AI-based Cyber Event OSINT via Twitter Data

Dakota Dale
University of Arkansas
dsdale@uark.edu

Kylie McClanahan
University of Arkansas
klmcclan@uark.edu

Qinghua Li
University of Arkansas
qinghual@uark.edu

***Abstract*—Open-Source Intelligence (OSINT) is largely regarded as a necessary component for cybersecurity intelligence gathering to secure network systems. With the advancement of artificial intelligence (AI) and increasing usage of social media, like Twitter, we have a unique opportunity to obtain and aggregate information from social media. In this study, we propose an AI-based scheme capable of automatically pulling information from Twitter, filtering out security-irrelevant tweets, performing natural language analysis to correlate the tweets about each cybersecurity event (e.g., a malware campaign), and validating the information. This scheme has many applications, such as providing a means for security operators to gain insight into ongoing events and helping them prioritize vulnerabilities to deal with. To give examples of the possible uses, we present three case studies demonstrating the event discovery and investigation processes.**

***Index Terms*—Cybersecurity, OSINT, AI**

I. Introduction

Cyber attacks on network infrastructures are becoming more frequent and catastrophic. For example, Microsoft was a victim of a data breach discovered in January 2020 that caused over 250 million customer records leaked online [1]. In May 2021, Colonial Pipeline was hit with a ransomware attack [2] that caused gas supply to the east coast of the United States to be completely cut off for days. Cyber attacks are also constantly evolving. Sonicwall found 442,151 new malware variants in 2021, an increase of 65% over 2020 [3].

To better protect network and information systems, it is crucial for a system to identify these cyber events as they occur and also aggregate all of the necessary pieces of information. Open-source intelligence (OSINT) is one of the most important tools for securing cyberspace [4]. OSINT refers to the search and collection of intelligence through public resources such as datasets, blogs, or social media sites. This process is also aiding decision-making for policy, foreign affairs, and the economy [5]. There are two main categories of cybersecurity intelligence sources: formal and informal. Formal sources are typically government-sponsored sites that collect technical information on cyber vulnerabilities such as the National Vulnerability Database (NVD) or the Cybersecurity & Infrastructure Security Agency (CISA). Conversely, informal sources are developed by independent entities such as contractors or hobbyists and then made available online through a blog or social media page.

Due to the widespread usage of social media by both governmental organizations and independent entities, as well as the volume and velocity with which data is produced, Twitter seems to be a highly viable data source for cybersecurity intelligence, with 330 million monthly active users posting 500 million tweets per day [6]. The global 2017 "Petya/NotPetya" ransomware attack was discussed on Twitter as early as four months before the attack went public [7]. Hacktivists themselves even take to the social media to disseminate vulnerability information amongst their collective. In one incident, a malicious threat actor known as SandboxEscaper once released a zero-day, or previously unknown, vulnerability as well as linked proof-of-concept code in a public GitHub repository on Twitter. Less than two days later, a group known as PowerPool began exploiting the vulnerability in their own hacking campaign [8].

In this study, we leverage AI and Twitter data for OSINT. We propose a method to automatically collect and correlate the necessary pieces of information about specific cyber events (e.g., a malware campaign or vulnerability) for security operators to better understand them and make more timely and informed decisions. Specifically, the system pulls tweets and passes them through a multi-step AI pipeline. The first step serves as a cyber event detection model to identify not only tweets in the cybersecurity domain, but also to detect those that contain valuable information about specific events. The second step performs named entity recognition (NER), a natural language processing (NLP) technique, so that the names of different malware, threat actors, or companies can be extracted. Next, the named entities are used to create a word co-occurrence network to assess the correlation amongst entities and cluster them around events. Lastly, the subcomponents of the word co-occurrence network are validated against phenomena such as Twitter spam that can hinder the integrity of the information collected. The results of this pipeline are output to the user to show the current sub-topics in the cybersecurity community on Twitter and their potential to represent real events occurring. We provide evaluation results and present three case studies, demonstrating event discovery and investigation processes, as examples of the possible uses.

The paper is organized as follows. Section II reviews related work. Section III describes our approach. Section IV presents evaluation results. Section V provides three use cases. The last section concludes the paper.

II. Related Work

Twitter has been used for early detection of cybersecurity threats. CyberTwitter [9] proposes a profiling system that extracts vulnerability information about a user's installed software programs and browser extensions. [10] provides cyber intelligence by using NLP techniques and a specialized cyber-domain entity extractor, but it does not correlate the information between multiple tweets as we do. [11] also aims to address cyber event detection, but it uses keyword-based heuristics to identify cyber event tweets. [12] creates a system to identify tweets about cyber threats and extract named entities from each tweet to generate a security alert, but does not analyze how the content of different tweets may be linked. [8] also identifies cyber threat tweets but through keyword heuristics and sentiment analysis. In contrast with our study, these work do not discern the truthfulness of tweets, and they rely on the number of entities included in a tweet, the number of tweet followers, or the level of sentiment to decide security risks.

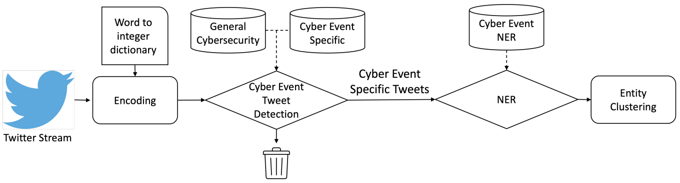


Fig. 1. System workflow.

III. Our Approach

A. Overview

Our pipeline (see Figure 1) contains three main phases: Cyber Event Tweet Detection, NER, and Event-centric Entity Cluster Identification and Validation. The first begins with pulling tweets from Twitter. These tweets are then passed to our Cyber Event Tweet Detection model which we trained using the concept of Transfer Learning. Transfer Learning is used in cases where the target data is too limited to train a classifier, so instead the classifier is trained on a related subject area with more prevalent data, then finalized using the target data. In our case, a cyber event-specific dataset can only be built by hand but a large dataset would be required to build a competent model. Thus, we pre-train the model on a large number of general cybersecurity tweets, which are easier to automatically obtain, and then fine-tune it on a small dataset of manually selected cyber event tweets. Once the model identifies the cyber event tweets, they are passed on to the next phase. The NER phase extracts the valuable information from the tweets, so that they can be clustered by occurrence in the Event Cluster Identification phase. In this last phase, each cluster, representing an independent cyber event, is also validated to provide users with metrics to gauge the spread and validity of the information. The user is provided with a report of each event with notable tweets for each and an interactive word co-occurrence network.

The primary users of our solution are professionals that oversee the cybersecurity risks of organizations such as security operators. They can use our approach for many purposes, e.g., investigating how on-going cyber attacks could affect their organization and how their un-patched vulnerabilities could soon be exploited.

B. Data Collection

Data for Cyber Event Tweet Detection: Due to the use of transfer learning, the cyber event tweet detection module needs general cybersecurity tweets, non-security tweets, and cyber event-specific tweets. To begin building a dataset of general cybersecurity tweets, we elected to first identify accounts belonging to industry professionals, similar to [8]. Through the use of Twint, an open-source Twitter scraping tool, we extracted 119 accounts who had posted a tweet containing keywords pertaining to cybersecurity such as “vulnerability”, “cybersecurity”, “phishing”, etc. To validate that the account is a credible source, we filtered the accounts by ensuring that their bios contain a title granting them some credibility in the field, essentially certifying the account. Table I shows the list of the search and filtering keywords. These steps derive the 332,518 positive data samples of general cybersecurity tweets.

TABLE I
Search & filtering terms to identify viable Twitter accounts

Filtering?	Keywords
Initial Twitter Stream	#cybersecurity, #vulnerability, #cyber, #cyberattack, #infosec, #ransomware, #malware, #hack, #hacker
Account Bio	Founder, Analyst, Scientist, Director, cybersecurity, hacker, Center, Centre, Dr., Doctor, CIO, Chief Innovation Officer, CEO, Chief Executive Officer

We still need negative samples of general cybersecurity tweets. A 2009 study developed a means of detecting the sentiment, or the opinions/emotions, of tweets. Due to the lack of large publicly available tweet datasets, researchers in that study had to create one [13]. This dataset, commonly known as the Sentiment-140, contains 1.6 million tweets each labeled as having either negative, neutral, or positive sentiment. Since its release, the Sentiment-140 dataset has been used to assess the opinions of scientific studies online [14], to predict stock movement [15], and even to enhance other datasets [16]. Considering these samples were queried from the Twitter stream according to one of two emoticons, :) and :(, it is reasonable to assume the majority are distributed into categories outside of cybersecurity. After the removal of stop words such as “the” and “at” along with any links present in the tweet, we were able to analyze the word frequencies of the data set. This analysis shows that the most common words used were “good”, “day”, “get”, “like”, and “go” thus showing no indication of cybersecurity tweets. Additionally, we found that the words “cybersecurity”, “cyber”, “hacker”, “malware”, “vulnerability”, and “exploit” only accounted for 219 of the 12,276,829 word occurrences. Thus, any tweets from this dataset that happen to fall in the cybersecurity domain should be overshadowed by the others, and their interference to the model should be minimal. 320,351 tweets are randomly selected from the Sentiment-140 dataset to serve as the negative class in our dataset.

Next, we collected the cyber event-specific dataset. This dataset will be used to fine-tune the cyber event tweet detection model. Due to the nature of cyber security information on Twitter, these tweets were selected by hand. Companies often share basic tips and tricks (i.e. “10 tips on how to avoid phishing scams”), but these are not useful for security operators to understand a specific event. Following a similar methodology as before, we manually searched a series of keywords related to cybersecurity and selected 181 tweets containing valid information such as CVEs (Common Vulnerabilities and Exposures), software assets, or malware families to serve as the positive

TABLE II
ENTITY LABELS FOR THE NER MODEL

Label	Examples
AttackType	Phishing, DDos, SQL Injection
Cardinal	1, two, 3, four
CVE	CVE-2022-23657, cve-2022-23658
Global-Political Entity (GPE)	United States, Russia, China, Canada
MalwareType	Ransomware, Spyware, Ryuk, Petya
Money	$50, six dollars, one-hundred euro
Ordinal	First, 2nd, Third, 4th
Organization (ORG)	Apple, Microsoft, Meta
Nationality, Religious, or Political groups (NORP)	American, Russian, Muslim, Democrat, Republican
Percent	10%, twenty percent
Product	iPhone, Windows, iOS

samples. Again, we appended an equivalent number of tweets from the Sentiment-140 dataset for the negative samples.

Data for Named Entity Recognition: Again using Twint, we scraped approximately 11,000 new tweets by keywords defined in Table I, and then routed them through the cyber event tweet detection model for filtering. Due to the output of neural networks being a continuous variable, we established a threshold of 0.5 to convert the probabilistic values into binary decisions. Tweets generating a value less than or equal to 0.5 are reassigned to 0 (denoting a non-cyber event tweet) and the rest to 1 (denoting a cyber event tweet). This left around 5,000 tweets containing valuable cyber event information. Using the Prodigy annotation software, we labeled the entities that may be useful for security operators to know, such as types of malware or CVEs. A complete list of entity labels, as well as some examples, are provided in Table. II.

C. Cyber Event Tweet Detection

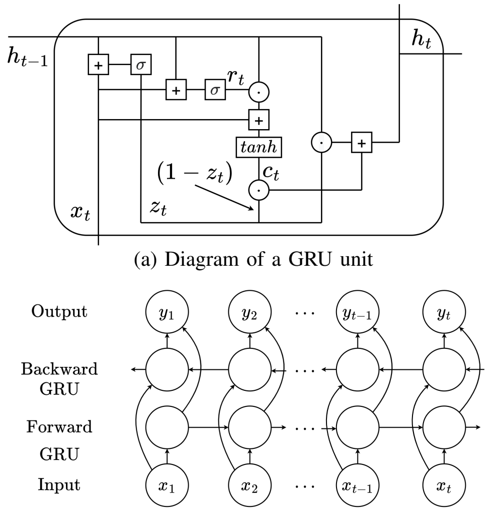
Architecture: We adopt a similar neural network architecture to [17] using BiGRU layers, but instead of GloVe embeddings, we use a randomly initialized embedding layer which will convert n integer-based word encodings into vectors of length d, thus creating an $n * d$ matrix representation of the entire tweet. We did not use GloVe embeddings because we found that approximately 44% of our tokens lacked a GloVe mapping.

The equations to compute the output of a GRU unit as shown in Figure 2a with input x_t and the output of the previous unit h_{t-1} are shown below, where σ is the sigmoid function and W_i, W_r, and W_c are the weights for the input, the reset gate, and the current memory content, respectively.

$$\begin{aligned} z_t &= \sigma(W_i * [h_{t-1}, x_t]) && (1) \\ r_t &= \sigma(W_r * [h_{t-1}, x_t]) && (2) \\ c_t &= tanh(W_c * [r_t \cdot h_{t-1}, x_t]) && (3) \\ h_t &= (1 - z_t) \cdot c_t + z_t \cdot h_{t-1} && (4) \end{aligned}$$

For a BiGRU architecture, one set of GRU units processes the input from start to end, while a second processes the input in reverse, as shown in Figure 2b.

We use an output dimension d of size 25 along with an embeddings regularizer ($L_2 = 0.0438$). The architecture contains a BiGRU layer with 128 state cells. Contrary to [17], we skip the concatenation of the BiGRU outputs and instead route them through a fully connected (FC) layer of 128 units. Lastly, the model includes a dropout layer with a 33% rate and a FC layer with 1 unit to achieve a binary classification.



(a) Diagram of a GRU unit

(b) A BiGRU architecture showing the direction

Fig. 2. Diagrams for GRU and BiGRU

Encoding: Before we could begin training the model, we first need to encode the tweets. The embedding layer present in the architecture detailed in Table III takes in an array of integers and converts them into vectors of uniform distribution. The maximum number of words possible in a tweet is 140, meaning that our array would contain 140 integers. Thus, we created a dictionary mapping each unique word present in the training data to a specific integer, with a few exceptions.

TABLE III
CYBER EVENT DETECTION MODEL ARCHITECTURE

Layer Type	Output Shape
Embedding	(None, 140, 25)
Bidirectional	(None, 256)
Dense	(None, 128)
Dropout	(None, 128)
Dense	(None, 1)

First, as shown in Table II, one of our primary labels for NER is the CVE. CVEs essentially act as reference tags for vulnerabilities. This also means that every CVE is unique. Mapping every CVE to its own integer is unhelpful, especially since any tweet containing a CVE should automatically be considered a positive case for the cyber event detection algorithm. Instead, we elected to map all CVEs to one integer: 2. Similarly, the set of unique words in the dataset is hardly all-encompassing and new malware and threat groups come out frequently. To remedy this and avoid issues with the model, any words not present in the dictionary, i.e. those without a mapping, are mapped to 1. Lastly, most tweets will not contain 140 words, so we must pad those entries with 0s to maintain a uniform size.

Training: As aforementioned, transfer learning is used. We first train a model to identify general cybersecurity tweets, and then train a model to identify cybersecurity events. To train the model to identify tweets relating to the cybersecurity domain as a whole, we trained it with 50% of the "General Cybersecurity dataset" mentioned previously and validated it with the remaining 50%. We specified 20 epochs and a batch size of 64. Then, using the "Cyber Event Specific" dataset, we focused the pre-trained model to identify tweets containing valuable event-specific information. Because this dataset is rather small, we elected to use a 90/10 training/testing split, 100 epochs, and a batch size of 20.

Training utilized binary crossentropy loss (Eq. (5)) and the Adam optimizer with a learning rate of 0.001. We also elected to implement two functions that augment the training process as a whole, known as callbacks: ReduceLRonPlateau and EarlyStopping. ReduceLRonPlateau keeps track of the validation loss and lower the learning rate by a factor of 0.1 to avoid overshooting the local minima. Similarly, EarlyStopping also keeps track of the validation loss but will stop the training process entirely if it stagnates for longer than 3 epochs.

$$\text{loss} = -y \log \hat{y} - (1-y)\log(1-\hat{y}) \tag{5}$$

D. Named Entity Recognition

At this point, the Twitter stream has been filtered to only include tweets containing valuable information about cybersecurity events. Thus, it is time to extract the specific entities that may be useful in describing them. The NLP library spaCy provides several pre-trained model pipelines, notably the en_core_web_md pipeline. This pipeline has been used in multiple domains from extracting brand names for sentiment analysis [18] to correlating diseases with specific pre-existing conditions [19]. It takes the raw text as input and passes it through a series of components, each taking in the output of the previous component and passing its own output to the next. Most of these components provide simple but necessary functions, such as word vectorization and part-of-speech tagging. Lastly, the NER component allows for the labeling of non-overlapping spans, so we can extract entities such as companies and malware families from tweets. Overall, this pipeline features 685 thousand keys and 20 thousand vectors of dimension 300. For training the pipeline, we kept the default configuration of an 80/20 training/testing split and the initial learning rate of 0.01. The components besides tok2vec and NER were frozen to avoid changing their weights.

E. Event Cluster Identification and Validation

After the entities are extracted, we aggregate them into specific events and analyze their degree of co-occurrence. For this phase we used three python libraries: Pandas, Pyvis, and Networkx. After iterating through the entities and creating a Pandas DataFrame of entity pairs and their number of co-occurrences, we used Pyvis and Networkx to construct undirected network graphs. Fig. 3a shows an example Pyvis graph. The nodes show the entities present in the initial scrape of the program. Each edge connecting two nodes shows the number of co-occurrences, or weight, of the pair. To avoid cluttering the network, we filtered out any edges where the weight was one, as one co-occurrence is not evidence of a strong correlation. Using the Networkx graph, we segmented the graph into its connected subcomponents (i.e., subgraphs) for further analysis. Each subgraph can be understood as a self-contained event and demonstrates the connections between the primary entities.

$$\text{Diffusion Index} = \#\text{Unique Users}/\#\text{Total Tweets} \tag{6}$$

$$\text{Spam Index} = \frac{\sum_{n=0}^{\#users}\left(\frac{1}{\#\text{ nth User Tweets}}\right)}{\#\text{Total Tweets}} \tag{7}$$

Due to the varying degree of veracity, or truthfulness, of social media data, it needs some means of validation. Measuring the frequency of tweets about a certain topic is not enough to gauge their validity because in extreme cases, those tweets could be coming from a single user in an attempt to clog or redirect the focus of the Twitter stream. Thus, measuring the number of tweets about a subject while also adjusting for number of accounts participating in the conversation is important. We adopt two metrics from a prior study [20]. The Diffusion Index, given by Eq. (6) measures how quickly information has spread. The Spam Index, given in Eq. (7), measures repeated tweets from the same user. It can be viewed as inflating the diffusion. In practice, we would grant more trust to topics with high diffusion and spam indices. A low diffusion would signify that a very small number of users, potentially a community, are discussing the event. Though this is does not inherently make the information false, when coupled with a low spam index (signifying disproportionate tweet contribution), it could be evident of tactics to obscure other, more severe, events.

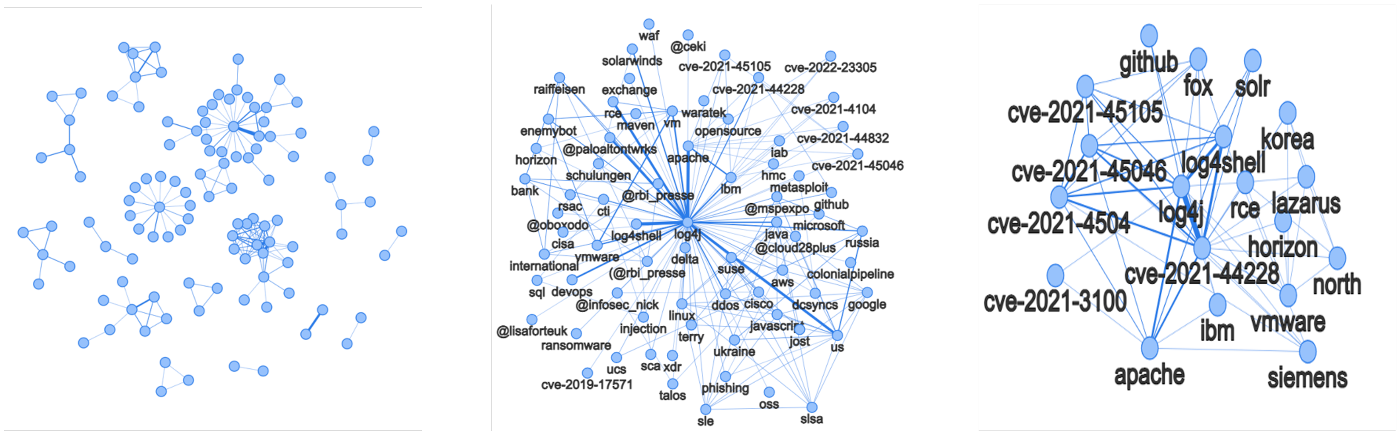
IV. Evaluations

A. Evaluation Metrics

We use 6 metrics: Area under the Receiver Operating Characteristic curve (AUC-ROC), Accuracy, Precision, Recall, F_1-score, and Specificity. The AUC-ROC score compares the true and false positive rates across different discrimination thresholds. The curve can be compared to a line from (0,0) to (1,1) which represents an untrained classifier that will label the samples randomly. The greater the ROC curve's deviation from this line, the higher the performance. The remaining metrics all operate on a binary basis, meaning any values above a threshold of 0.5 are converted to 1 (positive) and the rest to 0 (negative). The use of binary classification generates four cases: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Accuracy is defined as the number of correct, or true, predictions divided by the total number of predictions. Precision is defined as $\frac{\text{TP}}{(\text{TP+FP})}$. It focuses on how well the model detects the positive class and does not take into account any values concerning the negative case. Conversely, recall (defined as $\frac{\text{TP}}{(\text{TP+FN})}$) accounts for the false negatives, or samples incorrectly classified as negative. F_1-score is defined as $F_1 = 2 * \frac{\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$. This metric is more robust against class imbalances than standard accuracy is, but it does not demonstrate the models' performance on the negative class. Specificity is defined by $\frac{\text{TN}}{(\text{TN+FP})}$. It denotes the proportion of true negatives to the number of samples the model classified as negative, and should help demonstrate how well non-cyber event tweets are discarded.

B. Cyber Event Tweet Detection

Fig. 4 shows the performance of cyber event tweet detection. The pre-trained model achieved a 98.6% accuracy and a 98.6% AUC-ROC (Fig. 4c) score. It yielded 97.7%, 99.6%, 98.7%,



(a) An example network graph

(b) log4j

(c) log4j cve-2021-44228

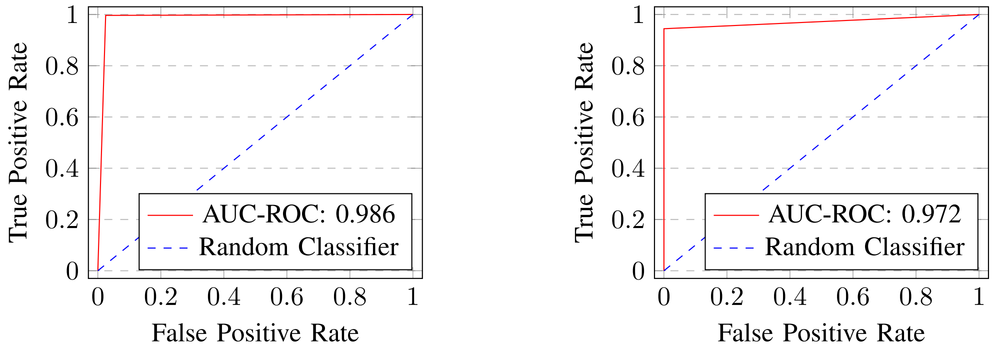
Fig. 3. Word Co-occurence networks for investigation search terms

		Predicted	
		Non-Cyber.	Cyber.
True	Non-Cyber.	0.975	0.025
	Cyber.	0.004	0.996

(a) Confusion Matrix for Pre-Trained Model

		Predicted	
		Non-Event	Event
True	Non-Event	0.938	0.062
	Event	0.00	1.00

(b) Confusion Matrix for Final Model



(c) ROC Curve for Pre-Trained Model

(d) ROC Curve for Final Model

Fig. 4. Performances of Cyber Event Tweet Detection

and 97.6% for precision, recall, F_1 score, and specificity respectively. The results show that the model can very accurately identify general cybersecurity tweets. The final cyber event tweet detection model achieved a 97.2% AUC-ROC score (Fig. 4d) and 96.9% accuracy on its testing set. After inspecting the confusion matrix in Fig. 4b, it is clear that the model can correctly classify cyber event tweets, as it receives a recall of 1. However, a small portion of tweets that did not pertain to a cyber event were not discarded, thus generating false positives and resulting in a 94.1% precision. The model got an F_1 of 97.0% and specificity of 95.2%. Overall, the performance is good, although not as good as the pre-trained model due to insufficient cyber event tweets in the training data.

C. Named Entity Recognition

Overall, the method received 88.55%, 89.70%, and 87.43% for F_1, precision, and recall respectively. However, further examination revealed that entity labels crucial to the function of the scheme, such as CVE, MalwareType, and AttackType, all receive F-scores above 94.0%. The labels where the model's ability is lacking (F_1 below 85%) are Org, Product, and Cardinal. A full breakdown of the NER model's performance by label is provided in Table IV.

TABLE IV
NER Performance Per Label

Label	Precision	Recall	F_1 Score
CVE	98.85	98.85	98.85
MalwareType	94.74	96.64	95.68
Ordinal	100.00	90.00	94.74
AttackType	98.32	90.70	94.74
Money	94.74	90.00	92.31
NORP	97.83	84.91	90.91
GPE	92.02	89.82	90.91
Percent	81.82	90.00	85.71
Org	82.59	76.34	79.34
Product	75.00	84.00	79.25
Cardinal	71.11	78.05	74.42

D. Entity Clustering

To evaluate the efficacy of clustering named entities by their co-occurrence for cyber event detection (see Fig. 3a for examples), we randomly selected nine event clusters as test

TABLE V
EVENT CLUSTER VALIDITY

Central Node	#Edges	Percentage of Correct Edges
CVE-2022-30129	2	100.00%
CVE-2022-29866	6	100.00%
Spotify	3	100.00%
Facebook	10	100.00%
Apple	65	96.92%
Android	36	88.89%
Microsoft	21	76.19%
MacOS	69	72.46%
cve-2022-32893	19	68.42%

cases. By manually comparing each edge within the cluster network against what is available online and our own knowledge, we were able to gauge the relevancy/correctness of the edges/connections between different nodes. Since the full clusters contain too many edges to manually verify, for each cluster we only checked the edges connected to the central node (i.e., the node with most edges in the cluster). As shown in table V, the percentage of correct edges ranges from under 70% to 100%. Overall, we found the percentage of correct edges returned by our solution to have a weighted average of approximately 84.41%, showing a good accuracy.

V. Case Studies

A. Cyber Event Discovery

The first use case is the discovery of new cyber events. Under these circumstances, the user would use the program's default search parameters to capture the largest breadth of Twitter chatter possible. Security officers could in turn use this information to defend their systems, potentially even before major security information providers begin reporting on it. This case study was conducted on May 23, 2022 at approximately 10 a.m. After an initial run, the program revealed 14 events that might be occurring. The top 5 events with the highest weight are provided in Table VI along with their diffusion and spam indices, and a notable tweet. With the tweet column, we are able to examine the events with better context. In fact, all 5 clusters contain a CVE and four of the five refer to a specific version number for the vulnerable asset. This may not always be the case, however. Thus, there are instances in which entity clusters may need further investigation, which is discussed further in section V-B. We were able to find corroborating articles for each of the clusters [21, 22, 23].

B. Cyber Event Investigation

Under some circumstances, the user may already know of an event and would like to investigate it. To simulate this use case, we focus on the "log4j" vulnerability that affected many companies in early 2022. The vulnerability was first disclosed in early December 2021 as a remote code execution (RCE) bug with a critical severity classification. Less than a day later, the vulnerability was being exploited by multiple threat actors, such as the Mirai botnets. Information like this is crucial for security operators to monitor their systems, especially considering Cloudflare and Cisco suffered attacks more than a week before the vulnerability was publicly disclosed [24]. With our proposed solution, professionals would have had access to information about this event and its implication, regardless of whether or not the provider had reported on it.

To begin our investigation we searched "log4j" and found 1017 tweets. Passing them through our cyber event detection model resulted in 487 tweets. An initial analysis is shown in Table VII. The word co-occurence network, provided in Figure 3b, shows a strong correlation between log4j, VMware, Apache, and RCE. There is also a number of CVE tags present in the network. To continue our investigation, we included CVE-2021-44228 in the search parameters as it has the highest co-occurence with "log4j". This small change dramatically reduced the network as shown in Figure 3c. For the final run we wanted to see if we could gather more information about what Lazarus may refer to. With the addition of this entity, the program was able to find a tweet containing all three entities, provided in Table VII. Of the entities found, North Korea was among the nation-states seen exploiting log4j [25], CVE-2021-44228 was the first of the CVEs revealed during this event [24], and Lazarus was a threat group targeting VMware servers [26]. In this process, many of the key entities of this event were identified in less than a minute.

C. Vulnerability Prioritization

In this case study, we adopted the perspective of a security operator who just became aware of two vulnerabilities in our company's system, namely CVE-2022-26862 and CVE-2022-26717, but we did not know how to prioritize patching them. Our OSINT tool can provide intelligence that complements existing AI-based vulnerability management solutions [27, 28, 29, 30]. Specifically, we followed a similar methodology as section V-B. After passing CVE-2022-26862 through our OSINT system, we found that the diffusion and spam indices of this topic are rather low, 0.6 and 0.467, meaning that the conversations about this vulnerability are fairly isolated. Our system also found two notable tweets showing that the vulnerability affects certain versions of Dell BIOS, and allows a locally authenticated malicious user to bypass security controls.

An initial run of our OSINT tool found a strong correlation between CVE-2022-26717 and "safari". Once we included "safari" in our search terms, we found diffusion and spam indices of 1.0 and two notable tweets. The first provided a link to an exploit dated May 8th. The second also linked this exploit, but provided the patched version number as well.

We found CVE-2022-26717 the more dangerous vulnerability and should be prioritized as such. Not only were there fewer conversations about CVE-2022-26862, it also required a local user. This concludes our investigation. In fact, CVE-2022-26717 is much more widespread and has a publicly available exploit circulating. Thus, it indeed deserves a higher priority, validating our tools recommendation.

VI. Conclusions

We proposed a scheme of discovering and analyzing cyber events through the use of OSINT based on Twitter data. Through a multi-model pipeline, it can filter the twitter stream to identify cyber event tweets, extract the valuable information

TABLE VI

TOP FIVE ENTITY CLUSTERS WITH THEIR ASSOCIATED OUTPUTS. TWEETS ARE PROVIDED WITHOUT MODIFICATION.

Entity Cluster	Diff.	Spam	Sample Tweet
cve-2022-20821, xr, cisco	1.0	1.0	Cisco Warns of Exploitation Attempts Targeting New IOS XR Vulnerabi... (Securityweek) The flaw, tracked as CVE-2022-20821, was discovered by Cisco during the resolution of a s...
cve-2022-29599, maven, commandline	1.0	1.0	CVE-2022-29599 : In #Apache Maven maven-shared-utils prior to version 3.3.3, the Commandline class can emit double-quoted strings without proper escaping, allowing shell injection attacks....
cve-2021-30028 range wi-fi	1.0	1.0	Emerging Vulnerability Found CVE-2021-30028 - SOOTEWAY Wi-Fi Range Extender v1.5 was discovered to use default credentials (the admin password for the admin account) to access the TELNET service, allowing attackers to erase/read/write the firmwar
cve-2021-42863 jerryscript	0.7	0.583	Potentially Critical CVE Detected! CVE-2021-42863 A buffer overflow in ecma_builtin_typedarray_prototype _filter() in JerryScript version fe3a5c0 allows an attacker to con... CVSS: 8.80 #CVE #CyberSecurity
cve-2022-1816 zoo	1.0	1.0	CVE-2022-1816 A vulnerability, which was classified as problematic, has been found in Zoo Management System 1.0. Affected by this issue is /zoo/admin/public_html/view_accounts?type=zookeeper of the content module....

TABLE VII

DIFFUSION AND SPAM INDEX VALUES FROM EXAMPLE ENTITY CLUSTERS. TWEETS ARE PROVIDED WITHOUT MODIFICATION.

Entity Cluster	Diff.	Spam	Sample Tweet
log4j vmware us	0.923	0.885	#Log4Shell reminded us how important it is to have a trusted open-source software provider. This blog post by @mpermar explains the vulnerability in detail and how VMware Application Catalog brings confidence to developers and operators. https://t.co/OhLawPexfJ #Log4j #CVE
cve-2021-4422 lazarus north	1.0	1.0	North Korea-linked group Lazarus is exploiting the Log4J RCE #vulnerability (CVE-2021-44228) to compromise VMware Horizon servers. If not already do the right thing: Patch yours! #becybersmart https://t.co/EYFNTcEfOz

about specific cyber events, and validate their veracity. The evaluation results showed that the approach is feasible in practice. Three case studies were presented to show its usefulness.

ACKNOWLEDGMENT

This work is supported in part by the National Science Foundation under award 1751255 and by the Department of Energy under award DE-CR0000003.

REFERENCES

[1] Davey Winder. *Microsoft Security Shocker As 250 Million Customer Records Exposed Online*. Forbes.

[2] Gloria Gonzalez, Ben Lefebvre, and Eric Geller. *'Jugular' of the U.S. fuel pipeline system shuts down after cyberattack*. POLITICO.

[3] *2022 SonicWall Cyber Threat Report*. 2022.

[4] Dmytro Lande, Igor Subach, and Alexander Puchkov. "A system for analysis of big data from social media". In: *Information & Security* 47.1 (2020), pp. 44–61.

[5] Agata Ziółkowska. "Open Source Intelligence (OSINT) as an Element of Military Recon". In: *Security and Defence Quarterly* 2 (2018), pp. 65–77.

[6] Abdullah Talha Kabakus and Resul Kara. "A survey of spam detection methods on twitter". In: *International Journal of Advanced Computer Science and Applications* 8.3 (2017).

[7] Anna Sapienza et al. "DISCOVER: Mining Online Chatter for Emerging Cyber Threats". In: *The Web Conference 2018*. 2018, pp. 983–990.

[8] Ariel Rodriguez and Koji Okamura. "Social media data mining for proactive cyber defense". In: *Journal of Information Processing* 28 (2020), pp. 230–238.

[9] Sudip Mittal et al. "CyberTwitter: Using Twitter to generate alerts for cybersecurity threats and vulnerabilities". In: *2016 IEEE/ACM Int'l Conf. on Advances in Social Networks Analysis and Mining*.

[10] Satyanarayan Raju Vadapalli, George Hsieh, and Kevin S Nauer. "Twitterosint: automated cybersecurity threat intelligence collection and analysis using twitter data". In: *International Conference on Security and Management (SAM)*. 2018, pp. 220–226.

[11] Avishek Bose et al. "A novel approach for detection and ranking of trendy and emerging cyber threat events in twitter streams". In: *Int'l Conf. on Advances in Social Networks Analysis and Mining*. 2019.

[12] Nuno Dionísio et al. "Cyberthreat detection from twitter using deep neural networks". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–8.

[13] Alec Go, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision". In: *CS224N project report* 1.12 (2009).

[14] Natalie Friedrich et al. "Adapting sentiment analysis for tweets linking to scientific papers". In: *CoRR* abs/1507.01967 (2015).

[15] Sai Vikram Kolasani and Rida Assaf. "Predicting stock movement using sentiment analysis of Twitter feed with neural networks". In: *Journal of Data Analysis and Information Processing* 8.4 (2020), pp. 309–319.

[16] Dilara Torunoğlu et al. "Wikipedia based semantic smoothing for twitter sentiment classification". In: *2013 IEEE INISTA*, pp. 1–5.

[17] Catherine Lee, Jacob Shiff, and Sridatta Thatipamala. *Predicting US Political Party Affiliation on Twitter*. Stanford University, 2018.

[18] Puti Cen. "Predicting Consumers' Brand Sentiment Using Text Analysis on Reddit". undergraduate. University of Pennsylvania, 2020.

[19] Dhwani Dholakia et al. "HLA-SPREAD: a natural language processing based resource for curating HLA association from PubMed abstracts". In: *BMC genomics* 23.1 (2022), pp. 1–14.

[20] T.K. Ashwin Kumar, Prashanth Kammarpally, and K.M. George. "Veracity of information in twitter data: A case study". In: *2016 International Conference on Big Data and Smart Computing*, pp. 129–136.

[21] Sergiu Gatlan. *Cisco urges admins to patch IOS XR zero-day exploited in attacks*. BleepingComputer. May 2022.

[22] *Security Bulletin 08 Jun 2022*. Singapore Computer Emergency Response Team. June 2022.

[23] *Security Bulletin 25 May 2022*. Singapore Computer Emergency Response Team. May 2022.

[24] Raphael Hiesgen et al. "The Race to the Vulnerable: Measuring the Log4j Shell Incident". In: (2022).

[25] *Hackers used the Log4j flaw to gain access before moving across a company's network, say security researchers*. ZDNet.

[26] Pierluigi Paganini. *North Korea-linked Lazarus APT uses Log4J to target VMware servers*. Security Affairs.

[27] Philip Huff et al. "A Recommender System for Tracking Vulnerabilities". In: *Int'l Conf. on Availability, Reliability and Security (ARES)*. ACM, 2021.

[28] Kylie McClanahan and Qinghua Li. "Automatically Locating Mitigation Information for Security Vulnerabilities". In: *IEEE SmartGridComm*. 2020, pp. 1–7.

[29] Fengli Zhang et al. "A Machine Learning-based Approach for Automated Vulnerability Remediation Analysis". In: *IEEE Conference on Communications and Network Security (CNS)*. 2020, pp. 1–9.

[30] Philip Huff and Qinghua Li. "Towards Automated Assessment of Vulnerability Exposures in Security Operations". In: *Int'l Conf. on Security and Privacy in Communication Networks (SecureComm)*. 2021.