Partition-then-Overlap Method for Labeling Cyber Threat Intelligence Reports by Topics over Time

Ryusei NAGASAWA†, Nonmember, Keisuke FURUMOTO††, Makoto TAKITA†††, Members, Yoshiaki SHIRAISHI††††, Senior Member, Takeshi TAKAHASHI†††, Member, Masami MOHRI††††, Senior Member, Yasuhiro TAKANO†, Member, and Masakatu MORII††, Fellow

SUMMARY The Topics over Time (TOT) model allows users to be aware of changes in certain topics over time. The proposed method inputs the divided dataset of security blog posts based on a fixed period using an overlap period to the TOT. The results suggest the extraction of topics that include malware and attack campaign names that are appropriate for the multi-labeling of cyber threat intelligence reports.

key words: topic model, cyber threat intelligence, text mining, multi-labeling, security blog posts

1. Introduction

Security blog posts published by security vendors include an analysis of threat information and alerts. Security blog posts are useful because they suggest methods to prevent and respond to cyberattacks.

However, the number of posts continues to increase day by day, and their contents change over time. It is not easy to find security blog posts that contain the desired content under such circumstances.

Security blog posts are occasionally labeled to aid in information retrieval, although the criteria for labeling are not standardized and vary from publisher to publisher. In addition, some posts do not have labels. In summary, there are no unified methods to search for desired information from a wide range of security blog posts.

It is necessary to assign appropriate multi-labels depending on their content to make it easy for security operators to obtain information from the security blog posts, which are increasing in number every day.

The goal of our study is to assign appropriate labels to security blog posts. This labeling is intended to allow security operators to collect relevant information associated with the subject of a search. Therefore, it is necessary to assign key phrases contained in a document to other documents as labels. Typical keyword extraction methods [1]–[3] cannot be used when attempting to achieve this goal because they extract keywords from the words in a document; thus, it is impossible to assign keywords that are not included in the document as labels. Named entity extraction [4], [5] cannot be used because it extracts key phrases from a document, whether it is supervised or unsupervised. In addition, when we extract named entities using supervised machine learning, labeled training data are required. However, publicly available trained models do not support security domain-specific key phrases (e.g., malware and attack campaign names). Thus, we cannot extract named entities appropriate for the labels.

Topic models are proposed as a statistical modeling method to obtain information from a large and heterogeneous set of documents. In addition to the LDA [6], which is a representative topic model, keyword extraction models based on LDA [7] and entity topic models (ETM) [8] have been developed. The multiple labels that meet the purpose of our study can be assigned to a document by understanding its topic.

The tendency of certain topics to occur in numerous document sets, including security blog posts, changes over time. However, the general topic model may result in unclear and suboptimal topics because it does not consider topic estimation.

The topics over time (TOT) model [9], which is a topic model that explicitly models time, is proposed to grasp topics in dynamic documents. The topics occurring in each document can be mapped in a time series using the TOT model.

The goal of this study is to assign appropriate labels to security blog posts. Our approach to meet this goal is as follows. First, we apply the security blog posts to the TOT model and generate time-sensitive topics. Next, we extract words with high compositional proportions from the generated topics and determine appropriate labels to be assigned to the posts.

However, it is unlikely that key phrases suitable as labels can be extracted when a dataset is entered as a batch into the above topic model, which includes TOT. Although multi labeling for security blog posts expects labels to include malware and attack campaign names, many of these named entities are not included in the documents. Therefore, to achieve the goal of this study, the extraction of a...
security domain-specific vocabulary that appears locally is a problem to be solved.

Another difficulty with this problem is that common phrases in the security domain remain, even after preprocessing security blog posts with common natural language processing tools such as NLTK [10] to remove frequent words and stop words. These common phrases are widely distributed across the entire document set, whereas malware names and attack campaign names are distributed locally. It is therefore necessary to find phrases with a high frequency of occurrence locally, and not only phrases that are widely distributed throughout the document set. Partitioning of the dataset is expected to increase the likelihood that locally frequent phrases will be used in the organization of topics.

When a dataset is partitioned without an overlap period, if there is a concentration of important words near the partitioning points, these words will be divided. If we input the segmented data into the TOT, there is a possibility that important words will not be captured, and the topics related to these words will not be formed. The proposed method therefore divides the dataset by a fixed period with an overlap period and inputs it to TOT. It prevents malware names and attack campaign names from being buried by common phrases.

2. Proposed Method: Partition-then-Overlap for Labeling by TOT

2.1 Topics Over Time (TOT)

Before introducing the TOT model [9], our notations are summarized in Table 1; the graphical model representation of the TOT models is shown in Fig. 1. The TOT model is a topic model based on LDA [6]. The TOT model considers not only the co-occurrence information of words but also information on the document’s published time in estimating topics. In other words, TOT prevents confusion with co-occurrence patterns and the occurrence of ambiguous and suboptimal topics by mapping the topics in a document to the time series.

TOT is a generative model of timestamps and words in timestamped documents. The TOT generative process is as follows. First, $T$ multinomials $\Phi$, are drawn from a Dirichlet prior $\beta$ for each topic $z$. For each document $d$, a multinomial $\theta_d$ is drawn from a Dirichlet prior $\alpha$. Next, for each word $w_{di}$ in document $d$, a topic $z_{di}$ is drawn from a multinomial $\theta_d$, a word $w_{di}$ is drawn from a multinomial $z_{di}$, and a timestamp $t_{di}$ is drawn from Beta $\psi_{zd}$.

2.2 Partition-then-Overlap TOT

In the set of $M$ documents to be analyzed, we define a dataset with an ordered time set, $T = \{t_1, t_2, \ldots, t_M\}$, and an ordered document set, $D = \{d_1, d_2, \ldots, d_M\}$, the lists of which are ordered by their publication dates. Here, $f: T \to D$ is an order-preserving bijection. With the proposed method, we divide the ordered set $T$, $D$ into $N$ pieces by specifying a fixed division period and overlap period.

For $i = 1, 2, \ldots, N$, we denote $p_1, p_2, \ldots, p_2N$ as the dates at both ends of the partition period. We then denote by $T_{PI} = \{t_i \in T | t_{2i-1} < t_i \leq t_{2i}\}$ as the set of publication dates of documents that exist in the partitioning period $(p_{2i-1}, p_{2i})$. Here, $T_{PI}$ is a subset of $T$ and is an ordered set. We denote $P = \{(p_1, p_2), (p_3, p_4), \ldots, (2N-1, 2N)\}$ as the set of partition periods. We denote by $D_P$, the set of documents $d_i$ in $D$ corresponding one-to-one to $t_i$ in $T_{PI}$, during each partitioning period. If $P_{i+1} < P_{2i}$, partitioning periods $(p_{2i-1}, p_{2i})$ and $(p_{2i+1}, p_{2i+2})$ have overlapping dates during the period $(p_{2i}, p_{2i+1})$. That is, partitioned datasets have fixed overlap periods. Figure 2 shows an idea of the partition-then-overlap method.

We denote the ordered set families of the partitioned time set $T_{PI}$ and partitioned document set $D_P$, as $T_P = \{T_{P1}, T_{P2}, \ldots, T_{PN}\}$, and $D_P = \{D_{P1}, D_{P2}, \ldots, D_{PN}\}$, respectively.

We preprocess the elements of the ordered set family $D_P$ as

![Fig. 1 TOT graphical model](image1)

![Fig. 2 Idea of the Partition-then-Overlap method](image2)

| Symbol | Description |
|--------|-------------|
| $\gamma$ | number of topics |
| $M$ | number of documents |
| $N_d$ | number of word tokens in document $d$ |
| $\theta_d$ | multinomial distribution of topics specific to document $d$ |
| $\Phi$ | multinomial distribution of words specific to topic $z$ |
| $\psi_z$ | beta distribution of time specific to topic $z$ |
| $z_{di}$ | topic associated with $i$th token in document $d$ |
| $w_{di}$ | $i$th token in document $d$ |
| $t_{di}$ | timestamp associated with $i$th token in document $d$ |
• Extract compound terms using TermExtract [11]
• Remove stop words
• Remove words with an occurrence rate of above 50%
• Remove words containing numbers, symbols, and quotation marks
• Remove words with less than four appearances

We denote by \( W_i = \{ w_1, w_2, \ldots, w_N \} \) the set of words made up of the remaining words and compound words after the preprocessing in a single document. We define the set of partitioned documents after preprocessing \( D_{P_i} \) as \( \Delta P_i = \{ W_1, W_2, \ldots, W_N \} \), and then the ordered set family of \( \Delta P_i \) is \( \Delta P = (\Delta P_1, \Delta P_2, \ldots, \Delta P_N) \). Combining the ordered set families \( \Delta P \) and \( T_P \), we construct the input set family as

\[
(\{\Delta P_1, T_P\}, \{\Delta P_2, T_P\}, \ldots, \{\Delta P_N, T_P\}).
\]

The input set families are inputted into TOT by specifying the number of topics \( \gamma \), the hyperparameters \( \alpha \) and \( \beta \), and the number of iterations \( \delta \). TOT outputs \( \theta \), \( \phi \), and \( \psi \) for \( \{\Delta P, T_P\} \). For an input set family, we obtain the output set family \( (\{\theta_1, \phi_1, \psi_1\}, \{\theta_2, \phi_2, \psi_2\}, \ldots, \{\theta_N, \phi_N, \psi_N\}) \).

From the topics obtained for each partitioning period, we estimate the characteristic topics containing key phrases that can be labeled. For topic estimation, we extract keywords with a high probability of belonging to topics using \( \phi \) from the set of outputs of a partitioned period, \( \{\theta_i, \phi_i, \psi_i\} \).

Defining \( K \) as the total number of words in the partitioned word set \( \Delta P \), \( \phi_i \) can be expressed as

\[
\phi_i = \begin{bmatrix}
    h_{11} & h_{12} & \ldots & h_{1\gamma} \\
    h_{21} & h_{22} & \ldots & h_{2\gamma} \\
    \vdots & \vdots & \ddots & \vdots \\
    h_{K1} & h_{K2} & \ldots & h_{K\gamma}
\end{bmatrix}
\]

(0 < i ≤ N)

The columns of the two-dimensional array \( \phi_i \) represent the multinomial distribution of topics specific to words, and the rows represent the multinomial distribution of words specific to a topic. Here, \( h_{kj} \) (0 < \( k \leq K \)) represents the probability that word \( k \) belongs to a topic \( j \). For each column vector of \( \phi_i \), Z words with the highest value of \( h_{kj} \) are extracted as the key phrases representing the topic. Appropriate labels were manually selected from the words that constitute the topics. In our study, we use malware names and attack campaign names as the leading labels.

3. Applying TOT to Security Blog Posts

The dataset consists of 2386 security blog posts from January 1, 2017 to December 31, 2019, collected from the blog pages of eight security vendors (Netscout, Barracuda, Cisco, Druva, FireEye, Paloalto, NortonLifeLock, and TrendMicro).

The implementation of the proposed method is based on Python3 with pandas, numpy, and scipy using a TOT code [12]. The experimental environment was Ubuntu 18.04 in Intel Core i7-7820X and 64 GB of memory.

The dataset was divided into a division period of 6 months and an overlap period of 3 months. As the initial setting, the number of topics \( \gamma = 8 \); the Dirichlet prior distribution hyperparameters \( \alpha \) and \( \beta \) were set to 50/\( \gamma \) and 0.1, respectively; and the iteration number \( \delta = 500 \). The ordered set families \( \Delta P \) and \( T_P \) defined in Sect. 2.2 were entered into TOT. Table 1 shows the results.

In Table 2, we extracted the five words with a high proportion of topics for the number of topics specified for each partitioning period. Malware names and attack campaign names were underlined, and topics containing the words were enclosed in a bold frame.

From Table 2, we were able to grasp malware names and attack campaign names that can be labeled for each partition period. Malware names such as “wannacry” have long appeared, and malware names such as “bad rabbit” and “samsam” appeared only during a single partitioning period.

We examined the relationship between the labels obtained in our experiments and the malware and attack campaigns that were prevalent at the same time. First, Meltdown and Spectre are both CPU vulnerabilities, and the article was published in November 2017 [13]. Since then, security updates for Meltdown and Spectre were made by Apple, Google, Microsoft, and other companies in March of 2018.

The experimental results in Table 1 show that the Meltdown and Spectre labels appeared during the periods of “October 2017 to March 2018” and “January to June 2018.” In addition, “mirai” is a malware that emerged around November 2016, and mirai variants have been active since December 2017 [14], [15]. The experimental results in Table 2 show that the mirai label appeared during the periods of “October 2017 to March 2018,” “January to June 2018,” and “April to September 2018.” As shown in the above examples, the labels that appeared in the experimental results in Table 2 are thought to capture the malware and attack campaigns that were prevalent during the same period.

4. Evaluation of Partition-then-Overlap Method

We addressed the comparative results of the proposed method against the approach (batch method) in which the original dataset is entered into the TOT [9] and LDA [6] models in a batch. LDA was implemented using Gensim [16], an open-source library for unsupervised topic modeling and natural language processing. In the batch method, we set the number of topics to 46; the number of topics was derived using three ratings: KL divergence, pairwise cosine distance, and coherence. For hyperparameters \( \alpha \) and \( \beta \), we used the same values as in the proposed method applied to TOT and the default values of Gensim in LDA. The results when entering the datasets into TOT and LDA in batches are shown in Tables 3 and 4.

We compared the extracted malware names and attack campaign names from the results of the proposed method when applying the TOT (with the batch method) and LDA (with the batch method). Malware names and attack campaign names are useful as labels for security blog posts and significantly affect the accuracy of the labels. Therefore, we
As indicated in Tables 2, 3, and 4, we extracted 12 malware and attack campaign names when applying the proposed method, 10 names when applying TOT (with the batch method), and 6 names when applying LDA (with the batch method). The proposed method extracted the greatest number of malware and attack campaign names because it could extract the names of the malware and attack campaigns that appear locally in the document set. In particular, “bad rabbit,” “samsam,” “notpetya,” and “meltdown” are only extracted by the proposed method. Documents containing them were published within a short period of time. Using the proposed method, we extracted the names that appeared locally in the document set. Thus, we could extract more key phrases as label candidates than either the TOT (with the batch method) or LDA (with the batch method) models. The proposed method is effective in appropriately labeling security blog posts, which is the objective of our study.

However, with the batch method, we extracted the malware and attack campaign names that were not extracted using the proposed method, i.e., “emotet,” “trick-

### Table 2  Result of entering partitioned datasets into TOT ($\gamma = 8$, $\alpha = 50/\gamma$, and $\beta = 0.1$)

| 2017/1-6 | 2017/4-9 | 2017/7-12 | 2017/10-2018/3 | 2018/1-6 | 2018/4-9 | 2018/7-12 | 2018/10-2019/3 | 2019/1-6 | 2019/4-9 | 2019/7-12 |
|----------|----------|-----------|----------------|----------|----------|-----------|----------------|----------|----------|----------|
| **wanney** | **wanney** | iot device | **wanney** | ise | **wanney** | classifier | com | threat hunting | tec |
| exploit kit | tcp | packet | threat grid | smbs | iot | threat grid | rdp | threat detection | directory |
| botnet | petya | nist cf | bitcoin | **wanney** | healthcare organization | umbrella | smbh | registry | modr | disaster recovery |
| cser | smbh | **wanney** | bad rabbit | healthcare device | medical device | threat hunting | certificate | scam | cybersecurity team | vms |
| com | steal data | worm | dashboard | transparent | healthcare industry | imperative | **samsam** | smbh | active directory | ciso |
| **thamos** | security vendor | ise | cash | federal agency | carrier | blockchain | federal agency | iot device | cash | xdr |
| shellcode | cloud ready | non-profit | cloud apps | vms | apple | carrier | devops | federal agency | dlp | backup system |
| hitrust | map | identity theft | equifax | salesforce | installation | transaction | endpoint detection | threat hunting | business unit | siem |
| middle east | maps | iot device | disclose | cloud environment | china | byte | cdm | threat response | workforce | magecart |
| python | hybrid environment | trustsec | dropbox | saas | iot device | middle east | macro | disclose | disclose | prevention |
| **wanney** | linux | petya | shopper | threat grid | google play | federal agency | small business | red team | tweet | volatility |
| petya | exploit kit | bitcoin | uber | appdata | phone supplier | maps ai model | accuracy | iran | dlp |
| fbi | petya | **wanney** | cio | javascript | mobile security | prevention | cybersecurity team | iot | com | aps |
| smbh | ukraine | ddos attack | holiday season | sep | social network | cio |law | tax | dataset | webinar |
| **kudr** | kit | notpetya | etc | ciso | threat detection | attendee | germany | maps | align | workforce |
| certificate | macro | linux | sep | macro | appdata | atm | scam | malware ati | dsa | shellcode |
| dip | india | bot | federal agency | iot device | hash | small business | scanner | mean time | maps | binary |
| online safety | proxy | security vendor | javascript | vmsware | javascript | installation | botnet | fraud | webinar | utility |
| proxy | align | webinar | ise | binary | macro | apple | chinese | cloud device | backup system | byte |
| confidential data | legacy | likelihood | mobile security | rdp | dli | ciso | bot | promem | cot | cloud grid |
| medical device | nst cf | drmac | iot device | **mirai** | blockchain | scam | atm | cash | email thread | threat hunting |
| apple | healthcare organization | google play | **mirai** | iot device | **mirai** | exploit kit | theft | rto | query | workplace |
| ciso | cser | sender | dss | dss | transaction | tax | emea | saas | response team | ise |
| trump administration | congress | smbh | vms | ddos attack | crypocurrency mining | japan | steal data | ciso | downtime | local government |
| wmi | hitrust | console | default password | connect device | tax | governance | transaction | security program | conjunction | dao |
| keem lab | bot | saaf | dmec | scanner | threat grid | ise | ise | active directory | red team | maps |
| team sniper | android device | shopper | meltdown | meltdown | binary | vmsware | tax | malware analysis | devops | pillar |
| incident security | ddos attack | invoice | controller | google play | com | api | maps | destination | ciso | meantime |
| uaf | dip | implementation | healthcare organization | **spectre** | ciso | cwp | api | sensor | tips | it team |
| bot | google play | redirect | spectre | shellcode | saas | saas | api | dip | appliance | professional |
| conference | medical device | controller | bot | blockchain | **hec** | **hec** | iot device | devops | scam | scam |
| india | shellcode | kit | directory | hash | maps | dlp | rdp | china | **hec** | scanner |
| threat grid | iot device | cloud apps | temp | bitcoin | scam | devops | com | bot | scanner | **hec** |
| amp | endpoint protection | it organization | folder | minor | smbs | iot | registry | login credential | fraud | dsa |
| api | worm | business leader | packet | transaction | map | entity | **hec** | guidance | gmail | potential victim |
| dpu | bitcoin | equifax | scanner | folder | federal agency | iot device | iot | source code | source code | cybersecurity team |
| cif | authority | south korea | google play | cloud apps | dip | com | umbrella | carbon | xdr | threat detection |
| macro | cloudsec | com | middle east | cash | ymlfiter | **samsam** | persistent threat | scanner | carbon | linux |
| tcp | non-profit | cloudsec | transparency | apple | simulation | bot | threat hunting | scam | siem | google play |
| mist cf | threat grid | neasm | transparent | mobile security | ise | binary | threat response | binary | ciso | modr |
### Table 3
Result of entering the dataset into TOT in a batch ($\gamma = 46, \alpha = 50/\gamma$, and $\beta = 0.1$)

| salesforce | persistence | small | macro | byte | word | vmware | dme | api |
|------------|-------------|-------|-------|------|------|--------|-----|-----|
| trend | ad | admission | threats | cio | ciso | team | cloud | security patch | source code | epr | apif | dme | api |
| threat | cloud | security | team | cloud | security | patch | source | code | epr | api | apif | dme | api |
| threat | cloud | security | team | cloud | security | patch | source | code | epr | api | apif | dme | api |
| threat | cloud | security | team | cloud | security | patch | source | code | epr | api | apif | dme | api |
| threat | cloud | security | team | cloud | security | patch | source | code | epr | api | apif | dme | api |

### Table 4
Result of entering the dataset into LDA in a batch ($\gamma = 46, \alpha = 1/\gamma$, and $\beta = 1/\gamma$)

| apif | federal | certificate | box | comobject | redteam | security | manage | iran | maps |
|------|---------|-------------|-----|-----------|---------|---------|--------|------|------|
| salesforce | cloud | security | team | cloud | security | patch | source | code | epr | api | apif | dme | api |
| threat | cloud | security | team | cloud | security | patch | source | code | epr | api | apif | dme | api |
| threat | cloud | security | team | cloud | security | patch | source | code | epr | api | apif | dme | api |
| threat | cloud | security | team | cloud | security | patch | source | code | epr | api | apif | dme | api |
| threat | cloud | security | team | cloud | security | patch | source | code | epr | api | apif | dme | api |
bot,” and “triton” because the documentation on these malware names was widely distributed over a five-part period (1.5 years). Moreover, unlike the documents on “wannacry” and “bec,” which are also widely distributed, there are few documents on these types of malware. Therefore, the proposed method, which is a partitioning approach for extracting local key phrases, could not capture “emotet,” “trickbot,” or “triton.”

5. Conclusions

In this paper, we proposed a method for constructing data into TOT that allows us to extract distinctive labels from security blog posts. The key idea is to not enter the datasets into TOT in batches (using a batch method), but to instead enter the partitioned dataset into TOT with a fixed period of overlap. Our proposed method captures malware and attack campaign names that appear locally and extracts key phrases that can be more useful labels than when applying the batch method. In addition, by adding overlaps, we could extract malware names such as “bad rabbit,” which had been buried when using the partitioning method. Therefore, we can state that the partition-then-overlap method is useful for extracting key phrases that can be used as labels. By using the results of both the batch method and the partition-then-overlap method, we obtain more appropriate search results. It is a future task to confirm the usability of a search system of the security blog post with these labels through user experiments.

Acknowledgments

This research was conducted under a contract of “Research and development on IoT malware removal/make it non-functional technologies for effective use of the radio spectrum” among “Research and Development for Expansion of Radio Wave Resources (JPJ000254),” which was supported by the Ministry of Internal Affairs and Communications, Japan.

References

[1] S. Rose, D. Engel, N. Cramer, and W. Cowley, “Automatic key-word extraction from individual documents,” in Text Mining: Applications and Theory, eds. M.W. Berry and J. Kogan, Wiley Online, pp.1–20, 2010.
[2] R. Mihalcea and P. Tarau, “TextRank: Bringing order into texts,” Proc. 2004 Conference on Empirical Methods in Natural Language Processing, pp.404–411, 2004.
[3] I.H. Witten, G.W. Paynter, E. Frank, C. Gutwin, and C.G. Neville-Manning, “KEA: Practical automatic keyphrase extraction,” Proc. Fourth ACM Conference on Digital Libraries, pp.254–255, 1999.
[4] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D.S. Weld, and A. Yates, “Unsupervised named-entity extraction from the web: An experimental study,” Artificial Intelligence, vol.165, no.1, pp.91–134, 2005.
[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” Proc. NAACL-HLT 2019, pp.4171–4186, 2019.
[6] D.M. Blei, A.Y. Ng, and M.I. Jordan, “Latent dirichlet allocation,” Journal of Machine Learning Research, vol.3, pp.993–1022, 2003.
[7] J. Park, J. Kim, and J.-H. Lee, “Keyword extraction for blogs based on content richness,” Journal of Information Science, vol.40, no.1 pp.38–49, 2014.
[8] H. Kim, Y. Sun, J. Hockenmaier, and J. Han, “Etm: Entity topic models for mining documents associated with entities,” Proc. 2012 IEEE 12th International Conference on Data Mining, pp.349–358, 2012.
[9] X. Wang and A. McCallum, “Topics over Time: A non-Markov continuous-time model of topical trends,” Proc. 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.424–433, 2006.
[10] Natural Language Toolkit, https://www.nltk.org/.
[11] pytermextract, http://gensen.dl.itc.u-tokyo.ac.jp/pytermextract/. (in Japanese).
[12] Topics Over Time, https://github.com/ahmaurya/topics_over_time
[13] Kernel index [LWN.net] - Meltdown and Spectre, https://lwn.net/Kernel/Index/#Security-Meltdown_and_Spectre.
[14] Rise of One More Mirai Worm Variant, https://www.fortinet.com/blog/threat-research/rise-of-one-more-mirai-variant.
[15] Warning: Satori, a Mirai Branch is Spreading in Worm Style on Port 37215 and 52869, http://blog.netlab.360.com/warning-satori-a-new-mirai-variant-is-spreading-in-worm-style-on-port-37215-and-52869-en/.
[16] Gensim - PyPI, https://pypi.org/project/gensim/.