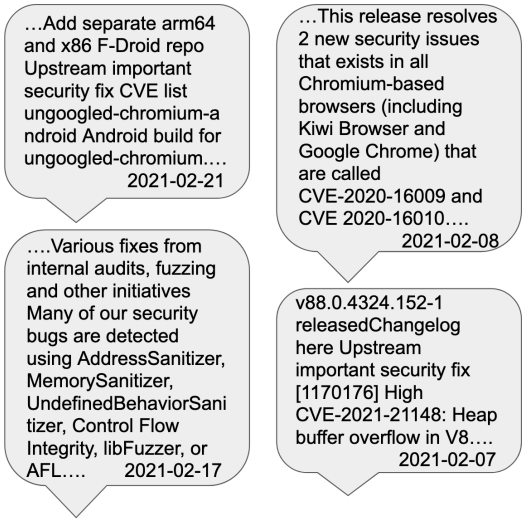

Figure 4.5: Some Posts of the Unique User of NPP-CARS Method in Forum 4.

conversations of them.

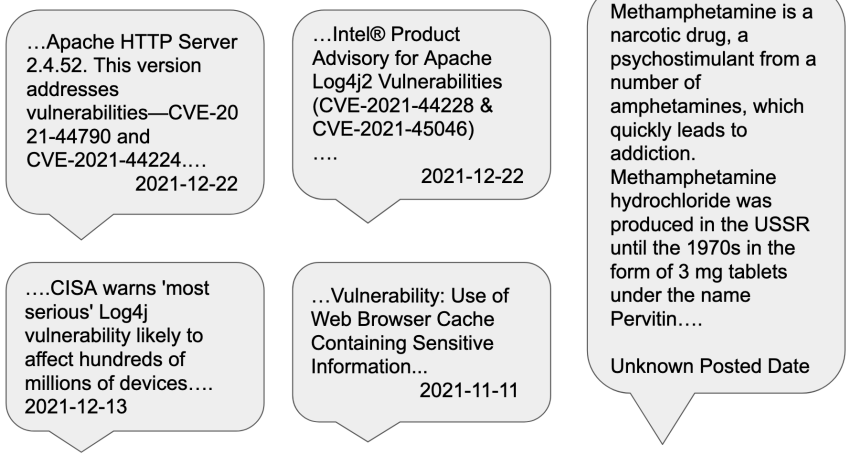

Figure 4.6: Some Posts of the Unique User of NPP-CARS Method in Forum 5.

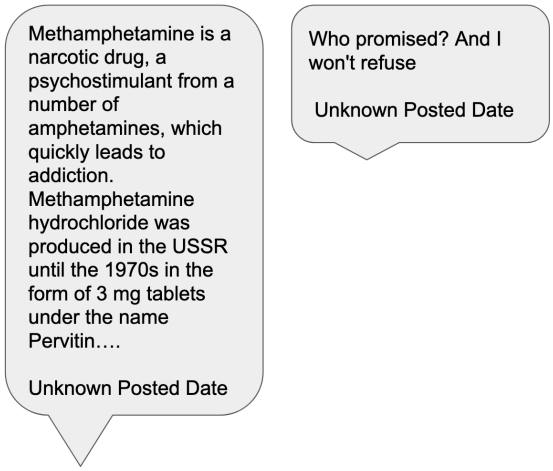

Figure 4.7: Some Posts of the Unique User of NPP-CARS Method in Forum 6.

4.7 Conclusion

Social network analysis is one of the important tasks used to identify important users who provide useful information in hacker forums and communities to predict or prevent future cyber incidents and attacks. The user interaction is one of the key factors of building the social network, especially unstructured threads in forums. Most of the recent works use the assumption methods to assume user interaction from unstructured threads. In addition, many hacker forums have not only discussed cyber incidents or attacks but also other topics such as gaming, drug, and other non-cybersecurity related topics. Social network analysis without filtering these unrelated topics may lead to inaccurate results. Thus, we propose a new approach incorporating several methods to construct thread structures based on the contents, and apply new metrics for posts to determine how much cyber incident or attack related content a post has. Then, we conduct social network analysis on six English hacker forums with a proposed approach. We also use the combination of PageRank, Betweenness centrality, and Closeness centrality to identify the top 10 users in each social network.

Firstly, we propose a new metric, the Cyber Attack Relevance Scale (CARS), for measuring the importance of a document based on the content related to a cybersecurity incident or attack. The human expert annotated CARS dataset is evaluated by several models and the best model performs average F1 score 0.81. Secondly, we combined the best CARS model and a thread structure construction method: Next Paragraph Prediction (NPP), for social network analysis with the threads from six English hacker forums. The comparison of the extracted users from the traditional approaches and our new approach, NPP-CARS, shows that the NPP-CARS approach can extract more potentially useful users who post many cybersecurity related topics. Thus, our new approach will be useful for several cybersecurity tasks such as extracting cyber threat intelligence from hacker forums, and predict future cyber incidents or attacks from the conversations in the hacker forums.

Chapter 5

DETECTING CYBERSECURITY TRENDING TOPIC PHRASES CONSIDERING CYBER ATTACK RELEVANCE

5.1 Introduction

Threat intelligence (Cyber threat intelligence or CTI) is collecting data and analysis to gain information about existing and emerging cyber threats by cyber criminals, and can help prevent security breaches in cyber space. Cybersecurity related discussion forums have the conversations that contain data that may help assist in the discovery of CTI. Thus, the adaption of CTI is important to keep one step ahead of cyber attacks or detect the on-going incidents quickly. The discovery of CTI from the forums' conversations will support to keep security measures to be proactive.

Some future threats may be detected or recognized before they turn into a problem. One of the ways to obtain CTI is to purchase a curated and analyzed feed from a specialised company such as Cognyte [3] or SurfWatch [1]. Another way is to collect Open Source Intelligence (OSINT) available from various sources on the internet. The recent researches in this space are centered around analyzing and extracting CTI information from Social Network Services and discussion forums [165, 161, 39, 38, 130, 125]. TIMiner [165] has an efficient domain recognizer based on convolutional neural network to identify CTI's targeted domain, extract an indicator of compromise (IOC) based on word embedding and syntactic dependence to identify unseen types of IOCs. The other work [161] uses a machine learning tool to classify the type of exploits targeted by hackers from the forum posts.

Topic modeling is widely used when a large collection of documents cannot be

reasonably sorted/categorized through by a person. For a given corpus comprised of many documents, a topic model discovers the latent semantic structure or topics that present in the documents. Then, topics can be used to extract high level summaries of a large collection of documents, search for documents of interests, and group similar documents together.

The topics of virality or trending terms within social medias and forums where the cybersecurity related topics are discussed, have been studied [26, 160, 37, 73, 64]. The research about the virality of cybersecurity information is sparse. One study [61] considered mentions of vulnerabilities and found that "while more security vulnerabilities are discussed on Twitter, relevant conversations go viral earlier on Reddit." Detecting trending terms in forum discussions [64] uses a lightweight method for identifying currently trending terms in relation to a known prior of terms. The method detects trending terms in longitudinal historical noisy text data of an underground hacking forum. Many of the recent researches focus on identifying importance of topics to each time period and how the topics are changing.

On the other hand, viral cybersecurity contents may be important for raising security awareness or counting widespread cybersecurity threats. Security researchers and analysts want to know the topics of posts during the specific time period in the social media and forums to understand the attention topics. For instance, Figure 5.1 shows the number of posts that contains at least one of the keywords related to Malware per week in 185 different English hacker forum sites. In this case, there are significant number of posts are observed in the different weeks. Cybersecurity experts, analysts, and researchers want to know what the topics of these weeks are discussed for understanding the contents related to security risk. In addition, they want to find some clues for ongoing or future cyber attacks and incidents from the posts.

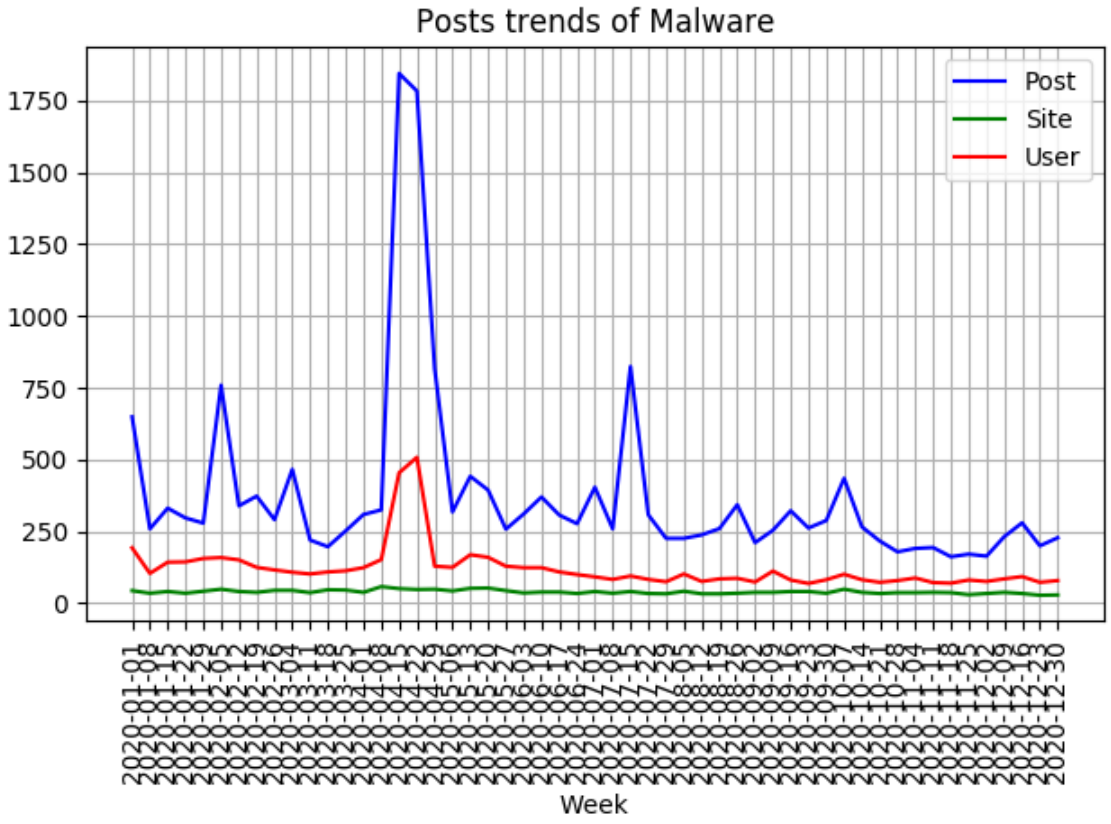

Figure 5.1: The Post Trends of Malware Category. It Shows the Changes of the Number of Posts, Unique Sites, and Unique Users per Week. There Are Significant Number of Posts in the Week of April 15th, 2020 and April 22nd, 2020.

DISCOVER [135] is an early cyber threat warning system that identifies terms related to emerging cyber threat leverages multiple online data sources such as social media, cybersecurity blogs, and forums. This framework requires some keyword dictionaries related to cybersecurity for the process. SYNAPSE [15] is a Twitter-based streaming threat monitor for threat detection. It filters the collected tweets based on the monitored infrastructure, and classifies the remaining tweets as either relevant or not, then it clusters the relevant tweets for presenting as IOC. This framework requires to train the classifier for the tweets that are related to cybersecurity or not.

The top2vec [17] and BERTopic [55] are the topic modeling methods to extract

the distributed representation of topics using semantic embeddings. They can cluster the posts in the specific week and extract the representing words of each cluster for interpretable topics. However, many cybersecurity related keywords are usually phrases such as product name and version number (e.g. Windows 10 and OSX 10.16), company name and product name (e.g. Adobe Acrobat Reader), and target and cyber attack keyword (e.g. iOS hacking). Thus, top2vec and BERTopic are not able to use cybersecurity field directly.

We propose new frameworks that receive the lists of posts per specific terms, clusters the posts in the list, and extracts the topic phrases. Our goal in this paper is to investigate the performance of existing methods and our newly proposed methods, TrendTopicExtractor, to detect the trending topic phrases. We introduce new methods to extract the distributed representation of topic phrases using semantic embeddings and Cluster-Phrase-TF-IDF (cp-TF-IDF), and Cyber Attack Relevance Scale (CARS) to categorize the relevance of posts to the cyber attack or incident (cp-TF-IDF with CARS). The experimental evaluation compares with the existing topic extraction methods; top2vec and BERTopic, with the posts that contain at least one keywords from the three cyber attack types (Malware, Phishing, and Denial-of-Service) from 185 English forum sites from January 1st, 2020 to December 31st, 2020. The evaluation results show that our methods, cp-TF-IDF and cp-TF-IDF with CARS, can cluster the posts in a specific week to a similar number of clusters from the other methods, and extract useful phrases that represent each cluster topic. The extracted phrases contain several cyber attack related tool names, targets of attacks or tools, and download URLs of the tools. In addition, our deep analysis found that our methods can find some clues of cyber attacks and incidents prior to the attack. Especially, cp-TF-IDF with CARS method gets the highest percentage of extracted topic phrases that are linked to the attacks prior to the attack happened

weeks. Thus, our proposed methods will be able to use a part of future cyber attack prediction based on the trending topic phrases in hacker forums.

The main contributions of this paper include:

- Introducing Cyber Attack Relevance Scale (CARS) for scaling posts based on how much exploited cyber incident or attack related information in the posts.
- Introducing new methods, TrendTopicExtractor (cp-TF-IDF and cp-TF-IDF with CARS), to cluster and extract topic phrases that represent the topics of given documents.
- TrendTopicExtractor can cluster the given documents into the similar number of clusters from the other existing methods without any additional input for clustering.
- The extracted phrases from cp-TF-IDF and cp-TF-IDF with CARS contain many useful phrases to represent the topics of clusters such as tool names with specific version number, attack types, and the target names.
- The analysis results show that cp-TF-IDF and cp-TF-IDF with CARS methods can extract useful phrases that are the clues of future or ongoing cyber attacks and incidents.

The rest of the paper is organized as follows: we introduce several terms and applications related to our proposed approach in Section 5.2, proposed CARS, our new method and Cluster-Phrase-TF-IDF in Section 5.3, training performance of CARS in Section 4.4, then the experimental evaluations in Section 5.4, finally the analysis and discussion of the experimental evaluation in Section 5.5.

5.2 Related Work

5.2.1 Automatic Scoring of Posts

Measuring the importance of a comment or post in online communities has been widely researched, and there are both manual and automatic ways of performing the task. Using a Japanese news site as an example, the constructiveness score [49] is introduced to label each comment on the site with a graded numeric score that represents the level of constructiveness for ranking comments. They defined the C-score as the number of crowdsourcing workers who judged a comment to be constructive as an answer to a yes-or-no question. For instance, a C-score of 8 in a comment means that eight workers judged the comment as constructive. However, their experimental result shows that C-scores are not always related to users' positive feedback.

The ability to automatically rate postings in online discussion forums, based on the value of their contribution, enhances the ability of users to find knowledge within this content. In general, Quality Dimensions (QDs) are some common features that are applied for enhancing information and the thread retrieval [58, 30, 8, 109]. Many QDs features were used for identifying the non-quality (irrelevant), low-quality (partially relevant) and high-quality (relevant) replies in the threads to their initial posts of the threads. In addition, the classification and the feature selection techniques were used for identifying appropriate features for the forum threads, which could help in achieving significant improvement in retrieval performance.

In [153], the authors applied the relevancy dimension and the popularity dimension features for evaluating. The evaluation is to see if a post was related to the topic of discussion or if the post was quoted or answered by other users in the thread. They introduced five categories (22 features): (i) Relevance, (ii) Originality, (iii) Forum-specific features, (iv) Surface features, and (v) Posting-component features. Some

studies applied four feature classes: the lexical syntactic, surface, forum specific, and similarity features for assessing the forum post quality [158, 157]. On the other hand, the appropriateness of the lexical dimension features is not confirmed, since the thread postings of the forum do not follow correct linguistic rules [157, 153]. The work by Osman et al. [110] used 28 different quality features in six quality dimensions: Relevancy dimension, Author Activeness dimension, Timeliness dimension, Ease-of-understanding dimension, Politeness dimension, and Amount-of-data dimension.

There is no measurement for the posts that are mentioned about cyber incidents or attacks. Thus, we introduce the Cyber Attack Relevance Scale (CARS).

5.2.2 TF-IDF

Term-frequency inverse-document-frequency (TF-IDF) [68] calculates and determines common words or terms in a document, however, they are not common across the entire document. This technique is a popular NLP technique and provides a mechanism for ranking words which are "important" to a document. However, texts in discussion forums are usually noisy, with varying spelling of words, and grammatical errors. Since TF-IDF requires stemming or lemmatization, the above issue affects the performance.

5.2.3 LDA

While TF-IDF assumes that each document is based on a single topic even if forum data, posts and threads may discuss several topics, LDA [23] assumes each document is built from a number of topics, with one major topic, by learning a distribution of terms in topics. LDA is a generalized Probabilistic Latent Semantic Analysis (PLSA) [59] by adding a Dirichlet prior distribution over document's topic and topic word distributions. This method requires finding a suitable tokenisation approach and

representation of a document similar to TF-IDF, and this is more computationally than TF-IDF.

5.2.4 Trending Topic Techniques

TF-IDF and LDA are both widely used, however, they have limitations and improved models have been proposed. Common NLP methods for detecting trending topics on Twitter were studied by Aiello et al. [6]. According to their work, n-gram co-occurrence (i.e. group of words typically appearing in the same document), and $DF - IDF_t$ topic ranking which is an adaption of TF-IDF to search common topics unique to a given time window in comparison to prior time window performs the best. In their approach, they also boosted the score of proper nouns and found that these are useful keywords for trending topics.

The previous work has focused on static snapshots of events, on the other hand, temporal analysis to identify both peaky and persistent topics are studied by Shamma et al. [140]. They used normalised term frequency, with the number of tweets containing the word instead of the number of times a word is used, and the peaks look at windows particular to an exact slot of time. Persistence looks at peaks of normalised term frequency.

5.2.5 Distributed Representations of Topics

A semantic space is considered as a spatial representation in which distance represents semantic association [54]. Semantic embedding of words has received much attention. For instance, word2vec is a model to generate distributed word vectors, and has been shown to capture syntactic and semantic regularities of language [97, 99].

The doc2vec model can learn document and word vectors jointly embedded in the same space, or improve the quality of the learned document vectors using pre-trained

word vectors [79]. These document and word vectors that are jointly embedded, are learned as document vectors that are close to semantically similar word vectors. This nature can be used to find which words are most similar to a document, or most representative of a document. The paragraph or document vector acts as a memory of the topic of the document according to [80]. Therefore, the most similar word vectors to a document are more likely the representative of the document's topic. Since distance in the embedded space scales semantic similarity between documents and words, this joint document and word embedding is a semantic embedding.

The top2vec [17] is an algorithm for topic modeling and semantic search. It automatically detects topics present in text and generates jointly embedded topic, document and word vectors such that distance between them represents semantic similarity. The difference between top2vec and probabilistic generative models such as LDA is how each approach models a topic. LDA models topics as distributions of words which are used to reproduce the original document word distributions with minimal error. This causes that uninformative words which are not topical have high probabilities in the topics since they occupy a large proportion of all documents. On the other hand, a top2vec topic vector in the semantic embedding represents a prominent topic shared among documents. A topic vector's nearest words describe the topic and its surrounding documents best.

BERTopic [55] is a topic modeling technique that uses transformers and the class-based TF-IDF to create dense clusters that allow for interpretable topics whilst keeping important words in the topic descriptions. Usually, TF-IDF is applied for a set of documents when we compare the importance of words between documents. The class-based TF-IDF considers all documents in a single category (e.g., a cluster) as a single document and then applies TF-IDF. The result would be scores of importance for words within a cluster, and the more important words within a cluster would

represent the cluster's topic.

5.2.6 Cybersecurity Trending Topics

The recent study into trending topics in cybersecurity has focused on identifying new threats, using data from Social Network Service (SNS), blogs, and underground forums. The creation large-scale framework, DISCOVER, is proposed by Sapienza et al. [135], and it detects emerging threats across datasets while this depends on annotations of known keywords. Since the constantly changing lexicon, this is problematic for research in the cybersecurity field.

A tool released by Behzadan et al. [20] assists annotators in exploring Twitter data, with an annotated dataset of 21,000 tweets on cyber threats. However, this tool still requires manual identification of new terms.

Topic ranking is required to avoid overwhelming a user once a trending topic is identified. Bose et al. [24] proposed a method to detect and flag known serious threats to highlight current important topics.

There have been other approaches to detect trends on forums and marketplaces. A large topic model by Tavabi et al. [146] is used to map the evolution of different forums as they evolve.

In the cybersecurity field, changing meanings of words and evolving lexicon happens over time, and these changes should be taken into account with longitudinal topic modelling. Bhandari and Armstrong [22] explored the use of high affinity terms used by communities at subforums of Reddit, and looked at the change of the semantics of these terms. The statistical approach by Hughes et al. [64] proposed a lightweight method for identifying currently trending terms in relation to a known prior of terms, using a weighted log-odds ratio with an informative prior, and it supports analysis of linguistic change and discussion topics over time without training a topic model for

each time interval for analysis.

SYNAPSE [15] is a Twitter-based streaming threat monitor for threat detection. Its pipeline is composed of filtering the collected tweets based on the monitored infrastructure, extracting the features, and classifying the remaining tweets as either relevant or not, then it clusters the relevant tweets for presenting as IOC. This framework can collect highly relevant, timely and actionable information. However, it requires training the models to perform well.

Many works have been done to detect the changing of topics in the given time windows. However, the research to find the topics in posts in the specific term is not widely researched in the cybersecurity field.

5.3 Methodology

In this section, we present a detailed description of how to extract trending topics using Cyber Attack Relevance Scale (CARS). CARS are defined in Chapter 4. We use the same definition and best model from Chapter 4 for this task.

We introduce the TrendTopicExtractor method that takes a pre-trained CARS model and the list of posts, *pList*, and clusters the posts based on the semantic similarity, then extracts the phrases to represent each cluster. This method has three steps: (i) create semantic embedding step, (ii) clustering step, and (iii) find topic phrases step. The algorithm of the TrendTopicExtractor with CARS is shown in Algorithm 9, and more detail of each step is described in the following subsections.

5.3.1 Create Semantic Embedding

The advantage of semantic embedding is the learning of a continuous representation of topics. The documents and words that are jointly embedded in document and word vector space are represented as positions in the semantic space. Each document

Algorithm 9 TrendTopicExtractor(*pList*)

1: *model* = Load the *SentenceTransformer*
2: *semanticEmbeddings* = *model*(*pList*)
3: *umapEmbeddings* = UMAP reduces the dimension of *semanticEmbeddings*
4: *cluster* = clustering *umapEmbeddings* with HDBSCAN
5: *clusterPosts* = splitting the posts in *pList* to each cluster based on the post label from *cluster*
6: *clusterPhraseList* = Extract all noun phrases from each cluster
7: Calculate cp-TF-IDF based on *clusterPhraseList* with post weights from the CARS model.
8: **return** Result of Cluster-Phrase-TF-IDF

vector in this space is treated as representing the topic of documents [80]. Thus, the word vectors nearest to a document vector represent the most semantically descriptive of the document's topic.

In this step, we convert the documents to semantic embeddings (numerical data). Since there are many pre-trained models available, we follow the BERTopic [55] approach to use BERT for converting the documents based on the context of the word, and use the sentence-transformers package [129] in our implementation. Since the transformer models have a limitation of tokens, we split the documents into paragraphs in the case for the large documents to fit the limit.

5.3.2 Clustering

In this step, we process that the given posts with similar topics are clustered together for finding the topics within these clusters. To do this process, it requires to reduce the dimensionality of the semantic embeddings since many clustering algorithms cannot handle high dimensionality well.

Since there are two main problems that are introduced by the "curse of dimensionality" which results from the high dimensional document vectors; sparse document vectors and high computational cost [88], dimension reduction on the document vectors is required. In order to reduce dimension on the document vectors, we use the algorithm Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) [96] since UMAP is a manifold learning technique for dimension reduction with strong theoretical foundations. There is another popular dimensional reduction technique; T-distributed Stochastic Neighbor Embedding (t-SNE) [147], however, t-SNE does not preserve global structure as well as UMAP and not process well to large datasets.

UMAP has several hyper-parameters for determining how it performs dimension reduction. There are three most important parameters, the number of nearest neighbors, the distance metric, and the embedding dimension. The number of nearest neighbors controls the balance between preserving global structure versus local structure in the low dimensional embedding. The distance metric is used to measure the distance between points in the high dimensional space. We use the cosine similarity [97, 98] that is the major distance metric for the document vectors and measures similarity of documents irrespective of their size. We follow the parameter settings of top2vec [17] and BERTopic [55], and set the number of nearest neighbors as 15, and the embedding dimension as 5 respectively.

After reducing the dimensionality of the document embeddings, we cluster the documents with Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [28, 94, 95]. HDBSCAN is used to detect the dense areas of document vectors while it handles both noise and variable density clusters [94], and assigns a label to each dense cluster of vectors and a noise label to all vectors that are not in a cluster. This is one of the benefits of using HDBSCAN. The cluster label

'-1' is the special label for noise cluster.

HDBSCAN requires a hyper-parameter, *minimum cluster size*, and this parameter is at the core of finding clusters of varying density. This parameter means the smallest size of a cluster by the algorithm. We set this parameter value as the ten percent of the given document size so that users do not need to give the value of this parameter manually.

The vectors of topic and document allow for the size of topics (clusters) to be calculated. The document vectors can be partitioned by the topic vectors, thus, each document vector belongs to its nearest topic vector. This means that each document has exactly one topic, the one which is most semantically similar to the document. The size of each topic (cluster) is measured as the number of documents that belong to it.

HDBSCAN iterates the clustering process while merging the smallest cluster into its most semantically similar cluster until the desired number of clusters are reached. This process uses a weighted arithmetic mean of the topic vector of the smallest topic (cluster) and its nearest topic vector, each weighted by their topic size (number of documents in the cluster). The topic sizes are recalculated for each topic after each merge process. This hierarchical topic reduction brings the advantage of extracting the topics which are most representative of the documents, as it biases topics with greater size.

5.3.3 Find Topic Phrases with CARS

Once the clusters are generated, we want to know what makes one cluster different from other clusters based on their content.

BERTopic [55] introduced cluster-TF-IDF (c-TF-IDF) that is a class-based variant of TF-IDF, that allows to extract what words make each cluster unique compared to

the others. However, many of potential topic keywords are noun phrases instead of words such as software name with version number, company name with its product name, and malicious application names. Thus, we extend this c-TF-IDF to accept phrases.

First, for each cluster's posts, we extract their noun phrases using a part-of-speech (PoS) tagger, and store the phrases in a list per cluster. This treats all posts in a single category (e.g., a cluster) as a single document. Then, we apply the cluster-based Phrase TF-IDF (Cluster-Phrase-TF-IDF, $cp - TF - IDF$) with CARS:

$$cp - TF - IDF_i = \frac{p_i}{tp_i} \times \log \frac{m}{\sum_j^n p_j}$$

, where the frequency of each phrase p_i is extracted for each cluster i and divided by the total number of phrases tp_i, and the total unjoined number of posts m is divided by the total frequency of phrase p across all clusters n. To consider the CARS, we define that CARS-NR has no weight and CARS-L has weight 1, then higher scale's weight increases by 100% each. Thus, we assign the weight of post k's CARS $CARS_k$ as follows: $CARS_k(\mathrm{NR}) = 0$, $CARS_k(\mathrm{L}) = 1$, $CARS_k(\mathrm{M}) = 2$, and $CARS_k(\mathrm{H}) = 3$. Then, we calculat p_i as follows.

$$p_i = \Sigma_k \text{number of } p \text{ in } post_k \text{ in } cluster_i \times CARS_k$$

Each phrase's $cp - TF - IDF$ represents a single importance value for each phrase in a cluster which can be used to create the topic of the cluster. In order to create a topic representation, we extract the top ten words or phrases per topic based on their $cp - TF - IDF$ scores since the scores are proxy of information density of topic.

5.4 Evaluation

To evaluate our approach, we use two ways; comparing the culturing results with other methods which is used to evaluate SYNAPSE system [15], and testing the

framework on real data which is used to evaluate both DISCOVER and SYNAPSE systems [135, 15]. More detail of the data we use is described in the next Data subsection. In addition, we also use the cyber attack timeline data with the attack occurring date in 2020 from Hackmageddon [114] to evaluate the extracted words and phrases that are related to the observed cyber attacks.

5.4.1 Data

For evaluating our method, we use the dataset obtained from a cyber-threat reconnaissance firm (called CYR3CON [1]), and the dataset contains the posts from real-world cyber threat conversations from 185 English Forum sites in the time range from January 1st, 2020 to December 31st, 2020. These posts contain at least one of the keywords under the three cyber attack type categories; Malware, Phishing, and Denial-of-Service. Malware category has the keywords; "malware", "ransomware", "spyware", "Drive-by attack", "Trojan Horses", "Macro viruses", "File infectors", "System or boot-record infectors", "Polymorphic viruses", 'Stealth viruses", "Trojans", "Logic bombs", "Worms", "Droppers", 'Adware', "Malvertising", "RAT", "Remote Access Trojan", "Fileless Malware", "Rootkits", 'Keyloggers', "Bots", and "Mobile Malware". Phishing category has the keywords; "phishing", "Spear Phishing", and "Whale Phishing". Denial-of-Service category has the keywords; "Denial-of-Service", "DoS", ''Distributed-denial-of-service", "DDoS", "TCP SYN flood attack", "Teardrop attack", "Smurf attack", "Ping of death attack", and "Botnets". The statistics of each category is shown in TABLE 5.1.

We also collect the posts from six English hacker forums from March 1st, 2021 to March 31st, 2021 for additional evaluation. The posts are split by week window start from March 1st. There are five week windows in this dataset. We call this data as

[1]https://www.cyr3con.ai

Category	# of Posts	# of sites
Malware	19,384	165
Phishing	3,326	82
Denial-of-Service	2,402	131
TOTAL	25,112	185

Table 5.1: The Statistics of the Data We Use. It Shows the Number of Posts and the Number of Sites for Each Category.

Week	1	2	3	4	5
# of Posts	2344	2644	2965	2620	3679

Table 5.2: The Statistics of the March 2021 Data. The Posts from Six English Hacker Forums During the Term from March 1st, 2021 to March 31st, 2021.

March 2021 data. Table 5.2 shows the statistics of this dataset.

We avoid publishing details that could identify individuals, including usernames, original post contents and the site names.

5.4.2 Preprocessing Data

In the preprocessing stage, we split the posts of each category by posted date and group the posts by subgroup of weeks. This process helps to visualise the weekly post trends of each category. Figure 5.1, 5.2, and 5.3 show the post trends of each category. In each category, there are some spike weeks when the significant number of posts are submitted in the week.

5.4.3 Results: Method Evaluation

Since top2vec and BERTopic automatically decide the size of clusters or they fix the minimum number of posts per cluster in their methods, we compare our methods' cluster sizes with them.

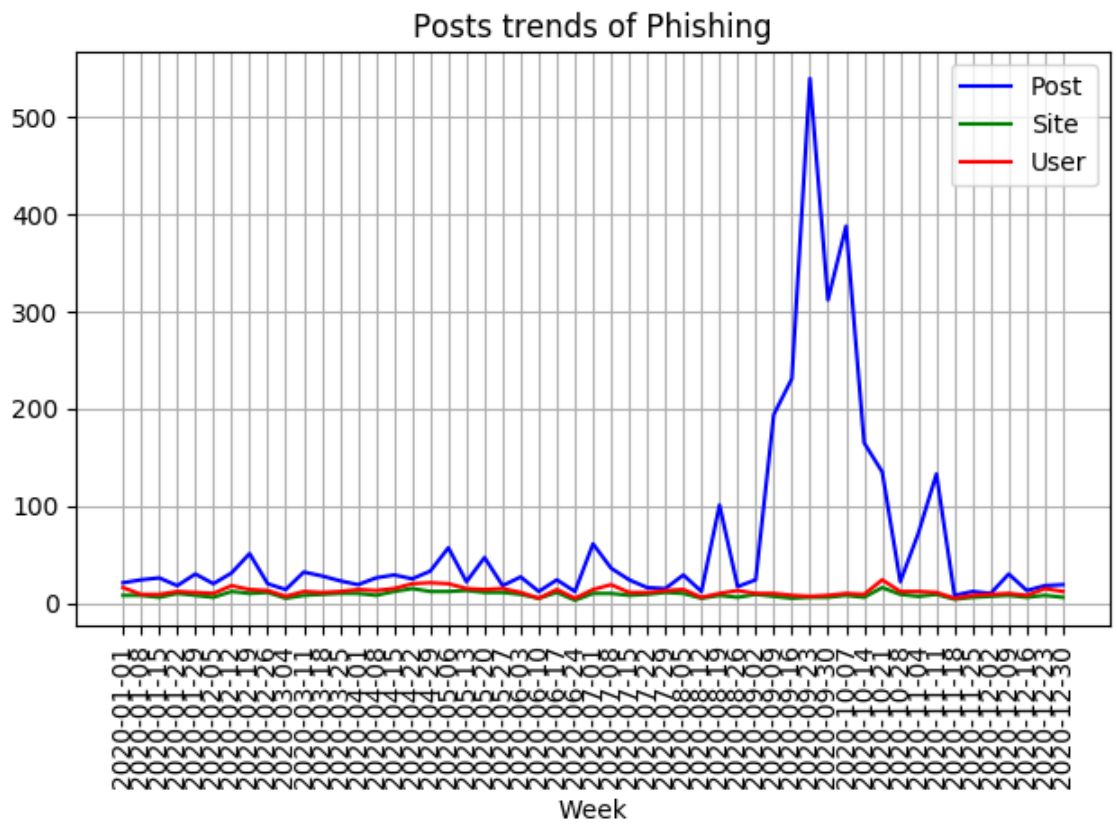

Figure 5.2: The Post Trends of the Phishing Category. It Shows the Changes of the Number of Posts, Unique Sites, and Unique Users per Week. There Are a Significant Number of Posts in the Week of September 23rd, 2020 and October 7th, 2020.

Figure 5.4 shows the number of clusters of Malware related posts by three methods per week. Most of the weeks have a similar number of clusters by three methods. The weeks of April 15th, 2020 and April 22nd, 2020 have more clusters in the top2vec method compared to the other methods. In contrast, the week of September 23rd, 2020 has more clusters in our method.

Figure 5.5 shows the number of clusters of Phishing related posts by three methods per week. Since top2vec does not work if the number of documents are small (it does not work if the number of posts are less than 100 in our case), we could not get the results of top2vec most of the weeks. The weeks of September 23rd, 2020 and October

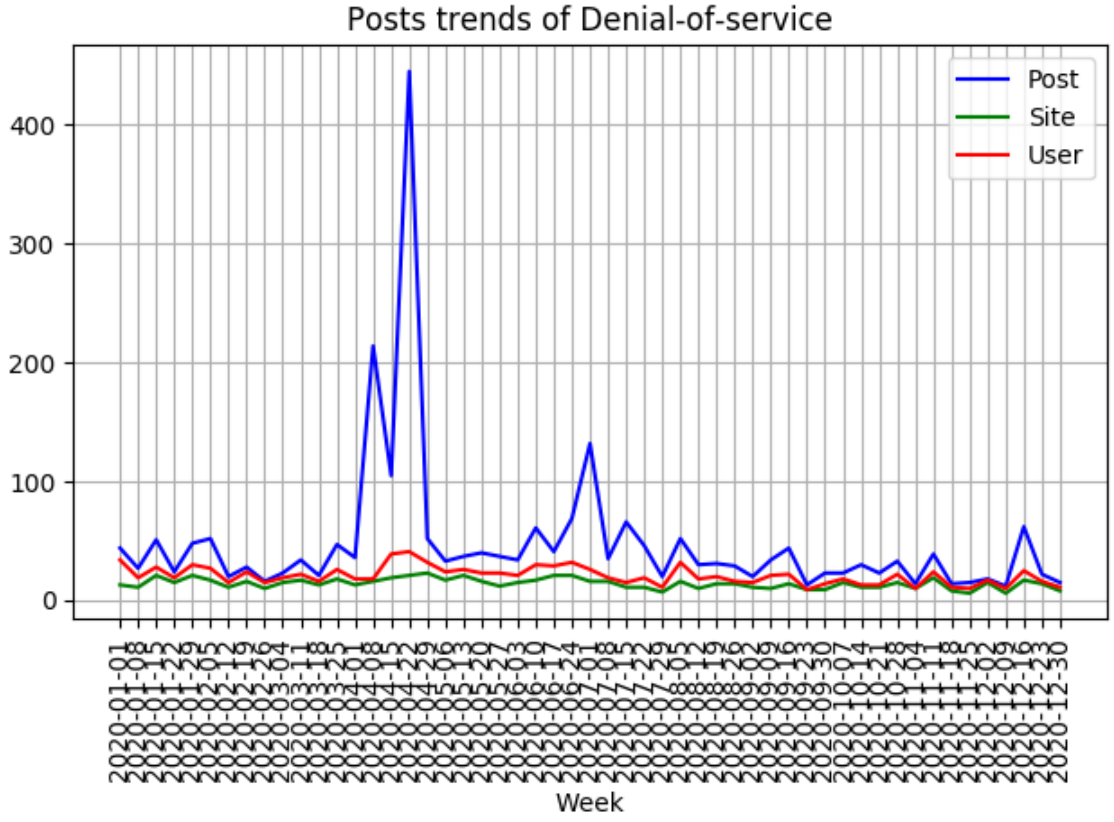

Figure 5.3: The Post Trends of Denial of Service Category. It Shows the Changes of the Number of Posts, Unique Sites, and Unique Users per Week. There Are a Significant Number of Posts in the Week of April 8th, 2020 and April 22nd, 2020.

7th, 2020 have the double number of clusters in top2vec and our method compared to BERTopic. The week of September 30th, 2020 has double the number of clusters in our method compared to the other methods.

Figure 5.6 shows the number of clusters of Denial-of-Service related posts by three methods per week. As we mentioned in the Phishing case, top2vec does not work the most of the weeks in this case since the number of posts in the most of the weeks is less than 100. It seems that our method generates more clusters than BERTopic.

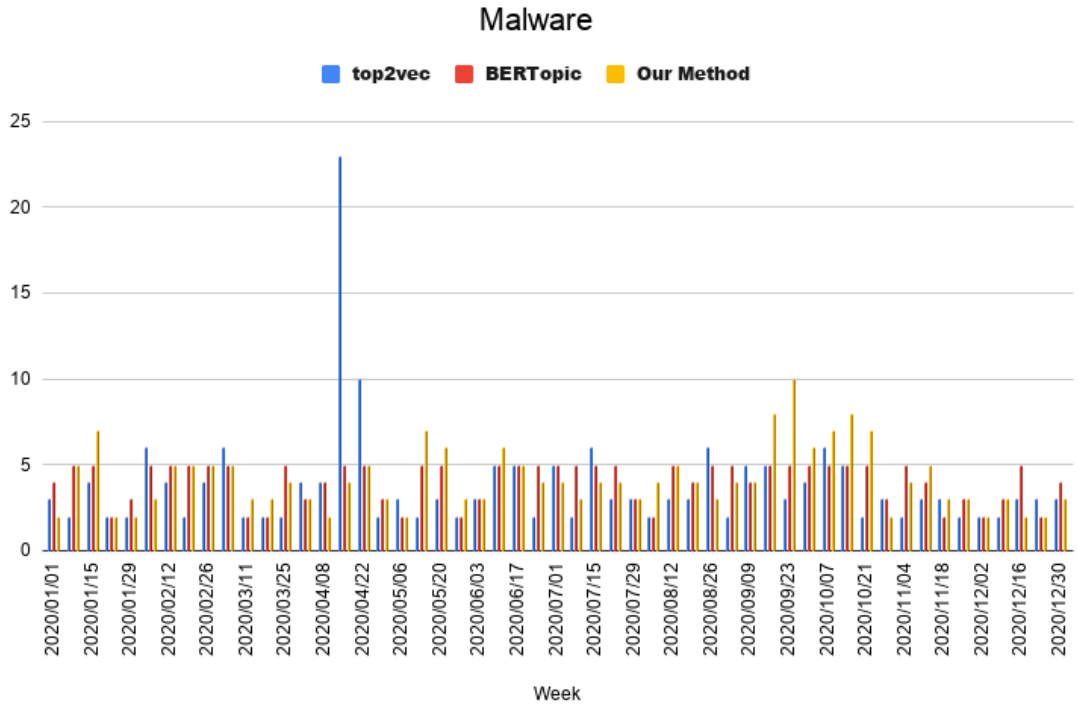

Figure 5.4: The Number of Clusters of Malware Related Posts by Three Methods per Week.

5.4.4 Results: Topic Analysis

To evaluate the extracted topic words and phrases, we check with Hackmageddon's cyber attack timeline in 2020 [114]. According to the semi-monthly timeline reports from Hackmageddon from January 2020 to July 2022, 173 cyber attacks or incidents are reported in 2020 and known the dates of occurrence. There are 70 Malware related, 48 Phishing related, and 4 Denial-of-Service related attacks or incidents occurred in 2020 respectively. We checked the description of each attack, and found that 33 out of 70 in Malware, 28 out of 48 in Phishing, and 4 out 4 in Denial-of-Service attacks have the details of the attacks such as the target names, industrial category, damage, and used tools. This information is useful to compare with the extracted topics to

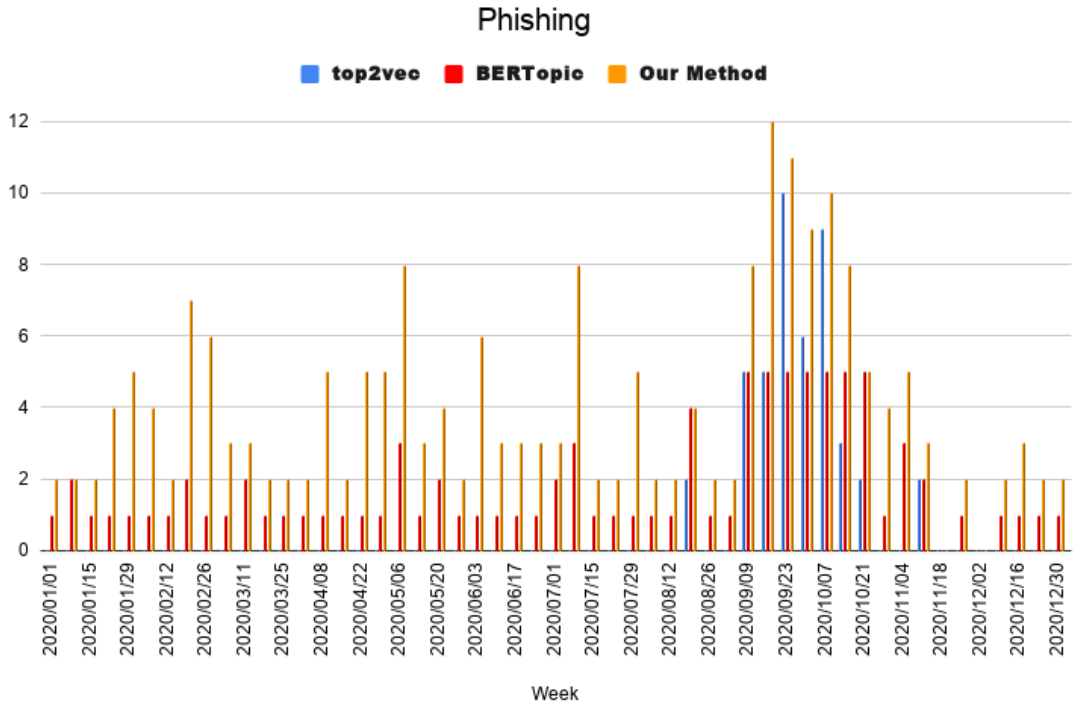

Figure 5.5: The Number of Clusters of Phishing Related Posts by Three Methods per Week.

see if the methods can find the attack related topics prior weeks of the attacks.

Figure 5.7 shows the number of incidents occurring per week in 2020 based on Hackmageddon's cyber attack timelines.

We compared our proposed methods, cluster-phrase-TF-IDF (cp-TF-IDF) and cp-TF-IDF with CARS, to the existing methods, top2vec and BERTopic with the number and percentage of attacks that the methods find the related topics prior to the attacks happened weeks. TABLE 5.3 shows the statistics of each method and each attack type. The cp-TF-IDF with CRS shows the highest percentage to find the topics related to the cyber attacks prior to the attack happening weeks in three different attack types. The cp-TF-IDF method could not find the exact topics of the attacks in Malware and Denial-of-Service attacks. This method extracts many

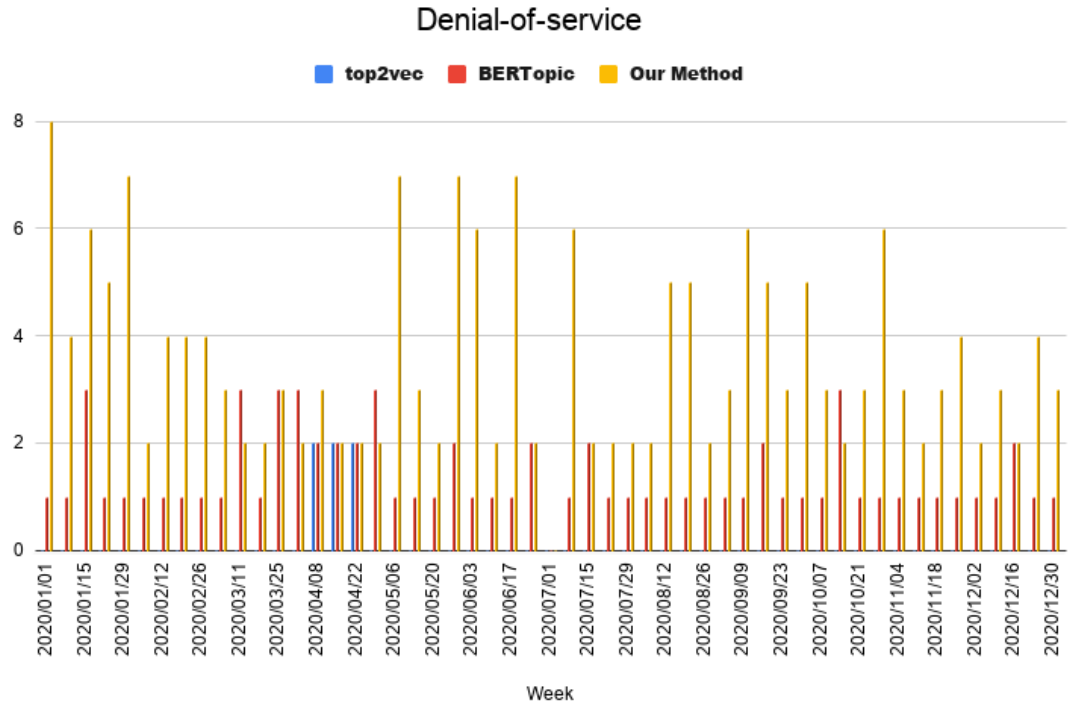

Figure 5.6: The Number of Clusters of Denial-of-Service Related Posts by Three Methods per Week.

phrases, however, most of them are not specific to lead to the attacks such as "ransomware", "cyber criminals" and "private account", and it detects six attacks prior to the attack happened weeks in Phishing attack. The top2vec method extracted several topic words linked to the attacks prior to the attacks. However, some important keys in some attacks are phrases such as "personal information", "data breach", and "sensitive data", and the extracted topic words are sometimes hard to link the attacks unless phrase words are appeared in one cluster of topic words such as "personal" and "content" or "information". BERTopic has a smaller number of clusters compared to top2vec method, and it did not extract any attack related topic words prior to the attacks.

In addition, we also compared the extracted topics by each method. Overall,

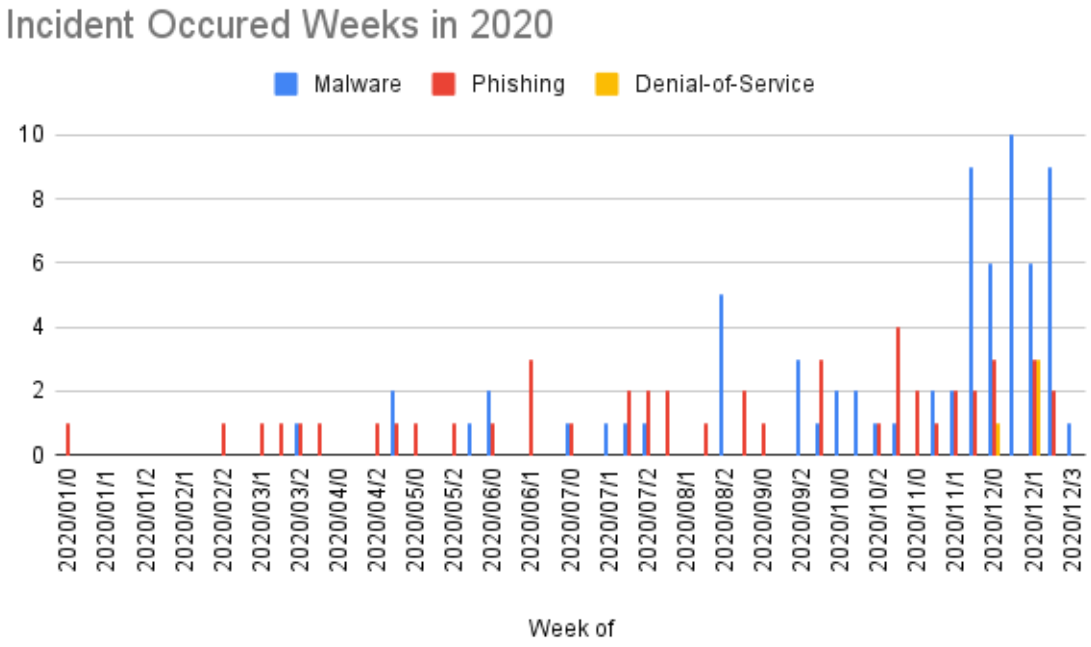

Figure 5.7: The Number of Cyber Attacks Occuring per Week in 2020.

		top2Vec	BERTopic	cp-TF-IDF	cp-TF-IDF with CARS
Malware	# of Attacks Predicted	9	7	0	13
	% of entire Attacks	12.86%	10.00%	0.00%	**18.57%**
	% of known Target/Method Attacks	27.27%	21.21%	0.00%	**39.39%**
Phishing	# of Attacks Predicted	8	0	6	19
	% of entire Attacks	16.67%	0.00%	12.50%	**39.58%**
	% of known Target/Method Attacks	28.57%	0.00%	21.43%	**67.86%**
Denial-of-Service	# of Attacks Predicted	0	0	0	1
	% of entire Attacks	**25.00%**	0.00%	0.00%	**25.00%**
	% of known Target/Method Attacks	**25.00%**	0.00%	0.00%	**25.00%**

Table 5.3: The Number and Percentage of Attacks That the Methods Find the Related Topics Prior to the Attacks Happened Weeks in Each Attack Type.

the top2vec and BERTopic can only extract words, and the most of them are basic keywords of each topic in Malware, Phishing, and Denial-of-Service. For instance, the top2vec and BERTopic extracted the common keywords of each attack type such as “ransomware”, “system”, and “hacking” in Malware, “account”, “password”, and “hijacking” in Phishing, and “bot”, “injection”, and “attacker” in Denial of Service respectively. On the other hand, cp-TF-IDF and cp-TF-IDF with CARS methods extract more detailed phrases such as “Proton RAT”, “Windows 10 version”, and “the malicious bootloader” in Malware, “your account settings”, “control system networks”, and “remote access” in Phishing, and “exchange server”, “DDoS methods”, and “DDoS attacks” in Denial-of-Service respectively.

We deeply analyze the weeks of the highest number posted and the most attacks and incidents occurred in each attack type. In Malware, the week of April 15th, 2020 is the highest number of posts observed, and the week of December 9th, 2020 is the most malware incident occurred in 2020. In Phishing, the week of September 23rd is the highest number of posts observed, and the week of October 28th, 2020 is the most phishing incident occurred in 2020. In Denial-of-Service, the week of April 22nd, 2020 is the highest number of posts observed, and the week of December 16th, 2020 is the most denial-of-service incident occurred in 2020. We extract the top five words or phrases from the three largest clusters from each method.

Malware: The highest number of posts related to Malware is observed in the week of April 15th, 2020, and we extracted several top words or phrases from each method. In the top2vec, “leecher” and “joker” appear top two words in cluster 1, and they are a part of the cracking tool, “Joker Combo Leecher”. “ewido” and “ikarust3” in the cluster 2 are the name of security software company and security software. “nitroflare” and “rapidgator” in cluster 3 are the names of file sharing sites. In the BERTopic, “joker” is also listed in a cluster. Other clusters have the

names of security software and companies. “antivirus” appears in multiple clusters’ top five words. In cp-TF-IDF method, game name “Counter-Strike” and failure of cheat detecting system “False VAC”, specific cyber attack name “ransomware” and OS name “Windows”, and RAT with version number “RAT v3 very simple rat” are listed in top five phrases in the clusters. In cp-TF-IDF with CARS method, “Universal Combo Software”, “Joker Combo Leecher”, and “combo generation Joker RAT”, “Wolfgang Amadeus Mozart”, and “the malware” are listed in top five phrases in the clusters. Joker combo Leecher is a malicious software. Mozart is normally known as a famous composer, however, this case is a name of malicious software.

The most Malware attacks and incidents reported week in 2020 is the week of December 9th. In the top2vec, the uploader site names and several company names are in the top words in the top two clusters such as “rapidgator”, “uploadgig”, “mercedes”, and “nitroflare”. In the BERTopic, “windows”, “ransomware”, and “threats” are in the top words in a top cluster. In cp-TF-IDF method, there are several cybersecurity related company names in the top phrases/words such as “Acronis” and “Avast” in a top cluster, and there are no specific malware related words in any cluster. In cp-TF-IDF with CARS method, there are not only the cybersecurity related company names mentioned cp-TF-IDF method but also several malicious software names and target such as “Qakbot”, “the ransomware” and “Ngrok” in the top words and phrases in a top cluster.

Phishing: The highest number of posts related to Phishing is observed in the week of September 23rd, 2020, and we extracted several top words or phrases from each method. In the top2vec, cyber attack keyword “spam” and message service “sms”, dating site name “tinder” and password cracker “ophcrack”, Android file format “apk” and cyber attack keyword “bleach” are in the top five words in three clusters. In the BERTopic, phishing tool “lockphish” with cyber attack keyword

"hack", cyber attack keyword "hacking" and product name "android", and cyber attack keyword "hacking", product name "iphone" and OS name "iOS" are mentioned in the top five words in the three clusters. In cp-TF-IDF method, "spam messages" and "spammers", "the remove_lock_root module" and "the Android phone", "Apkbleach tool", and "Cydia", "jailbreak" and "iOS hacking" are mentioned in the top five phrases in the four clusters. In cp-TF-IDF with CARS method, there are several cybersecurity company and tool names extracted in the top phrases in the clusters such as "HackerOne", "Synack", "OSINT Framework", "IP addresses", and "critical vulnerabilities".

The most Phishing attacks and incidents reported week in 2020 is the week of October 28th. In the top2vec, there are several cyber attack related keywords such as "hackers", "payload", and "metasploit" in the largest cluster. In the BERTopic, there is only one cluster generated and the cluster has "phishing", "attack" and "hacking" in the top words. In cp-TF-IDF method, there are several phishing attack related phrases such as "pro ATTACKER", "your Phishing Website", "Payload Bind Shell", and "an Advance Ethical Hacking Machine Instagram Hacking" extracted from the top cluster. In cp-TF-IDF with CARS method, there are more detailed phishing attack related phrases such as "your Phishing Website", "various CLI commands", "Metasploit Framework" and "contentspoiler" in one of the top clusters.

Denial-of-Service: The highest number of posts related to Denial-of-Service is observed in the week of April 22nd, 2020, and we extracted several top words or phrases from each method. In the top2vec, there is no cyber security related word found. In the BERTopic, "jspy", "cracked" and "trojans" are mentioned in the top five words in a cluster. In cp-TF-IDF method, "multi os jSpy v0.31" and "(RAT", and "possible attack", "tcp" and "SYN_RECV" are mentioned in the top five phrases in the two clusters. In cp-TF-IDF with CARS method, "Cisco IP Phone public PoC

- very easy DoS", "BlackNET Advanced", "BlackNET 3.0 botnet free download", "Slowloris", and "SpyEye" are in the top clusters.

The most Denial-of-Service attacks and incidents reported week in 2020 is the week of December 9th. In the top2vec, there are several Denial-of-Service related keywords from a top cluster such as "gpg", "privacy", "hsdir", and "trojan". In the BERTopic, there is only one cluster extracted and "security" and "web" are the only keywords related to Denial-of-Service attack. In cp-TF-IDF method, there are several specific ways or targets such as "document Login Spoofing", "another Debian based Linux distribution", and "Bank Wire" extracted from a top cluster. In cp-TF-IDF with CARS method, there are several related phrases and keywords in a couple of top clusters such as "Exchange Server", "your hidden service", "cybercriminals", and "1Gbps unmetered* 100Gbps DDoS".

The top2vec and BERTopic methods extracted some useful keywords to represent the topics related to the attack types. However, some of the keywords are ambiguous and it is hard to determine that the keywords surely represent the topic of the attack types. On the other hand, our cp-TF-IDF and cp-TF-IDF with CARS methods extract not only keywords but also phrases. Many phrases contain the same keywords from top2vec and BERTopic, and the phrases have more detailed information to solve the ambiguity of the keywords. For instance, 'phishing website' can specify the method of 'phishing'.

5.4.5 Result: March 2021 data

In week 1 from March 1st, 2021 to March 7th, 2021, top2vec, BERTopic, and cp-TF-IDF methods do not have the detailed topic keywords of potential attacks. For instance, they have "ransomware", "vulnerability", "hacking", and "malwarebytes", however, there is not specific name of tool and victims. On the other hand, cp-TF-

IDF with CARS method has some detailed phrases about an attack in one of the clusters; “Prism”, “PrismHR”, “the attackers”, “PEOs”, “their customers”, “small businesses”, and “victim organizations”.

In week 2 from March 8th, 2021 to March 14th, 2021, top2vec method has “ransomware” keyword in one of the clusters, and there is no additional or supporting keywords to specify the ransomware type or victims. BERTopic and cp-TF-IDF methods do not have any cyber attack related keywords or phrases in each cluster. However, cp-TF-IDF with CARS method has some detailed phrases from a cluster such as “2021 Vulnerabilities”, “AMNESIA:33”, and “a memory buffer”.

In week 3 from March 15th, 2021 to March 21th, 2021, only top2vec method has “leak”, “database”, “evileaks” and some of the paid member only website names. Other methods do not have specific keywords or phrase to link to cyber attacks.

In week 4 from March 22th, 2021 to March 28th, 2021, BERTopic, cp-TF-IDF and cp-TF-IDF with CARS methods have “DDoS” keyword. Especially, only cp-TF-IDF with CARS method has more details such as “DDoS attack”, “DDoS service”, and “remove rival corporations effectively” and same phrases in Russian.

In week 5 from March 29th, 2021 to March 31th, 2021, all four methods have “evileaks”, and “evilx”, and only cp-TF-IDF and cp-IF-IDF with CARS methods have more details such as “Daily Random Videos”, “update daily random videos”, and several uploader site names and URLs.

5.5 Discussion

In this section, we discuss the two topics; the clusters and the topic phrases which are extracted from our method.

5.5.1 Clusters

As we mention in the Result subsection, our method detects a fairly similar number of clusters compared to the top2vec and BERTopic. The top2vec, unfortunately, does not work when the number of given posts is smaller than around 100. However, when all three methods work, they cluster the given posts into a very similar number of clusters in most cases.

Figure 5.8, 5.9, and 5.10 show the UMAP reduced document vectors from the Malware posts in the week of April 15th, 2020, the Phishing posts in the week of September 23rd, 2020, and the Denial-of-Service posts in the week of April 22nd, 2020 respectively. Each colored area of points is a dense area of posts identified by HDBSCAN, the grey points are the posts that HDBSCAN has labeled as noise/outliers.

In the Malware posts in the week of April 15th, 2020, our method detects four clusters including the noise posts. Figure 5.8 shows the three clusters clearly and the noise posts are distributed at the edges of the space. This means that there are three major topics in that week.

In the Phishing posts in the week of September 23rd, 2020, our method detects 11 clusters including the noise posts. Figure 5.9 shows that there are many small clusters in space. This means that there are many topics discussed in that week.

In the Denial-of-Service in the week of April 22nd, 2020, our method detects two clusters. Figure 5.10 shows the two clusters clearly in the space. One cluster on the right is compact, in contrast, the other cluster on the left is widely spread.

We set the *minimum cluster size* of the clustering process with HDBSCAN as the ten percent of the given document size. We tested five percent and fifteen percent respectively, however, the small percentage gives more clusters and the larger percentage gives only noise labels in most cases. Thus, the ten percent of the given

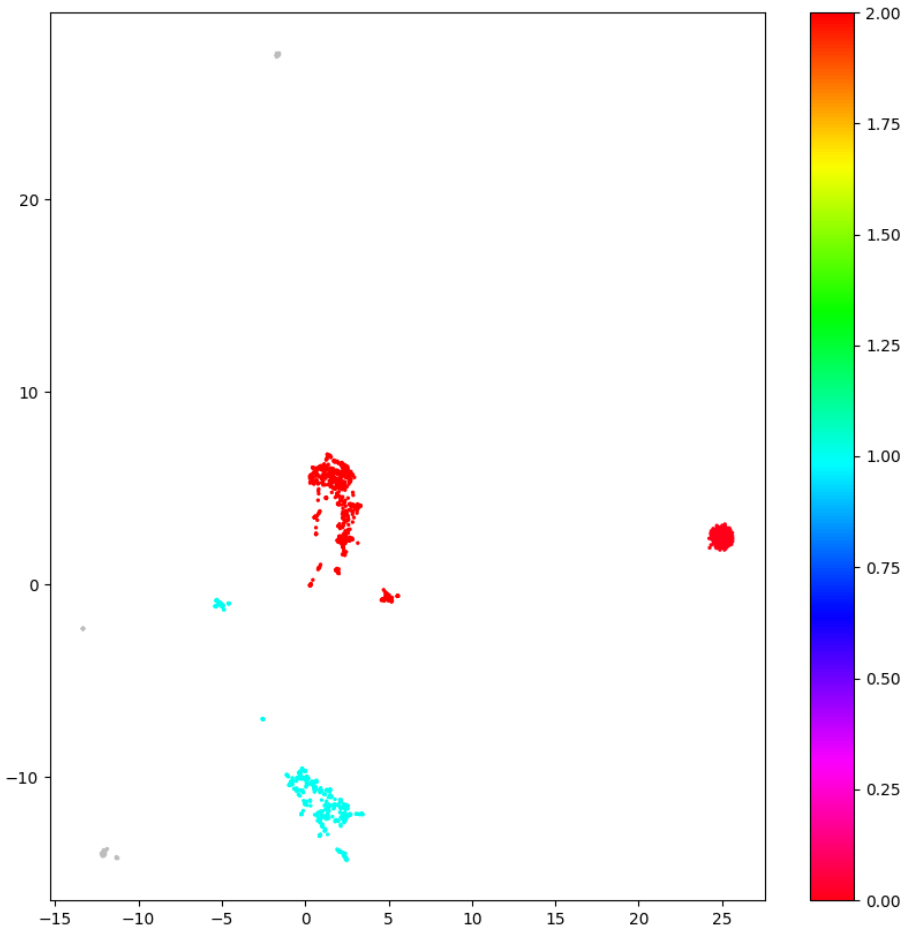

Figure 5.8: Clusters' Distribution of Malware in the Week of April 15th, 2020.

document size currently gives the best performance.

5.5.2 Topic Phrases: Case Study of Predicting Future Attacks and Incidents

TABLE 5.3 shows the number and percentage of attacks that the methods find the related topics prior to the attacks happening weeks. One of our motivations is to find

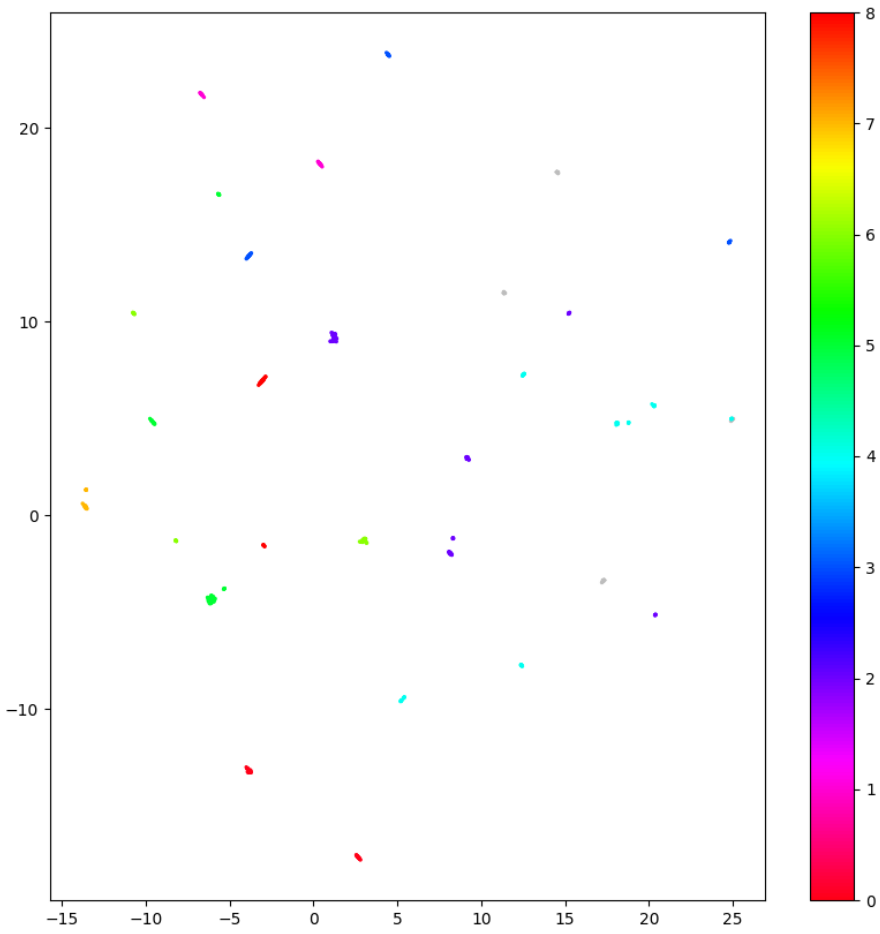

Figure 5.9: Clusters' Distribution of Phishing in the Week of September 23rd, 2020.

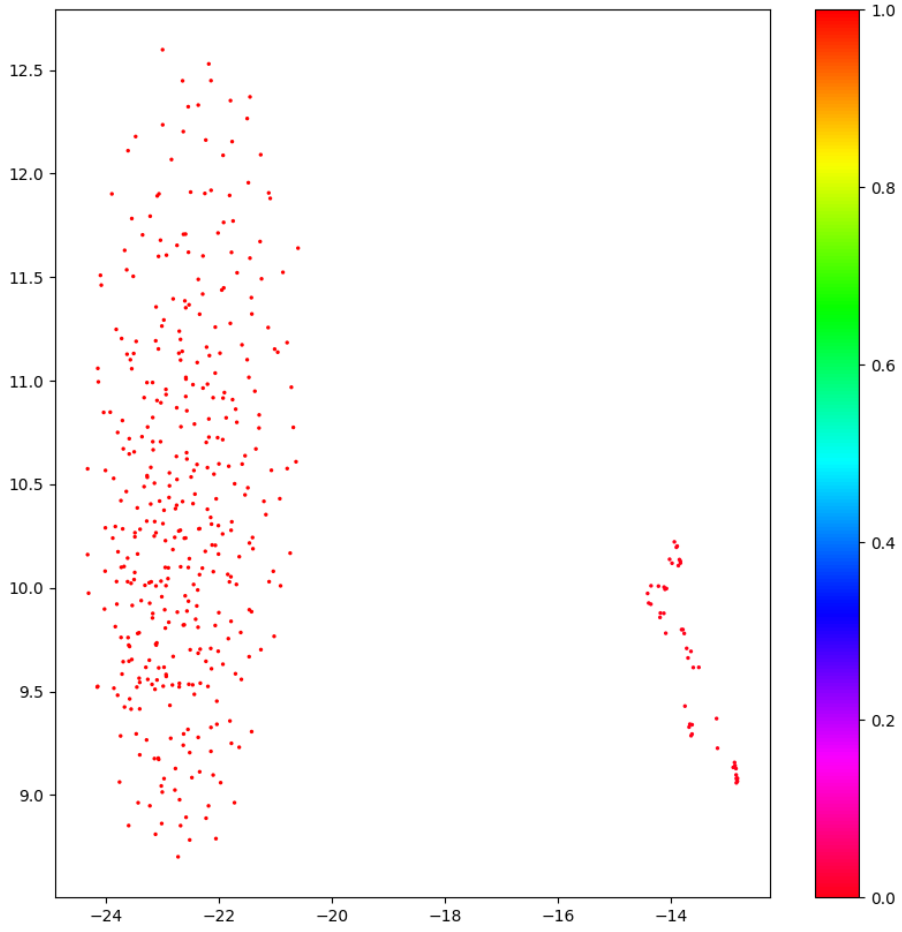

Figure 5.10: Clusters' Distribution of Denial-of-Service in the Week of April 22nd, 2020.

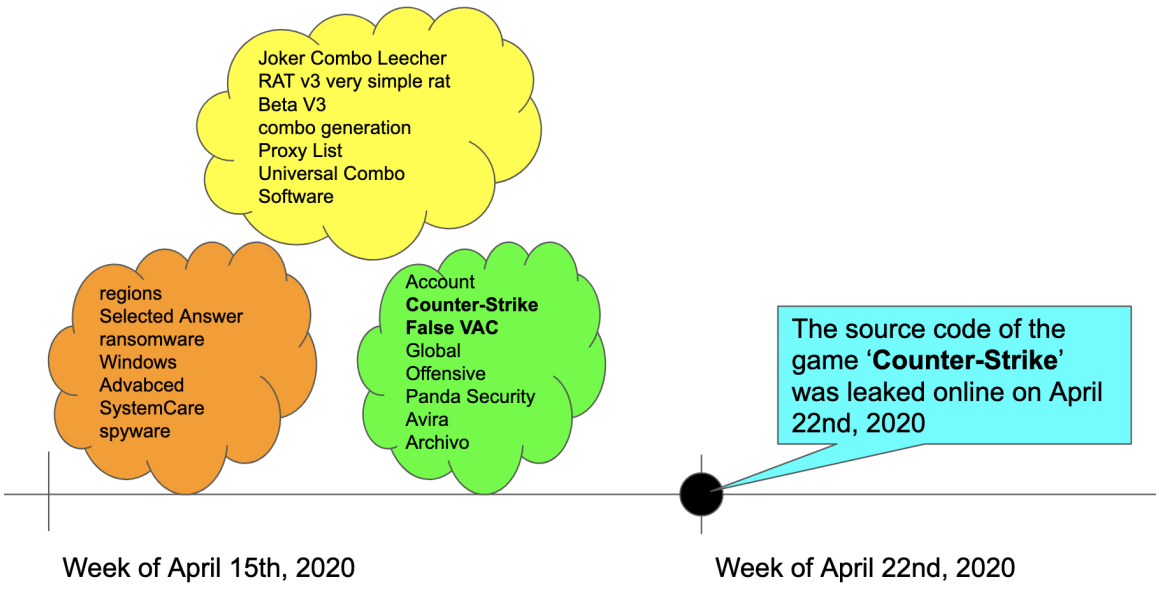

Figure 5.11: The Topic Phrases of cp-TF-IDF with CARS Method from the Week of April 15th, 2020, and Timeline of Counter-Strike Source Code Leak Incident on April 22nd, 2020.

clues of potential future cyber attacks and incidents from hacker conversations. Thus, we deeply analyze the extracted words and phrases for each week and Hackmageddon's cyber attack timeline to find any clues of early detection of attacks.

Malware: In the Malware posts from the week of April 15th, 2020, the interesting phrases from one of the clusters through our cp-TF-IDF method are "Counter-Strike" and "False VAC". According to a news article [2], the source code of the game "Counter-Strike" was leaked online April 22nd, 2020. Thus, at most a week before, our cp-TF-IDF method could extract some clues of this incident. Figure 5.11 shows the timeline of the incident and the topic phrases from cp-TF-IDF with CARS.

In addition, according to Hackmageddon's cyber attack timeline, "Conti" ransomware is used for two cyber attacks on December 22nd and 24th, 2020. Our cp-TF-IDF with CARS method extracted "Conti" and "The ransomware" from one of the top clusters in the week of December 9th, 2020. Thus, at most 13 days before

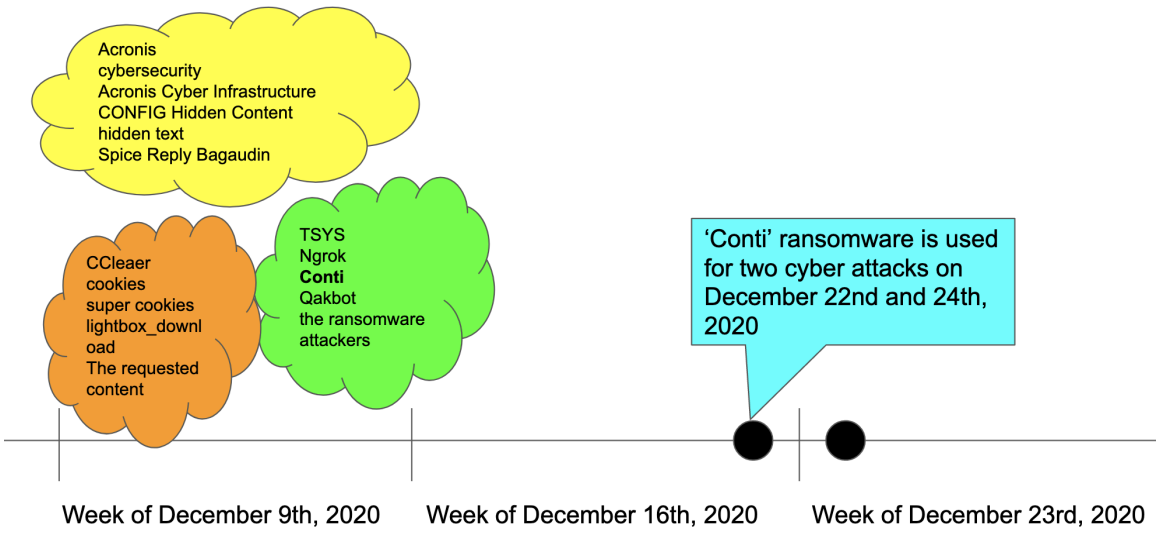

Figure 5.12: The Topic Phrases of Cp-TF-IDF with CARS Method from the Week of December 9th, 2020, and Timeline of "Conti" Ransomware Attacks on December 22nd and 24th, 2020.

the attacks, cp-TF-IDF with CARS method could extract clues. If we apply CARS to top2vec and BERTopic, both methods with CARS could also extract "conti" and "ransom" keywords in the weeks of December 9th, 2020 and December 16th, 2020. Figure 5.12 shows the timeline of the attacks and the topic phrases from cp-TF-IDF with CARS.

Phishing: In the Malware posts from the week of December 23rd, 2020, our cp-TF-IDF with CARS method extracted "Koei Tecmo", "the attack", and "personal data". According to the cyber attack timeline, "Japanese game developer Koei Tecmo discloses a data breach and takes their European and American websites offline after stolen data of 65.000 users is posted to a hacker forum" on December 20th, 2020. This attack is categorized as a Phishing attack, however, cp-TF-IDF with CARS method could find the clues immediately after the attack occurred.

Denial-of-Service: In the Denial-of-Service posts from the week of December 9th, 2020, cp-TF-IDF with CARS method extracted "malicious cyber actors", "ran-

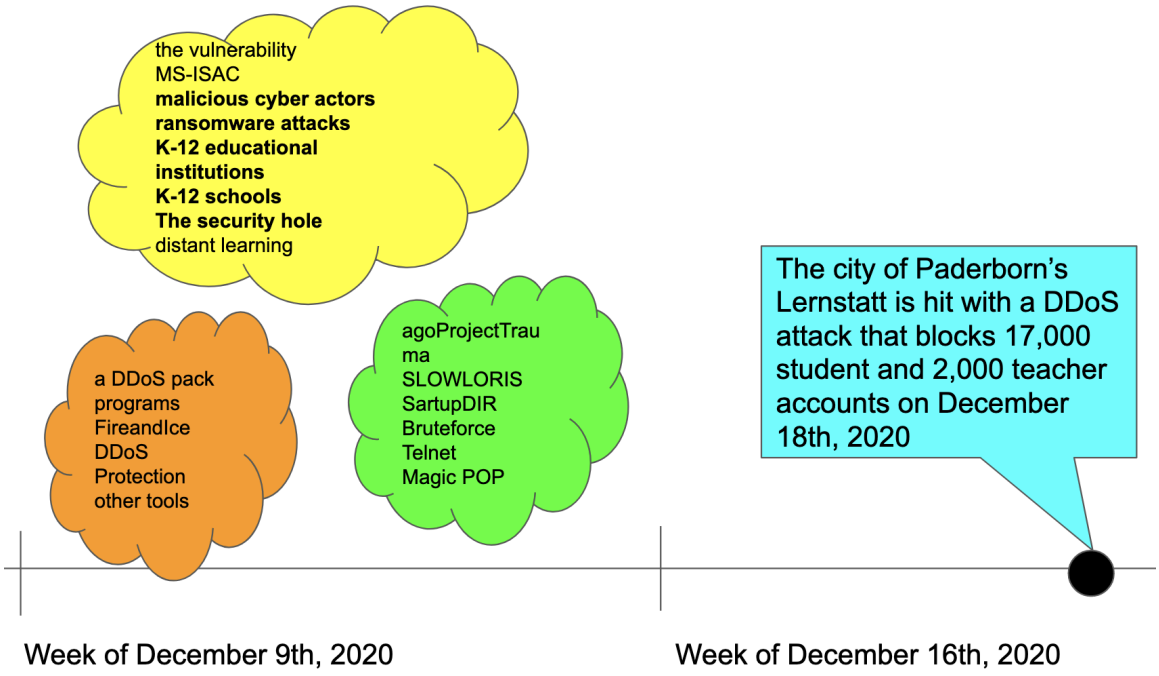

Figure 5.13: The Topic Phrases of cp-TF-IDF with CARS Method from the Week of December 9th, 2020, and Timeline of DDoS Attack Against a School Network on December 18th, 2020.

somware attacks", "K-12 educational institutions", "K-12 schools", and "security hole" from the top cluster. These phrases suggested some cyber attack against educational institutes. According to Hackmageddon's cyber attack timeline, "The city of Paderborn's Lernstatt network is hit with a DDoS attack that blocks 17,000 student and 2,000 teacher accounts" on December 18th, 2020, and this attack is reported on December 20th, 2020. Thus, our cp-TF-IDF with CARS method could get some clues of the attack at most 9 days prior to the incident occurring. Figure 5.13 shows the timeline of the incident and the topic phrases from cp-TF-IDF with CARS.

In addition, from the Denial-of-Service posts in the week of December 23rd, 2020, cp-TF-IDF and cp-TF-IDF with CARS methods extracted "the Citrix ADC", "Datagram Transport Layer Security", "message forgery", and "several targets". According to the cyber attack timeline, "Citrix confirms that an ongoing DDoS attack pattern

is affecting Citrix Application Delivery Controller (ADC) networking appliances" on December 21st, 2020 and this attack is reported on December 24th, 2020. Our methods extracted these clues in the same week of the attack reported.

March 2021 Data: Each method found some keywords potentially related to cyber attack or incident from some clusters. However, cp-TF-IDF with CARS method provided more detailed information. For instance, in week 1, "PrismHR" is in one of the clusters from cp-TF-IDF with CARS method, and the company got a cyber attack on February 24th, 2021, a week before. Thus, our method could catch the conversation of recent cyber attacks discussed by hackers right after the attack. In week 2, only cp-TF-IDF with CARS method has "AMNESIA:33" which is 33 critical vulnerabilities of Internet of Things (IoT) devices over millions of products that can be affected. "AMNESIA:33" is reported in 2020 but the same cluster has "2021 Vulnerabilities". Thus, the hackers may discuss newer vulnerabilities related to the set of the vulnerabilities. In addition, in the week 4, BERTopic, cp-TF-IDF and cp-TF-IDF with CARS methods have "DDoS" keywords, and only cp-TF-IDF with CARS method have more related phrases such as "DDoS attack", "DDoS service", and "remove rival corporation effectively" in both English and Russian. These phrases give us that someone wants to promote his/her/their DDoS service to the other users. Therefore, our cp-TF-IDF with CARS method can not only predict potential future attacks but also find the recent attacks that hackers discussed and promoting some cyber attack services.

5.6 Conclusion

Detecting the trending topics that are discussed in the specific time frame in the cybersecurity and cyber crime related forums is an important task to gain information about existing and emerging cyber threats by cyber criminals, and can help to prevent

security breaches in cyber space. Topic modeling is used for finding latent semantic structure as topics in a collection of documents. In this work, we introduce Cyber Attack Relevance Scale (CARS) and this scale helps to filter the relevance of posts to cyber attack and incidents. Then, we presented new methods for topic modeling using distributed representations of forum posts and words, and clusters the semantic embeddings of the posts, then it extracts the topic phrases from each cluster with the Cluster-Phrase-TF-IDF (cp-TF-IDF and cp-TF-IDF with CARS). Our methods, TrendTopicExtractor (cp-TF-IDF and cp-TF-IDF with CARS), do not require any keyword dictionary to process, and TrendTopicExtractor can extract several useful topic phrases to represent the topic of the clusters.

In the experimental evaluation, we compared our methods, cp-TF-IDF and cp-TF-IDF with CARS, with the top2vec and BERTopic on the real forum posts about Malware, Phishing and Denial-of-Service categories. The number of clusters by our methods is similar to the number of clusters by the other methods, and our methods can provide useful topic phrases for each cluster. Some of the topic phrases in a cluster from cp-TF-IDF and cp-TF-IDF with CARS are indicating some cyber attack or incident that happened later weeks. Especially, cp-TF-IDF with CARS method reached the highest percentage of extracting the topic phrases that are related to the attacks prior to the attack. Thus, our methods can detect some clues for ongoing or future cyber attacks and incidents from hacker forums' posts. However, extracting noun phrases has the space to improve since it contains a part of source code and incomplete phrases.

Named Entity Recognition (NER) is one of the other approaches instead of extracting noun phrases. There have been recent studies in using NER on noisy text [40, 5]. Kashihara et al. [70] proposed a bootstrapping method to generate the annotated training corpus for cybersecurity NER model with a small keyword

dictionary. In the future work, we will try NER to extract the important entities from the posts for identifying the topic words and phrases.

Additionally, we evaluate the proposed methods with pre-processed hacker forums' posts based on the given keywords. However, we found that some extracted topic phrases in a cyber attack type are linked to the other type such as malware names extracted in Phishing attack topic phrases. Thus, it will be better not to categorize the posts based on the attack types. Instead, in the future work, we will apply the CARS model to categorize the forum posts based on the cyber attack relevance. Then, we cluster only the cyber attack related posts since many hacker forums have not only cyber attack or incident topics but also a variety of non cybersecurity topics such as illegal drugs and pirate goods.

Chapter 6

CONCLUSION AND FUTURE WORK

With recent trends indicating cyber crimes are increasing in frequency and cost to business, it is imperative to develop new methods that leverage data-rich hacker forums to assist in combating ever evolving cyber threats. The hacker forums users often use jargon and previously unseen tool names often with vary different meanings to those previously used. The traditional dictionary and pattern matching methods cannot predict them correctly. In addition, defining interactions within hacker forums is critical as it facilitates identifying highly skilled users, which can improve prediction of novel threats and future cyber attacks. However, many hacker forums are unstructured and it is hard to see the direct interaction within the users. Furthermore, understanding the trend topics of hacker forums is an important role (function) when predicting future cyber attacks. However, many forums discussed not only cyber attacks but also various unrelated topics. Thus, more flexible ways to identify the cybersecurity related named entities, building the social network from unstructured forums for social network analysis considering user interactions, and filtering and extracting cyber attack related trending topics from the forums, are needed to enhance cyber defense and cyber attack prediction systems.

6.1 Summary of the Contributions

This dissertation proposes various methods using natural language processing techniques to empower cyber defense, demonstrating how semantic information from online conversations in hacker forums is valuable to enhance the current systems.

In Chapter 2, we build two different methods for named entity recognition (NER)

in cybersecurity domain; semi-automatic corpus generation method with small dictionary, and Unified Text-to-Text CyberSecurity (UTS) model in cybersecurity domain. We develop sentence category classifiers (SentCat and Category Classifier) for ambiguous keywords that can assign multiple categories in the annotation process that calculate the semantic similarity of each category of keywords in the annotating sentence to find the most suitable category of the keyword. This approach requires a smaller dictionary size to annotate. The experimental result shows that 70% of the original dictionary size can generate the annotated dataset that can perform very similar F1 score to the NER model trained with a fully annotated dataset. In addition, this method is semi-automatic that means the initial dictionary is the only part to require human efforts instead of requiring human experts to annotate all documents one by one. This significantly reduce the human efforts and makes it easier to generate new NER corpus. On the other hand, the UTS model is the combination of multi-task model and prompt-based approach that use task control codes as prompt-prefix for training the models in a multi-task setting. Since there is the limitation of publicly available datasets in the cybersecurity domain, we trained with four NER datasets in the cybersecurity domain. The results show that both BART and T5 with UTS improves the performance comparing to the BART and T5 trained with only one dataset, and T5 with UTS performs best in the methods we compared. Multi-task model is widely researched in the different domains, however, we proved that the multi-task model works in the cybersecurity domain as well.

In Chapter 3, we introduced new methods; Next Paragraph Prediction (NPP), Next Paragraph Prediction with instructional prompts (NPP-IP), and Flow Structure (FS) to predict the thread structures from unstructured forums considering the interaction of users. NPP is extended from BERT's Next Sentence Prediction to accept multiple sentences (paragraph) as inputs and the first time to predict the thread

structure and user interactions considering post contents. The performance of NPP is better than the existing assumption ways. In addition, we apply instructional prompts to the training for the NPP model to improve the performance. This is the first time instructional prompts have been applied in the cybersecurity domain, and the performance improved average 4% in F1 scores. Furthermore, we developed and applied the Flow Structure approach we developed to predict the thread structures. It is necessary to adapt the new thread structure predicting task, however, the performance shows the highest F1 scores in Reddit dataset. Our methods perform better than the existing assumption methods, and we believe that our methods will improve the way the current Social Network Analysis works in the hacker forums to find key users.

In Chapter 4, we define Cyber Attack Relevance Scale (CARS) to scale hacker forums posts for four categories; No Related, Low, Medium, and High. This scale helps to filter or weight the forum posts to find cyber attack related topics. We combined CARS model and NPP model from Chapter 3 (NPP-CARS) to generate social networks from unstructured hacker forums, and extract the key users who are the core of the conversations about cyber attack related topics. The experimental results show that NPP-CARS approach can extract more potentially useful users who post many cybersecurity related topics than other existing methods. Many hacker forums have not only cyber attack related topics but also other topics such as game, illegal drug, politics, and pirate items, and CARS can filter these non cybersecurity related topics out from the results.

In Chapter 5, we build a system to detect cybersecurity related trending topic phrases from hacker forums. The existing topic models, top2vec and BERTopic, have an issue in that they can find only the topic words, however, the cybersecurity field has phrases to represent the product names, corporation names, and tools names.

In addition, as we mentioned in before, many hacker forums have not only cyber security related topics but also other topics. Thus, we introduce a new method, TrendTopicExtractor, to extract the topic phrases through Cluster-Phrase-TF-IDF (cp-TF-IDF), and combine CARS model from Chapter 4 to weight the cyber attack related posts (cp-TF-IDF with CARS). The evaluation result shows that cp-TF-IDF with CARS gets the highest number of cyber attacks in Malware, Phishing and Denial-of-Service in 2020 through extracting the topic phrases that link to the cyber attacks prior to the attacks. This means that our system can reinforce the cyber attack predicting system through finding the cyber attack related topics a hacker discussed in the specific time frame.

By proposing the models detailed in this dissertation, we provide methods for the extracting of cybersecurity related named entities, building the social structure from unstructured forums, finding key users who are highly skilled and knowledgeable about cyber attacks from noisy forum posts, and finding trending topics in hacker forums to predict the future cyber attacks.

6.2 Future Directions

The work conducted in this dissertation will be able to extend in many directions in order to enhance cyber threat intelligence. Some potential research areas are discussed in the following.

- Apply NER model to trending topic phrases. We extracted all noun phrases from posts to extract trending topic phrases in the current approach, however, they have many general term phrases or words in the results. Our NER model [70] for cybersecurity will extract cybersecurity related entities (noun or noun phrases), and using them will improve the quality of extracted topic phrases.

- Vulnerability Prediction. There are some systems to predict the vulnerability exploitation [11, 13], and one of the features from hacker posts about the vulnerability. However, they only use the posts containing CVE ID. Since CVE ID is assigned by a CVE Numbering Authority, new vulnerabilities found by hackers may not have CVE IDs. In addition, some of the posts mentioned vulnerability, however they sometimes do not have CVE ID in the posts. We can extend our NER model to detect potential vulnerability related keywords and make some mapping to the existing CVE ID or create a different way to warn defenders to the new non CVE ID assigned vulnerabilities.

- Forecasting Cyber Attacks. DISCOVER [135] showed that multiple online data sources such as social media, blogs, and forums provides the terms related to emerging cyber threats. In addition, SYNAPSE [15] is a Twitter-based streaming threat monitor, and it filters and clusters for the cyber threat relevant tweets for presenting as IOC. Furthermore, our trend topic phrase detecting method showed that the approach can find extract the topic phrases from hacker forums that link the cyber attacks prior to the attacks in three different cyber attack types. Thus, we can extend our trend topic phrase extraction to be part of a new cyber attack prediction (forecasting) system that will take cyber attack related posts from hacker forums daily and each day's posts will be a feature to predict cyber attacks in specific time window. Since hacker forums discuss many topics, CARS will also help to filter the posts from the forums.

REFERENCES

[1] Threat analysis - intelligence — monitor - track cyber threats. https://www.surfwatchlabs.com/threat-intelligence-products/threat-analyst, 2018. Accessed: 2021-05-01.

[2] Valve says it's safe to play cs:go and tf2 after source code leaked online. https://www.zdnet.com/article/valve-says-its-safe-to-play-csgo-and-tf2-after-source-code-leaked-online/, 2020. Accessed: 2021-05-01.

[3] Cyber intelligence — cognyte. https://www.cognyte.com/cyber-intelligence/cyber-threat-intelligence/, 2021. Accessed: 2021-05-01.

[4] A. Abbasi, W. Li, V. A. Benjamin, S. Hu, and H. Chen. Descriptive analytics: Examining expert hackers in web forums. In *IEEE Joint Intelligence and Security Informatics Conference, JISIC 2014, The Hague, The Netherlands, 24-26 September, 2014*, pages 56–63, 2014.

[5] G. Aguilar, S. Maharjan, A. P. López-Monroy, and T. Solorio. A multi-task approach for named entity recognition in social media data. In L. Derczynski, W. Xu, A. Ritter, and T. Baldwin, editors, *Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 148–153. Association for Computational Linguistics, 2017.

[6] L. M. Aiello, G. Petkos, C. J. Martín, D. Corney, S. Papadopoulos, R. Skraba, A. Göker, I. Kompatsiaris, and A. Jaimes. Sensing trending topics in twitter. *IEEE Trans. Multim.*, 15(6):1268–1282, 2013.

[7] A. Akbik, D. Blythe, and R. Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1638–1649, 2018.

[8] A. T. Albaham, N. Salim, and O. I. Adekunle. Leveraging post level quality indicators in online forum thread retrieval. In T. Herawan, M. M. Deris, and J. H. Abawajy, editors, *Proceedings of the First International Conference on Advanced Data and Information Engineering, DaEng 2013, Kuala Lumpur, Malaysia, December 16-18, 2013*, volume 285 of *Lecture Notes in Electrical Engineering*, pages 417–425. Springer, 2013.

[9] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami. Contributions to the study of SMS spam filtering: new collection and results. In M. R. B. Hardy and F. W. Tompa, editors, *Proceedings of the 2011 ACM Symposium on Document Engineering, Mountain View, CA, USA, September 19-22, 2011*, pages 259–262. ACM, 2011.

[10] M. Almukaynizi, A. Grimm, E. Nunes, J. Shakarian, and P. Shakarian. Predicting cyber threats through hacker social networks in darkweb and deepweb forums. In *Proceedings of the 2017 International Conference of The Computational Social Science Society of the Americas*, page 12. ACM, 2017.

[11] M. Almukaynizi, E. Marin, E. Nunes, P. Shakarian, G. I. Simari, D. Kapoor, and T. Siedlecki. DARKMENTION: A deployed system to predict enterprise-targeted external cyberattacks. In *2018 IEEE International Conference on Intelligence and Security Informatics, ISI 2018, Miami, FL, USA, November 9-11, 2018*, pages 31–36. IEEE, 2018.

[12] M. Almukaynizi, E. Marin, M. Shah, E. Nunes, G. I. Simari, and P. Shakarian. A logic programming approach to predict enterprise-targeted cyberattacks. In *Data Science in Cybersecurity and Cyberthreat Intelligence*, pages 13–32. Springer, 2020.

[13] 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 E. Sobiesk, D. Bennett, and P. Maxwell, editors, *2017 International Conference on Cyber Conflict, CyCon U.S. 2017, Washington, DC, USA, November 7-8, 2017*, pages 82–88. IEEE Computer Society, 2017.

[14] M. Almukaynizi, E. Nunes, K. Dharaiya, M. Senguttuvan, J. Shakarian, and P. Shakarian. Patch before exploited: An approach to identify targeted software vulnerabilities. In *AI in Cybersecurity*, pages 81–113. Springer, 2019.

[15] F. Alves, A. Bettini, P. M. Ferreira, and A. Bessani. Processing tweets for cybersecurity threat awareness. *Information Systems*, 95:101586, 2021.

[16] K. Ameri, M. Hempel, H. R. Sharif, J. Lopez, and K. S. Perumalla. Cybert: Cybersecurity claim classification by fine-tuning the bert language model. *Journal of Cybersecurity and Privacy*, 2021.

[17] D. Angelov. Top2vec: Distributed representations of topics. 2020.

[18] A. Baevski, S. Edunov, Y. Liu, L. Zettlemoyer, and M. Auli. Cloze-driven pretraining of self-attention networks. *CoRR*, abs/1903.07785, 2019.

[19] P. Banerjee, K. K. Pal, M. V. Devarakonda, and C. Baral. Biomedical named entity recognition via knowledge guidance and question answering. *ACM Trans. Comput. Heal.*, 2(4):33:1–33:24, 2021.

[20] V. Behzadan, C. Aguirre, A. Bose, and W. H. Hsu. Corpus and deep learning classifier for collection of cyber threat indicators in twitter stream. In N. Abe, H. Liu, C. Pu, X. Hu, N. K. Ahmed, M. Qiao, Y. Song, D. Kossmann, B. Liu, K. Lee, J. Tang, J. He, and J. S. Saltz, editors, *IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018*, pages 5002–5007. IEEE, 2018.

[21] V. A. Benjamin, W. Li, T. Holt, and H. Chen. Exploring threats and vulnerabilities in hacker web: Forums, IRC and carding shops. In *2015 IEEE International Conference on Intelligence and Security Informatics, ISI 2015, Baltimore, MD, USA, May 27-29, 2015*, pages 85–90. IEEE, 2015.

[22] A. Bhandari and C. Armstrong. Tkol, httt, and r/radiohead: High affinity terms in reddit communities. In W. Xu, A. Ritter, T. Baldwin, and A. Rahimi, editors, *Proceedings of the 5th Workshop on Noisy User-generated Text, W-NUT@EMNLP 2019, Hong Kong, China, November 4, 2019*, pages 57–67. Association for Computational Linguistics, 2019.

[23] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[24] 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. In F. Spezzano, W. Chen, and X. Xiao, editors, *ASONAM '19: International Conference on Advances in Social Networks Analysis and Mining, Vancouver, British Columbia, Canada, 27-30 August, 2019*, pages 871–878. ACM, 2019.

[25] R. A. Bridges, C. L. Jones, M. D. Iannacone, and J. R. Goodall. Automatic labeling for entity extraction in cyber security. *CoRR*, abs/1308.4941, 2013.

[26] J. Brown, A. J. Broderick, and N. Lee. Word of mouth communication within online communities: Conceptualizing the online social network. *Journal of interactive marketing*, 21(3):2–20, 2007.

[27] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.

[28] R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, editors, *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II*, volume 7819 of *Lecture Notes in Computer Science*, pages 160–172. Springer, 2013.

[29] X. Carreras, L. Màrquez, and L. Padró. Learning a perceptron-based named entity chunker via online recognition feedback. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 156–159, 2003.

[30] K. E. K. Chai. *A machine learning-based approach for automated quality assessment of user generated content in web forums.* PhD thesis, Curtin University, 2011.

[31] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*, 2020.

[32] H. L. Chieu and H. T. Ng. Named entity recognition: A maximum entropy approach using global information. In *19th International Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, August 24 - September 1, 2002*, 2002.

[33] F. Chollet et al. Keras, 2015.

[34] Z. L. Chua, S. Shen, P. Saxena, and Z. Liang. Neural nets can learn function type signatures from binaries. In E. Kirda and T. Ristenpart, editors, *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017*, pages 99–116. USENIX Association, 2017.

[35] P. Cimiano, S. Handschuh, and S. Staab. Towards the self-annotating web. In *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004*, pages 462–471, 2004.

[36] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 999888:2493–2537, Nov. 2011.

[37] M. Del Vicario, A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi. The spreading of misinformation online. *Proceedings of the National Academy of Sciences*, 113(3):554–559, 2016.

[38] I. Deliu, C. Leichter, and K. Franke. Extracting cyber threat intelligence from hacker forums: Support vector machines versus convolutional neural networks. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 3648–3656. IEEE, 2017.

[39] I. Deliu, C. Leichter, and K. Franke. Collecting cyber threat intelligence from hacker forums via a two-stage, hybrid process using support vector machines and latent dirichlet allocation. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 5008–5013. IEEE, 2018.

[40] L. Derczynski, E. Nichols, M. van Erp, and N. Limsopatham. Results of the WNUT2017 shared task on novel and emerging entity recognition. In L. Derczynski, W. Xu, A. Ritter, and T. Baldwin, editors, *Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 140–147. Association for Computational Linguistics, 2017.

[41] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[42] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[44] Y. Dong, W. Guo, Y. Chen, X. Xing, Y. Zhang, and G. Wang. Towards the detection of inconsistencies in public security vulnerability reports. In N. Heninger and P. Traynor, editors, *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, pages 869–885. USENIX Association, 2019.

[45] Z. Fang, X. Zhao, Q. Wei, G. Chen, Y. Zhang, C. Xing, W. Li, and H. Chen. Exploring key hackers and cybersecurity threats in chinese hacker communities. In *IEEE Conference on Intelligence and Security Informatics, ISI 2016, Tucson, AZ, USA, September 28-30, 2016*, pages 13–18, 2016.

[46] J. R. Finkel, T. Grenager, and C. D. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 363–370, 2005.

[47] J. Fox. Cybersecurity statistics 2021. https://www.cobalt.io/blog/cybersecurity-statistics-2021, 2021. Accessed: 2022-04-01.

[48] T. Fu, A. Abbasi, and H. Chen. Interaction coherence analysis for dark web forums. In *IEEE International Conference on Intelligence and Security Informatics, ISI 2007, New Brunswick, New Jersey, USA, May 23-24, 2007, Proceedings*, pages 342–349, 2007.

[49] S. Fujita, H. Kobayashi, and M. Okumura. Dataset creation for ranking constructive news comments. In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2619–2626. Association for Computational Linguistics, 2019.

[50] H. Gasmi, A. Bouras, and J. Laval. Lstm recurrent neural networks for cybersecurity named entity recognition. *ICSEA*, 11:2018, 2018.

[51] S. Goel. Cyberwarfare: connecting the dots in cyber intelligence. *Commun. ACM*, 54(8):132–140, 2011.

[52] S. Goel. Cyberwarfare: connecting the dots in cyber intelligence. *Communications of the ACM*, 54(8):132–140, 2011.

[53] A. Graves, A. Mohamed, and G. E. Hinton. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, pages 6645–6649. IEEE, 2013.

[54] T. L. Griffiths, M. Steyvers, and J. B. Tenenbaum. Topics in semantic representation. *Psychological review*, 114(2):211, 2007.

[55] M. Grootendorst. Bertopic: Leveraging bert and c-tf-idf to create easily interpretable topics., 2020.

[56] K. Halder, M. Kan, and K. Sugiyama. Predicting helpful posts in open-ended discussion forums: A neural architecture. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3148–3157, 2019.

[57] D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[58] A. Heydari, M. Tavakoli, Z. Ismail, and N. Salim. Leveraging quality metrics in voting model based thread retrieval. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 10(1):117–23, 2016.

[59] T. Hofmann. Probabilistic latent semantic indexing. In F. C. Gey, M. A. Hearst, and R. M. Tong, editors, *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*, pages 50–57. ACM, 1999.

[60] M. Honnibal and I. Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7, 2017.

[61] S. Horawalavithana, A. Bhattacharjee, R. Liu, N. Choudhury, L. O. Hall, and A. Iamnitchi. Mentions of security vulnerabilities on reddit, twitter and github. In P. M. Barnaghi, G. Gottlob, Y. Manolopoulos, T. Tzouramanis, and A. Vakali, editors, *2019 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2019, Thessaloniki, Greece, October 14-17, 2019*, pages 200–207. ACM, 2019.

[62] C. Huang, Y. Guo, W. Guo, and Y. Li. Hackerrank: Identifying key hackers in underground forums. *Int. J. Distributed Sens. Networks*, 17(5):155014772110151, 2021.

[63] Z. Huang, W. Xu, and K. Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015.

[64] J. Hughes, S. Aycock, A. Caines, P. Buttery, and A. Hutchings. Detecting trending terms in cybersecurity forum discussions. In W. Xu, A. Ritter, T. Baldwin, and A. Rahimi, editors, *Proceedings of the Sixth Workshop on Noisy User-generated Text, W-NUT@EMNLP 2020 Online, November 19, 2020*, pages 107–115. Association for Computational Linguistics, 2020.

[65] H. Isozaki and H. Kazawa. Efficient support vector classifiers for named entity recognition. In *19th International Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, August 24 - September 1, 2002*, 2002.

[66] J. W. Johnsen and K. Franke. Identifying proficient cybercriminals through text and network analysis. In *IEEE International Conference on Intelligence and Security Informatics, ISI 2020, Arlington, VA, USA, November 9-10, 2020*, pages 1–7. IEEE, 2020.

[67] C. L. Jones, R. A. Bridges, K. M. T. Huffer, and J. R. Goodall. Towards a relation extraction framework for cyber-security concepts. In *Proceedings of the 10th Annual Cyber and Information Security Research Conference, CISR '15, Oak Ridge, TN, USA, April 7-9, 2015*, pages 11:1–11:4, 2015.

[68] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.

[69] K. Kashihara, H. S. Sandhu, and J. Shakarian. Automated corpus annotation for cybersecurity named entity recognition with small keyword dictionary. In K. Arai, editor, *Intelligent Systems and Applications - Proceedings of the 2021 Intelligent Systems Conference, IntelliSys 2021, Amsterdam, The Netherlands, 2-3 September, 2021, Volume 3*, volume 296 of *Lecture Notes in Networks and Systems*, pages 155–174. Springer, 2021.

[70] K. Kashihara, J. Shakarian, and C. Baral. Human-machine interaction for improved cybersecurity named entity recognition considering semantic similarity. In K. Arai, S. Kapoor, and R. Bhatia, editors, *Intelligent Systems and Applications - Proceedings of the 2020 Intelligent Systems Conference, IntelliSys, London, UK, September 3-4, 2020, Volume 2*, volume 1251 of *Advances in Intelligent Systems and Computing*, pages 347–361. Springer, 2020.

[71] K. Kashihara, J. Shakarian, and C. Baral. Social structure construction from the forums using interaction coherence. In K. Arai, S. Kapoor, and R. Bhatia, editors, *Proceedings of the Future Technologies Conference, FTC 2020, Volume 3*, Advances in Intelligent Systems and Computing, pages 830–843, Germany, 2021. Springer Science and Business Media Deutschland GmbH. Publisher Copyright: © 2021, Springer Nature Switzerland AG.; Future Technologies Conference, FTC 2020 ; Conference date: 05-11-2020 Through 06-11-2020.

[72] D. Khashabi, S. Min, T. Khot, A. Sabharwal, O. Tafjord, P. Clark, and H. Hajishirzi. Unifiedqa: Crossing format boundaries with a single QA system. In T. Cohn, Y. He, and Y. Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume

EMNLP 2020 of *Findings of ACL*, pages 1896–1907. Association for Computational Linguistics, 2020.

[73] J. W. Kim. They liked and shared: Effects of social media virality metrics on perceptions of message influence and behavioral intentions. *Comput. Hum. Behav.*, 84:153–161, 2018.

[74] Y. Kim. Convolutional neural networks for sentence classification. In A. Moschitti, B. Pang, and W. Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL, 2014.

[75] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[76] I. Kirillov, D. Beck, P. Chase, and R. Martin. Malware attribute enumeration and characterization. *The MITRE Corporation [online, accessed Apr. 8, 2019]*, 2011.

[77] H. Kwak, Y. Choi, Y. Eom, H. Jeong, and S. B. Moon. Mining communities in networks: a solution for consistency and its evaluation. In *Proceedings of the 9th ACM SIGCOMM Internet Measurement Conference, IMC 2009, Chicago, Illinois, USA, November 4-6, 2009*, pages 301–314, 2009.

[78] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270, 2016.

[79] J. H. Lau and T. Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. In P. Blunsom, K. Cho, S. B. Cohen, E. Grefenstette, K. M. Hermann, L. Rimell, J. Weston, and S. W. Yih, editors, *Proceedings of the 1st Workshop on Representation Learning for NLP, Rep4NLP@ACL 2016, Berlin, Germany, August 11, 2016*, pages 78–86. Association for Computational Linguistics, 2016.

[80] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1188–1196. JMLR.org, 2014.

[81] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019.

[82] G. L'Huillier, H. Álvarez, S. A. Ríos, and F. Aguilera. Topic-based social network analysis for virtual communities of interests in the dark web. *SIGKDD Explorations*, 12(2):66–73, 2010.

[83] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[84] Z. Liu, D. Huang, K. Huang, Z. Li, and J. Zhao. Finbert: A pre-trained financial language representation model for financial text mining. In C. Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4513–4519. ijcai.org, 2020.

[85] N. Lourie, R. L. Bras, C. Bhagavatula, and Y. Choi. UNICORN on RAINBOW: A universal commonsense reasoning model on a new multitask benchmark. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13480–13488. AAAI Press, 2021.

[86] X. Ma and E. H. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

[87] S. Marchal, J. François, R. State, and T. Engel. Phishstorm: Detecting phishing with streaming analytics. *IEEE Transactions on Network and Service Management*, 11:458–471, 2014.

[88] R. Marimont and M. Shapiro. Nearest neighbour searches and the curse of dimensionality. *IMA Journal of Applied Mathematics*, 24(1):59–70, 1979.

[89] E. Marin, J. Shakarian, and P. Shakarian. Mining key-hackers on darkweb forums. In *1st International Conference on Data Intelligence and Security, ICDIS 2018, South Padre Island, TX, USA, April 8-10, 2018*, pages 73–80, 2018.

[90] A. McCallum, D. Freitag, and F. C. N. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 591–598, 2000.

[91] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 188–191, 2003.

[92] B. McCann, N. S. Keskar, C. Xiong, and R. Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

[93] S. McDonnell, O. Nada, M. R. Abid, and E. Amjadian. Cyberbert: A deep dynamic-state session-based recommender system for cyber threat recognition. *2021 IEEE Aerospace Conference (50100)*, pages 1–12, 2021.

[94] L. McInnes and J. Healy. Accelerated hierarchical density based clustering. In R. Gottumukkala, X. Ning, G. Dong, V. Raghavan, S. Aluru, G. Karypis, L. Miele, and X. Wu, editors, *2017 IEEE International Conference on Data Mining Workshops, ICDM Workshops 2017, New Orleans, LA, USA, November 18-21, 2017*, pages 33–42. IEEE Computer Society, 2017.

[95] L. McInnes, J. Healy, and S. Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017.

[96] L. McInnes, J. Healy, N. Saul, and L. Großberger. UMAP: uniform manifold approximation and projection. *J. Open Source Softw.*, 3(29):861, 2018.

[97] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In Y. Bengio and Y. LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[98] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119, 2013.

[99] T. Mikolov, W. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In L. Vanderwende, H. D. III, and K. Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 746–751. The Association for Computational Linguistics, 2013.

[100] S. Mishra, D. Khashabi, C. Baral, Y. Choi, and H. Hajishirzi. Reframing instructional prompts to gptk's language. *CoRR*, abs/2109.07830, 2021.

[101] S. Morgan. Hackerpocalypse cybercrime report 2016. https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/, 2016. Accessed: 2022-04-01.

[102] M. Motoyama, D. McCoy, K. Levchenko, S. Savage, and G. M. Voelker. An analysis of underground forums. In *Proceedings of the 11th ACM SIGCOMM Internet Measurement Conference, IMC '11, Berlin, Germany, November 2-, 2011*, pages 71–80, 2011.

[103] V. Mulwad, W. Li, A. Joshi, T. Finin, and K. Viswanathan. Extracting information about security vulnerabilities from web text. In *Proceedings of the 2011 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2011, Campus Scientifique de la Doua, Lyon, France, August 22-27, 2011*, pages 257–260, 2011.

[104] T. H. Nguyen and R. Grishman. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 365–371, 2015.

[105] R. D. Nolker and L. Zhou. Social computing and weighting to identify member roles in online communities. In *2005 IEEE / WIC / ACM International Conference on Web Intelligence (WI 2005), 19-22 September 2005, Compiegne, France*, pages 87–93, 2005.

[106] NSF. National artificial intelligence (ai) research institutes accelerating. https://www.nsf.gov/pubs/2022/nsf22502/nsf22502.htm. Accessed: 2022-04-01.

[107] E. Nunes, A. Diab, A. T. Gunn, E. Marin, V. Mishra, V. Paliath, J. Robertson, J. Shakarian, A. Thart, and P. Shakarian. Darknet and deepnet mining for proactive cybersecurity threat intelligence. In *IEEE Conference on Intelligence and Security Informatics, ISI 2016, Tucson, AZ, USA, September 28-30, 2016*, pages 7–12. IEEE, 2016.

[108] A. Okutan, S. J. Yang, and K. McConky. Predicting cyber attacks with bayesian networks using unconventional signals. In J. P. Trien, S. J. Prowell, J. R. Goodall, J. M. Beaver, and R. A. Bridges, editors, *Proceedings of the 12th Annual Conference on Cyber and Information Security Research, CISRC 2017, Oak Ridge, TN, USA, April 4 - 6, 2017*, pages 13:1–13:4. ACM, 2017.

[109] A. Osman, N. Salim, and F. Saeed. Quality-based text web forum summarization-a review. *International Journal of Soft Computing*, 12(1):31–44, 2017.

[110] A. Osman, N. Salim, and F. Saeed. Quality dimensions features for identifying high-quality user replies in text forum threads using classification methods. *PloS one*, 14(5):e0215516, 2019.

[111] K. K. Pal, K. Kashihara, P. Banerjee, S. Mishra, R. Wang, and C. Baral. Constructing flow graphs from procedural cybersecurity texts. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3945–3957. Association for Computational Linguistics, 2021.

[112] K. K. Pal, K. Kashihara, P. Banerjee, S. Mishra, R. Wang, and C. Baral. Constructing flow graphs from procedural cybersecurity texts. *CoRR*, abs/2105.14357, 2021.

[113] P. Pantel and M. Pennacchiotti. Automatically harvesting and ontologizing semantic relations. In *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 171–195. 2008.

[114] P. Passeri. Hackmageddon. https://www.hackmageddon.com/. Accessed: 2021-08-01.

[115] S. Pastrana, A. Hutchings, A. Caines, and P. Buttery. Characterizing eve: Analysing cybercrime actors in a large underground forum. In *RAID*, 2018.

[116] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[117] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[118] K. Pei, J. Guan, M. Broughton, Z. Chen, S. Yao, D. Williams-King, V. Ummadisetty, J. Yang, B. Ray, and S. Jana. Stateformer: fine-grained type recovery from binaries using generative state modeling. In D. Spinellis, G. Gousios, M. Chechik, and M. D. Penta, editors, *ESEC/FSE '21: 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, August 23-28, 2021*, pages 690–702. ACM, 2021.

[119] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In A. Moschitti, B. Pang, and W. Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL, 2014.

[120] I. Pete, J. Hughes, Y. T. Chua, and M. Bada. A social network analysis and comparison of six dark web forums. In *IEEE European Symposium on Security and Privacy Workshops, EuroS&P Workshops 2020, Genoa, Italy, September 7-11, 2020*, pages 484–493. IEEE, 2020.

[121] U. Pfeil and P. Zaphiris. Investigating social network patterns within an empathic online community for older people. *Computers in Human Behavior*, 25(5):1139–1155, 2009.

[122] L. N. Phan, J. T. Anibal, H. Tran, S. Chanana, E. Bahadroglu, A. Peltekian, and G. Altan-Bonnet. Scifive: a text-to-text transformer model for biomedical literature. *arXiv preprint arXiv:2106.03598*, 2021.

[123] P. Phandi, A. Silva, and W. Lu. SemEval-2018 task 8: Semantic extraction from CybersecUrity REports using natural language processing (SecureNLP). In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 697–706, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[124] E. Phillips, J. R. Nurse, M. Goldsmith, and S. Creese. Extracting social structure from darkweb forums. 2015.

[125] R. S. Portnoff, S. Afroz, G. Durrett, J. K. Kummerfeld, T. Berg-Kirkpatrick, D. McCoy, K. Levchenko, and V. Paxson. Tools for automated analysis of cybercriminal markets. In R. Barrett, R. Cummings, E. Agichtein, and E. Gabrilovich, editors, *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 657–666. ACM, 2017.

[126] A. L. Queiroz, S. Mckeever, and B. Keegan. Detecting hacker threats: Performance of word and sentence embedding models in identifying hacker communications. In *AICS*, 2019.

[127] V. R., M. Alazab, A. Jolfaei, S. K. P., and P. Poornachandran. Ransomware triage using deep learning: Twitter as a case study. In *Cybersecurity and Cyberforensics Conference, CCC 2019, Melbourne, Australia, May 8-9, 2019*, pages 67–73. IEEE, 2019.

[128] V. R, M. Alazab, S. Srinivasan, Q.-V. Pham, S. Padannayil, and S. Ketha. A visualized botnet detection system based deep learning for the internet of things networks of smart cities. *IEEE Transactions on Industry Applications*, pages 1–1, 01 2020.

[129] 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.

[130] C. Sabottke, O. Suciu, and T. Dumitras. Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits. In J. Jung and T. Holz, editors, *24th USENIX Security Symposium, USENIX Security 15, Washington, D.C., USA, August 12-14, 2015*, pages 1041–1056. USENIX Association, 2015.

[131] S. Saganowski. Cybersecurity NER corpus 2019, 2020.

[132] S. Samtani and H. Chen. Using social network analysis to identify key hackers for keylogging tools in hacker forums. In *IEEE Conference on Intelligence and Security Informatics, ISI 2016, Tucson, AZ, USA, September 28-30, 2016*, pages 319–321, 2016.

[133] S. Samtani and H. Chen. Using social network analysis to identify key hackers for keylogging tools in hacker forums. In *2016 IEEE conference on intelligence and security informatics (ISI)*, pages 319–321. IEEE, 2016.

[134] E. F. T. K. Sang and S. Buchholz. Introduction to the conll-2000 shared task chunking. In *Fourth Conference on Computational Natural Language Learning, CoNLL 2000, and the Second Learning Language in Logic Workshop, LLL 2000, Held in cooperation with ICGI-2000, Lisbon, Portugal, September 13-14, 2000*, pages 127–132, 2000.

[135] A. Sapienza, S. K. Ernala, A. Bessi, K. Lerman, and E. Ferrara. DISCOVER: mining online chatter for emerging cyber threats. In P. Champin, F. L. Gandon, M. Lalmas, and P. G. Ipeirotis, editors, *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, pages 983–990. ACM, 2018.

[136] S. Sarkar, M. Almukaynizi, J. Shakarian, and P. Shakarian. Predicting enterprise cyber incidents using social network analysis on the darkweb hacker forums. In *2018 International Conference on Cyber Conflict, CyCon U.S. 2018, Washington, DC, USA, November 14-15, 2018*, pages 1–7, 2018.

[137] T. Satyapanich, F. Ferraro, and T. Finin. CASIE: extracting cybersecurity event information from text. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8749–8757. AAAI Press, 2020.

[138] T. Satyapanich, F. Ferraro, and T. Finin. CASIE: extracting cybersecurity event information from text. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8749–8757. AAAI Press, 2020.

[139] M. R. Shahid and H. Debar. Cvss-bert: Explainable natural language processing to determine the severity of a computer security vulnerability from its description. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1600–1607, 2021.

[140] D. A. Shamma, L. Kennedy, and E. F. Churchill. Peaks and persistence: modeling the shape of microblog conversations. In P. J. Hinds, J. C. Tang, J. Wang, J. E. Bardram, and N. Ducheneaut, editors, *Proceedings of the 2011 ACM Conference on Computer Supported Cooperative Work, CSCW 2011, Hangzhou, China, March 19-23, 2011*, pages 355–358. ACM, 2011.

[141] K. Simran, S. Sriram, R. Vinayakumar, and K. Soman. Deep learning approach for intelligent named entity recognition of cyber security. In *International Symposium on Signal Processing and Intelligent Recognition Systems*, pages 163–172. Springer, 2019.

[142] A. Sirotina and N. Loukachevitch. Named entity recognition in information security domain for Russian. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1114–1120, Varna, Bulgaria, Sept. 2019. INCOMA Ltd.

[143] J. Swarner. Before wannacry was unleashed, hackers plotted about it on the dark web, 2017.

[144] J. Swearingen. The creator of the mirai botnet is probably a rutgers student with the bad habit of bragging, 2017.

[145] J. Tabassum, M. Maddela, W. Xu, and A. Ritter. Code and named entity recognition in stackoverflow. In *The Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[146] N. Tavabi, N. Bartley, A. Abeliuk, S. Soni, E. Ferrara, and K. Lerman. Characterizing activity on the deep and dark web. In S. Amer-Yahia, M. Mahdian, A. Goel, G. Houben, K. Lerman, J. J. McAuley, R. Baeza-Yates, and L. Zia, editors, *Companion of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 206–213. ACM, 2019.

[147] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[148] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[149] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[150] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman. Robust intelligent malware detection using deep learning. *IEEE Access*, 7:46717–46738, 2019.

[151] R. Vinayakumar, K. P. Soman, and P. Poornachandran. Detecting malicious domain names using deep learning approaches at scale. *J. Intell. Fuzzy Syst.*, 34(3):1355–1367, 2018.

[152] R. Vinayakumar, K. P. Soman, and P. Poornachandran. Evaluating deep learning approaches to characterize and classify malicious url's. *J. Intell. Fuzzy Syst.*, 34(3):1333–1343, 2018.

[153] N. M. Wanas, M. El-Saban, H. Ashour, and W. Ammar. Automatic scoring of online discussion posts. In K. Tanaka, T. Matsuyama, E. Lim, and A. Jatowt, editors, *Proceedings of the 2nd ACM Workshop on Information Credibility on the Web, WICOW 2008, Napa Valley, California, USA, October 30, 2008*, pages 19–26. ACM, 2008.

[154] L. Wang, R. Li, Y. Yan, Y. Yan, S. Wang, W. Wu, and W. Xu. Instructionner: A multi-task instruction-based generative framework for few-shot NER. *CoRR*, abs/2203.03903, 2022.

[155] Y. Wang, W. Wang, S. R. Joty, and S. C. H. Hoi. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In M. Moens, X. Huang, L. Specia, and S. W. Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8696–8708. Association for Computational Linguistics, 2021.

[156] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[157] M. Weimer and I. Gurevych. Predicting the perceived quality of web forum posts. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 643–648, 2007.

[158] M. Weimer, I. Gurevych, and M. Mühlhäuser. Automatically assessing the post quality in online discussions on software. In J. A. Carroll, A. van den Bosch, and A. Zaenen, editors, *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics, 2007.

[159] H. Yan, T. Gui, J. Dai, Q. Guo, Z. Zhang, and X. Qiu. A unified generative framework for various NER subtasks. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5808–5822. Association for Computational Linguistics, 2021.

[160] J. Yang and S. Counts. Predicting the speed, scale, and range of information diffusion in twitter. In W. W. Cohen and S. Gosling, editors, *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*. The AAAI Press, 2010.

[161] A. Zenebe, M. Shumba, A. Carillo, and S. Cuenca. Cyber threat discovery from dark web. In *Proceedings of 28th International Conference*, volume 64, pages 174–183, 2019.

[162] X. Zhang and C. Li. Survival analysis on hacker forums. In *SIGBPS workshop on business processes and service*, pages 106–110. Citeseer, 2013.

[163] X. Zhang, A. Tsang, W. T. Yue, and M. Chau. The classification of hackers by knowledge exchange behaviors. *Information Systems Frontiers*, 17(6):1239–1251, 2015.

[164] Z. Zhang, Y. Ye, W. You, G. Tao, W. Lee, Y. Kwon, Y. Aafer, and X. Zhang. OSPREY: recovery of variable and data structure via probabilistic analysis for stripped binary. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 813–832. IEEE, 2021.

[165] J. Zhao, Q. Yan, J. Li, M. Shao, Z. He, and B. Li. Timiner: Automatically extracting and analyzing categorized cyber threat intelligence from social data. *Comput. Secur.*, 95:101867, 2020.