IoC Stalker: Early detection of Indicators of Compromise

1st Mariella Mischinger
IMDEA Networks Institute and Universidad Carlos III de Madrid
Madrid, Spain

2nd Sergio Pastrana
Universidad Carlos III de Madrid
Leganés, Spain

3rd Guillermo Suarez-Tangil
IMDEA Networks Institute
Madrid, Spain

***Abstract*—Online underground forums are used by cybercriminals to share information and knowledge related to malicious activities. Participants exchange "Indicators of Compromise" (IoCs) within the discussions. These may include Hashes, Domains, URLs, or IPs with potential malicious intent. While Open Source Intelligence (OSINT) eventually identifies these malicious IoCs, it may take an extensive amount of time, sometimes up to years, before they are identified as threats. However, the context in which these IoCs appear, and the information provided through the posts' and authors' context can already offer valuable insights about their malicious nature. Unfortunately, the large amount of unstructured noisy forum data presents a hurdle for automation. In this paper, we address the challenge of automatically distinguishing between posts containing IoCs posing a threat and those being harmless. We design a learning pipeline that does not use features derived from IoCs, enabling a timely identification of novel threats. We operate over a temporal representation of forum data and offer valuable insights into the optimal time window that tracks *concept drift*. We also study which types of IoCs are harder to predict (e.g., IPs) and how *transfer learning* from other types can help to improve their identification. We conduct our analysis on a prominent hacking forum, spanning over 18 years of data, and find that our model can detect IoCs ≈490 days before they appear in OSINT.**

***Index Terms*—Cybercrime, Cyber-Threat Intelligence, Indicator of Compromise, Underground Forums**

I. Introduction

In the dynamic landscape of cybersecurity, the continuous improvements in offensive techniques and tactics require a corresponding evolution of cyber defenses. Many security defenses rely on forensic traces, commonly referred to as Indicators of Compromise (IoCs). These traces play a crucial role for organizations leveraging Open Source Intelligence (OSINT) in detecting signs of potential compromise. The Cyber Threat Intelligence (CTI) community regularly shares IoCs to gain insight into potential threats [1].

Identifying IoCs promptly is crucial for appropriate threat management and incident response, as it allows corporations to proactively block attacks and identify breaches. However, the timing when these IoCs are included in OSINT feeds is unclear. Unfortunately, blocklists are ineffective if they do not offer timely intelligence [2], and there can be a significant delay between the time a malicious service or product is detected in the wild and when the corresponding IoCs are added to OSINT repositories [3]. This is an inherent limitation of relying on OSINT data.

Online underground forums play an important role for cyber-criminals, as they enable the exchange of knowledge and information, as well as the trade of illicit goods and services [4], [5] — including the newest knowledge and cybersecurity exploitation tools [6]–[8]. This gives interesting opportunities for security practitioners and law enforcement to monitor and understand modern threats, as well as their evolution and development [9], [10]. As miscreants engage in these forums, in many cases, they advertise and share IoCs *in the works*, i.e., products or services (e.g., URLs of shops) that are in the process of being monetized [11]. To gain confidence and increase marketing, they often show proof that their product is functional and safe from OSINT detection [1], (e.g., sharing Hashes to show a malware is undetected [12]), before actually trading them. Popular IoCs exchanged (either related to compromised or malicious devices) include IP addresses, Fully Qualified Domain Names (FQDN), URLs [2], or file Hashes (e.g. SHA1, SHA256) [13]. These are widely used by the AntiVirus (AV) and the OSINT industries to identify threats [14]. However, not all discussions on underground forums relate to actionable threats. There are also posts sharing harmless IPs, FQDN, URLs, and Hashes (e.g., actors discussing VPN services, identifying cloud providers, or sharing IP addresses of law enforcement to build deny lists). In this paper, we use the term *artifacts* to refer to *all* these elements regardless of whether they have malicious, or benign attributes. Instead, we refer to the term IoC when an *artifact* is malicious [13]. While previous efforts focus on the extraction of *artifacts* from textual content [13], [15], extracting these is not sufficient and requires further methods to distinguish between actual IoCs and benign *artifacts*.

In this paper, we describe a methodology to detect forum posts that are trading or advertising malicious products and services. Our system learns about potential IoCs in a threat-less, zero-shot manner, e.g., it does not require downloading malware or visiting malicious pages. We prove that our method is generic and adapts to detect different types of IoCs. It also serves as an early-warning system, allowing the detection of malicious content as soon as an IoC first appears in those forums. We note that our system can be seen as an initial scanning pipeline for malicious posts that could undergo further scrutiny.

For our work, we consider four different types of IoCs:

Hashes (MD5, SHA1, and SHA256), IPs, URLs and FQDN. We conduct our research on a prominent multilingual English-Russian hacking forum, whereby the historical data collected spans ≈18 years (2005 to 2023). We focus on actionable IoCs that can eventually be flagged as malicious by prominent OSINT platforms (e.g., VirusTotal [14]) and incorporated into blocklists for reputation-based detection systems [3]. OSINT can not inform about the most recent IoCs unless they have been observed (and reported) in a cyber incident. Therefore, we address this problem by developing a method that detects IoCs as early as they are advertised in underground forums without the need to conduct an exhaustive analysis of the *artifact* itself (i.e., malware sample, exploit, etc.). In our dataset, we find hackers discussing IoCs that remained under the radar of OSINT for years, which confirms the importance of investigating these online communities to gather CTI [7], [9], [11], [16].

We consider a real-world scenario, where we only rely on the information (features and labels) available at the time an *artifact* is posted to avoid temporal and spatial snooping [17]. Through extensive experimental work, we compare different automated approaches to deliver an accurate detection. To provide a generalizable system, we conduct an ablation study over the different types of IoCs. We see how the context in which certain types of IoCs are posted can be leveraged to identify others more accurately. Our findings can be summarized into the following points:

1) By investigating state-of-the-art NLP techniques, we obtain an F1-score of ≈0.8, thus showing the importance of considering context obtained 'in-the-wild' when predicting IoCs.
2) Our classifier can detect posts containing IoCs that take on average ≈490 days to be discovered by OSINT.
3) We show how knowledge transfer facilitates the identification of more complex IoC types and remains resilient to noisy IoCs. Furthermore, we demonstrate how to predict types of IoCs that are not part of the training ground truth.

We motivate our work and identify three requirements for IoC detection in underground forums in §II. Then we present our research questions and methodology for a context-based identification of IoCs in §III, which we evaluate in §IV. We finally validate our IoC detection in §V, and showcase our findings with a case study in §VI.

II. Motivation

Our main motivation is to build a system that detects posts containing IoCs published in online forums. Next, we identify three requirements for such a system and analyze the shortcomings of existing works in the literature.

- **Requirement R1. Need for Threat-less Zero-Shot Learning**. Investigating and analyzing an IoC is a resource-intensive task, requiring experts with specialized skills to react to a dynamic and fast-evolving threat landscape [18]. While threat actors exchange IoCs such as malware Hashes or IP addresses on underground forums, they also discuss harmless *artifacts*, making it even more challenging to identify genuine threats. We hypothesize that information about the malicious nature of the IoC can be inferred from the context of the post where it appears.
- **Requirement R2. Need for an Early Warning System**. The detection system must provide timely intelligence on IoCs. We hypothesize that IoCs are published in forums before they become publicly known through regular OSINT sources, as reported in previous works [1], [19]. We therefore stress the need for a system that can detect novel threats that appear referenced in forums.
- **Requirement R3. Need for a Generalizable Approach**. We desire a generalizable IoC detection method. Such a method would enable the integration of new IoC types and facilitate transfer learning across them. Transfer learning can improve the detection of rare IoCs or those with limited historical data.

Synthesis of Existing Work. In the context of the three requirements above, we observe prior work that partially addresses them, as shown in Table I. First, we see a large body of work offering detection methods based on very specific IoCs posted in comments (R1), mostly targeting malicious URLs [21], [22], [24] — very useful when dealing with SPAM or Phishing. One main limitation of these works is that they use features derived from the *artifact* itself. This can easily be evaded, as we have seen in the case of URLs through the use of shorteners. Instead, works such as in [1] relying on the context are designed to extract *artifacts* in general, but lack a solid validation of the maliciousness of such *artifacts*. Second, other related work explores extracting IoCs from unstructured text, including online forums [20], [23] (R2). However, they are limited in the timely identification of novel threats. In particular, works relying on features from the *artifact* itself such as [20]–[24] are restricted to detecting IoCs of known threats. Furthermore, for an effective early warning system, time dependencies can not be ignored in the analysis, as otherwise, it can lead to temporal snooping [17], [30]. However, we find existing works have not considered this. Third, authors in [28] and [29] present a generalizable IoC detection method (R3). Unfortunately, they rely on previous human action, and thus methods derived from this approach are not systematic [28] or the *artifact* classification relies on external sources [29].

Table I
Existing work and requirement fulfillment.
●= fulfilled, ◐= partly fulfilled, ○= not fulfilled.

	R1	R2	R3	Year	Comments
[20]	●	◐	○	2015	Using URL features.
[21]	●	◐	○	2018	Using URL features.
[22]	○	◐	○	2020	Using URL string and connection features.
[23]	○	◐	○	2021	Using URL string and connection features.
[24]	●	◐	○	2022	Using URL features.
[25]	●	◐	○	2017	Flexibility for other types of IoCs.
[19]	●	◐	○	2019	Flexibility for other types of IoCs. Based on [25].
[26]	●	◐	○	2018	Flexibility for other IoCs. Based on [19], [25].
[27]	●	◐	○	2019	Flexibility for other IoCs. Based on [19], [25].
[28]	●	○	●	2017	Relies on previous human action (Twitter post).
[29]	◐	◐	●	2021	External sources for classification. Latency issues [3].
[1]	◐	●	○	2024	Narrows the focus to appraisal posts.

In addition to the shortcomings discussed, we note that prior work does not address all three requirements at once. When considering textual features previous works rely on classical NLP methods (e.g., TF-IDF for word counts or word embeddings) [1], [19], [20], [25]–[27]. In this paper, we explore the potential of state-of-the-art methods such as sentence transformers, using time-invariant features, and apply them to a prominent underground forum, to address all requirements.

Experimental Analysis of Requirements. We next describe and compare the closest and most recent relevant related works. First, we look at works from Gharibshah et al. [19], [25] which are used as the basis of other works such as [26], [27]. Although these works only investigate IPs, they propose a methodology that relies on post-textual context — partially meeting Req. 1 and Req. 2 — and could be applied to other types of IoCs given the right experimental setting — potentially meeting Req. 3 as well. Thus we conduct a preliminary experiment applying their proposed approach to other types of IoCs, i.e., to see whether it meets Req. 3.

Second, we choose the work from Li et al. [1]. The authors show how to extract IoCs from a special type of post used for 'Appraisals.'[1] They use contextual features to extract the IoCs (potentially meeting Req. 1), and also show that the collected IoCs are posted before they are made public in OSINT sources (meeting Req. 2). Since they focus on appraisal posts, we examine whether their approach can be generalized to the entire forum.

Detecting IPs in underground forums. Gharibshah et al. developed a method to extract and classify malicious IP addresses from underground forums [19], [25]. Since the method does not investigate the *artifacts* themselves, it provides flexibility to be applied to other types of IoCs as well. Therefore, we investigate if we can use this concept to answer our problem. As the work by Gharibshah et al. does not offer an open implementation of their method, we reproduce their work by conducting the following steps:

1) We label posts with VirusTotal [14] and exclude unknown ones (we use the dataset as detailed in §IV-B1 and §IV-B2).
2) We mask the *artifacts*, remove stopwords, lowercase and tokenize the text.
3) As features we consider: 100 Post-Text features generated through TF-IDF and information gain; 11 author-behavior features; latent-post similarities by applying unsupervised clustering to the posts.
4) We then apply classification through Logistic Regression, under two different settings. (a) A random test and train set; (b) Imitating a real-world scenario and considering time dependencies: when predicting posts of year y, we use the information available in years $< y$ as in our work (cf. §III-D2).

When randomly splitting the dataset we report an F1-score of 0.5. When considering a timeline evaluation, we report an average F1-score over all years of 0.4. We therefore conclude that the method does not provide sufficient confidence to answer our problem.

[1]A feedback mechanism where an appraiser receives a free copy of a product or service and writes a post with an initial assessment.

Table II
APPRAISAL POSTS IN OUR DATASET, USING THE METHOD IN [1] TO DISTINGUISH BETWEEN *malicious* AND *not-malicious*.

	Others	Appraisals
not-malicious	1,063,076 (99.97%)	353 (0.033%)
malicious	41,021 (99.74%)	108 (0.263%)

CTI in Appraisal Posts. Li et al. [1] measured the ecosystem of appraisals in underground forums. In their work, they collect IoCs from the appraisals using contextual features. To this end, they conduct a literature review to get a taxonomy of IoCs that can be extracted from forums, including malware and malicious websites. Then, they leverage regular expressions and a custom ML classifier to extract IoCs related to such cybercriminal activities. They validated the model over a reduced sample of 50 posts per category, whereas in our work we rely on well-established ground truth from external OSINT sources. Additionally, their approach is not generalizable and thus does not meet Req. 3. To probe this, we use the same approach the authors did to check for the prevalence of potential IoCs *outside* appraisal posts. Specifically, we apply the same keyword search as their work but to our complete dataset, which includes posts that contain benign *artifacts* and also IoCs. Table II shows a high proportion of posts in our dataset (99.74%) contain malicious IoCs and do not contain the keywords to detect appraisals, and thus would be undetected by the approach from Li et al. [1]. Furthermore, the low proportion of appraisals in our dataset (0.03% of not-malicious and 0.3% of malicious posts) indicate that an ML approach for classification would be challenging, hindering the distinction between IoCs and benign *artifacts*. We conclude that their method, while providing an early warning system (Req. 2), does not meet Req. 3, and partially fails Req. 1.

III. CONTEXT-BASED IDENTIFICATION

To predict whether underground forum posts contain IoCs, we set out to answer the following Research Questions (RQs) designed to overcome the limitations of previous methods. In particular, we devise an RQ per requirement:

RQ1 Can we leverage the semantic meaning and metadata of a post to predict if an IoC is shared in that post? In other words, can we distinguish between posts with malicious IoCs and those with regular *artifacts*?
RQ2 Can underground forums offer a key advantage in the identification of novel malicious IoCs when compared to regular OSINT sources?
RQ3 Can our learning method apply to different types of IoCs? Even when a type of IoC is not present in the training set, can we still distinguish between posts sharing malicious and benign *artifacts*?

To answer these questions, we design a learning pipeline depicted in Figure 1. Our methodology has four main steps: collection of forum data (§III-A), identification of ground truth

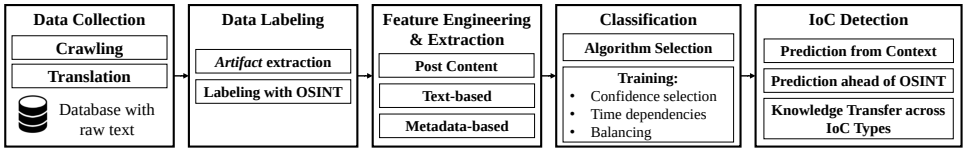


Figure 1. An overview of the methodology.

(§III-B) and features (§III-C), and the classification of IoCs (§III-D). We evaluate the methodology in §IV, and validate the requirement we have established in §V.

Overview: We extract *artifacts* from hacking forums, which may contain posts in different languages, and label them with OSINT. We build our ground truth at the post level considering different confidence thresholds. We extract posts' context and content information and use it to train our classifier with sentence embeddings as features. We investigate the optimal time window of historical data. We next describe in detail the methods that underpin our approach.

A. Collection of Underground Forum Data

Underground forums are known for their strict access restrictions, which hinders data crawling [11], [16]. To systematically collect data, crawlers require specialized techniques [31], [32]. In particular, the crawler first requires access to the forum, typically in the form of valid login credentials or access tokens. Stealthy data collection methods are crucial to prevent the account from being banned. Our method, influenced by CARONTE [32], mimics human navigation behavior, both in terms of connection frequency and data request rates. Whereas CARONTE concentrates on 'interesting' areas of the forum, we collect the whole content of the forum, presenting a more complex challenge. As we only have access to one account in our forum, we can not run multiple crawlers in parallel which affects the crawling speed. We obtained Ethics approval and discuss the ethical considerations in Appendix A. After collecting the data, we translate non-English text to English through a state-of-the-art multilingual neural machine translation tool [33].

B. Data Labeling

Forum posts contain non-textual elements like IPs, URLs, FQDN, etc. [1]. We use the term *artifacts* to refer to all these elements regardless of whether they have malicious, benign, or unknown attributes. Instead, we refer to the term "Indicator of Compromise" (IoC) when an artifact has a malicious attribute [13] (e.g., URLs of C&C servers, or malware Hashes). Prior work often associates IoCs with a type of *artifact* regardless of whether they are malicious or not [13], [15]. However, our work deals with data that may contain malicious *artifacts* (IoCs) as well as innocuous *artifacts* (e.g., sharing IPs of law enforcement, for hackers to add them to their blocklists). Hence, additional analysis is required to determine if an observed *artifact* is malicious.

As our end goal is to distinguish posts containing IoCs from those that do not, we label our dataset at the post level. We first extract the *artifacts* using [13], and then rely on OSINT (VirusTotal [14]) to determine whether those *artifacts* are malicious, benign, or unknown. This way, we collect ground truth from various AV tools, which we leverage to understand the balance between the number of engines (namely, votes) that identified an *artifact* as malicious and not malicious. A common practice is to provide a threshold on how many engines need to agree for an *artifact* to be considered an IoC [34], [35]. Next, we define our labeling strategy to characterize IoCs and posts.

Characterizing IoCs. To characterize our dataset we define a set $\mathrm{A}(\cdot) = \{a_1, a_2, ...a_n\}$ representing the list of n *artifacts* extracted from our dataset. We define a threshold τ that determines the minimum number of OSINT votes for us to consider an *artifact* malicious. We obtain the set $\mathrm{A}\tau(\cdot)$, containing the union of all the labeled IoCs that meet our threshold τ and all non-malicious *artifacts*. Concretely, given the number of malicious votes n for an *artifact* a, we get its corresponding label L_a as follows:

$$L_a = \begin{cases} malicious_\tau, & \text{if } n \geq \tau, \\ \textit{not-malicious}, & \text{if } n = 0, \\ excluded, & \text{if } (\tau\ != 1)\ \&\ (\tau > n \geq 1\ ). \end{cases} \quad (1)$$

Accordingly, we define an IoC as any *artifact* a with a malicious label ($L_a = malicious_\tau$), i.e., a malicious *artifact*. This means that for $\tau = 1$ we label all *artifacts* that at least have one engine classifying the element as malicious. For $\tau > 1$ we exclude *artifacts* with less malicious votes than τ. Thus, $\mathrm{A}\tau(\cdot) = \{a_1, ...a_i, ...a_j\}\ \forall\ L_{a_i} = malicious_\tau$ and $L_{a_j} = \textit{not-malicious}$. We abuse notation and refer to $\mathrm{A}\tau$ instead of $\mathrm{A}\tau(\cdot)$ to describe the set of *artifacts* with malicious$_\tau$ and *not-malicious* labels.

Characterizing Posts. We label posts by looking at the labels of the *artifacts* that they contain. Again, given a threshold τ of malicious votes, we obtain the set PMτ which includes all the posts that contain *at least* one IoC detected by τ engines or more. More formally, let $\mathrm{A}(p) = \{a_1, a_2, ...a_m\}$ be the list of m *artifacts* extracted from a post p (with $m \geq 1$), and $L(p) = \{L_{a_1}, L_{a_2}, ..., L_{a_m}\}$ their corresponding labels. We label a post p as follows:

$$L_p = \begin{cases} malicious, & \text{if } \exists\, a_i \in \mathrm{A(p)}\ |\ \ L_{a_i} = malicious_\tau \\ \textit{not-malicious}, & \text{if } \forall\, a_i \in \mathrm{A(p)}|\ \ L_{a_i} = \textit{not-malicious} \\ \text{“excluded”}, & \text{otherwise} \end{cases} \quad (2)$$

We exclude posts containing both *not-malicious* and *unknown artifacts*, to prevent noise in the datasets due to the uncertain status for which there is no OSINT.

C. Feature Engineering & Extraction

In this step, we extract metadata-based features (e.g., author's reputation, timestamps), text-based features (number of characters in a post), as well as the actual textual content of the posts. We process textual content as sentence embeddings, which we use later to identify if a post contains malicious

Table III
A SUMMARY OF FEATURES WE USE FOR CLASSIFICATION.

Feature Type	Feature Description
Post-content Features	1) Post content sentence embeddings 2) Thread headline sentence embeddings 3) Thread's first post sentence embeddings
Text-based Features	4) Total number of characters in a post 5) Number of *artifacts* in a post
Metadata-based Features	6) Author's community reputation 7) Author's forum rank based on their activity 8) Author's role 9) Author's total number of posts 10) Number of days between author registration and post time

IoCs. Simultaneously, this provides flexibility as the same method can be used to investigate other types of IoCs (e.g., malicious Bitcoin addresses or fake accounts).

1) Feature Engineering: A strong source of information to understand if a post contains an IoC is the post content itself. For example, the post *"Best Quality Call Service!. Calls to any phone in the world, both with and without number spoofing"* already indicates malicious activity from the text content. Therefore, our model leverages this type of information as the basis of the prediction. Unfortunately, forum discussions contain noisy unstructured text, including shortcuts, typos, as well as dark keywords [36]–[38] which are benign-looking words, but they might change their meaning (e.g., "rat" means "Remote Access Trojan") in the context of malicious activity. Sentence transformers are more resistant to this noise as words are considered in the context of the sentence, and are preferable to word-embeddings or word-count methods [39], [40].

To process posts, we mask *artifacts*, clean the text from special characters, and put it in lowercase. Then, we convert the text into sentence embeddings with our sentence-transformer model (i.e., "all-mpnet-base-v2") which is suitable for our forum dataset, as it was trained with more than 1 billion pairs of sentences from different online media, including Reddit or StackExchange [41]. Then, we reduce the high-dimensional sentence embedding vector to a lower-dimensional space. In our implementation, we use UMAP, an algorithm based on manifold learning to increase computational efficiency, and an embedding vector of 10 dimensions [42].

2) Feature Extraction: Consequently, for feature extraction, we consider *Post-content*, *Text-based*, and *Metadata-based features*, that are available directly at the time of the post creation. Table III details the type of features we use. The *post-content features* represent the content of the post, the thread headline, and the first post in the thread interpreted as sentence embeddings. By including the thread headline and the first post, we embed the context of the post together with all other replies that may be devoid of meaningful background (e.g., *"try this: {IoC}."*). The *Text-based features* and *Metadata-based features* offer further information to capture the structure of the post and the background of the author as used by prior work [19], [25].

For categorical features, we use a one-hot encoding. Specifically, *Author Rank*, and *Author Role* are mapped to an integer in an n-dimensional space, whereby *n* is the number of different classes for each feature. The remaining features are directly available as integer values. Finally, we conduct a feature importance analysis to identify the key features for predicting if a post contains an IoC.

D. Context-based Classification

We assess different classification algorithms to build the model that better addresses our problem. We try three different learning strategies to broaden the performance of our system. The first strategy plays with different confidence thresholds (τ) used for the labeling of *artifacts* as malicious. The second strategy considers time dependencies for training, investigating the optimal time window for an effective prediction. Third, we try different methods to balance the training data. We describe next these strategies in detail, which we evaluate in §IV, and finally describe the learning strategy we use for IoC Stalker to answer our RQs in §V.

1) Confidence Selection: As explained before, we set a threshold τ to determine the minimum number of malicious votes required for an *artifact* to be labeled as malicious. Intuitively, this threshold determines the *confidence* we have in OSINT sources: the lower the threshold, the higher the confidence in individual AV predictions. We thus introduce this threshold to ensure a reliable ground truth [43]. While it is commonplace to set this threshold to five when using OSINT to label malware [35], it remains unclear whether we can adopt more conservative thresholds for *artifacts*. This step of the methodology aims to select the value of τ that best adjusts to the correct identification of IoCs in underground forums. In our work, we compare $\tau = 1$ and $\tau = 5$. While the former offers a larger opportunity to train over stealthy IoCs, the latter is more conservative and has the potential to deliver a more reliable ground truth.

2) Time Dependencies for Training: When building training datasets, we take into consideration the timeline of when data appears in the forum to avoid temporal snooping [17], [30]. We therefore do not randomly generate test and training sets, but when predicting a post from a given year y, we only consider data from previous years ($< y$) for training. Considering these dependencies, we investigate different timeframes for training the classifier. In particular, we aim to determine the most suitable time window that hinders concept drift [44]–[46]. When predicting IoCs for year y, we seek to determine how many previous years $y - t$ should ideally be considered for training. We refer to t as *timeframe* in what follows. To answer this question we run the classifier on each sub-dataset and do this analysis for the last 10 years, meaning, we make predictions for IoCs discussed in year y and compare the classification results for different timelines and sub-datasets.

3) Balancing the Training Set: We follow best practices so that the classes within our training set are well balanced [47], and keep a constant ratio of malicious IoCs to benign *artifacts*. To respect time dependencies, we balance the training set as time progresses. When the ratio is imbalanced, we oversample data points misclassified in a previous timeframe $t - 1$ as OSINT becomes available for those samples in t. We use

the following oversampling strategies: 1) Random oversampling [48]; 2) Oversampling misclassified false negatives, meaning IoCs falsely classified as goodware; 3) Oversampling all misclassified *artifacts*; 4) Combining random oversampling and oversampling of misclassified malicious IoCs.

4) Learning Strategies: For learning how to detect malicious *artifacts*, we design the following training strategies: Step 1) `Everything`: we use all IoC types for testing and training, using our labeled dataset with a temporal training-testing split; Step 2) `<IoC_name>`: we train and test using a single IoC type, using the same split as before but filtering out all other IoCs (namely ablation study); Step 3) `<IoC_name>_excluded`: we leave one IoC type out from the training set (i.e., *<IoC_name>*), but include it in the testing; Step 4) `<IoC_name>_only` we train only with one type of IoC and test in all the others.

We use Step 1 to evaluate the general performance of IoC Stalker, Step 2 to assess the contribution of individual IoC types to the overall performance, and Steps 3 and 4 to evaluate the model under an open-world assumption to assess potential knowledge transfer across types. The performance we get from the former two steps may be seen as baselines to interpret the results in the latter steps. We note that the test set is permanent for all cases, as long as the IoC is part of the set of *artifacts* under evaluation. This means that the test set remains the exact same for Steps 1, 3, and 4. Naturally, the test set we use in Step 2 is tailored to the type of IoC being evaluated. We use the labeling strategy defined in §III-B for all cases. *Artifacts* are masked during the pre-processing stage (see §III-C1) but note that the implementation of our extraction process only captures popular IoC types and it may not be perfect. IoCs not captured are processed as part of the text.

IV. Evaluation

In this section, we implement and evaluate the different steps of the methodology. We later assess the performance of our system over our requirements in §V.

A. Collection of Forum Data

We collect data from an English-Russian hacking underground forum. The data collection spans over 18 years, thereby containing about ≈1.1M posts and ≈156k threads in 53 sub-forums with attentive daily activity. According to the forum website's statistics, there are ≈85k registered members, whereby our dataset registers ≈34k members who posted at least once over our data collection period.

B. Artifact Extraction and Characterization

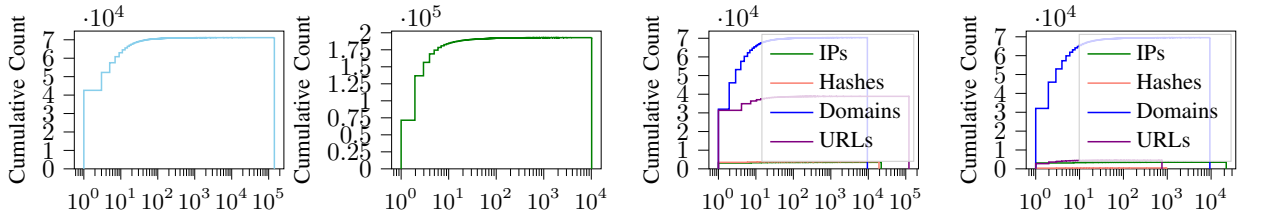
We use *iocsearcher* [13] to extract *artifacts* in our dataset. In this work, we consider four popular *artifacts* used in the realm of Threat Intelligence, i.e.: domain names (FQDN), IPs, URLs and Hashes. We find 885,417 *artifacts*, out of which we find FQDN (399,320), IPs (101,573), and URLs (327,325), as well as Hashes: MD5 (38,185), SHA1 (11,421) and SHA256 (7,593). Note that any other type of *artifact* is processed as part of the text. After extracting the *artifacts*, we query them on VirusTotal [14] between April and May 2023 and obtain reports for 503,489 *artifacts* (56.8%). A total of 381,928 *artifacts* (43.2%) are unknown and therefore excluded from our analysis.

Figure 2 shows the distribution of *artifacts* per thread and per post. Subfigures (a) and (b) show that ≈85% of threads and ≈95% of posts have fewer than 10 *artifacts*. Threads contain on average 12.4, and maximum of 142,591 *artifacts*, while posts contain on average 5, and max. 10,174 *artifacts*. In subfigure (c) we observe that IPs and Hashes are less spread out over the forum when compared to FQDN and URLs (the former two appear across 3.5k threads and 3.8k threads respectively and the latter two over 70.3k threads and 38.8k threads). Comparing subfigures (d) and (c), we observe that the prevalence of *artifacts* known to OSINT follows a similar distribution for the different types of *artifacts* (except for URLs, where there is significantly less OSINT intelligence available). This high prevalence of ground truth for the *artifacts* (as of May 2023) allows us to characterize our dataset. Specifically, we apply the labeling rules defined in our methodology using thresholds $\tau = 1$ and $\tau = 5$. As a result, we obtain two sets of *artifacts* A_1 and A_5 and two sets of posts P_1 and P_5, which we use to characterize the IoCs and posts.

1) Characterizing IoCs: We are interested in characterizing A_1 and A_5 to understand confidence threshold's impact on our dataset. Recall that A_τ is the union between *malicious*$_\tau$ and *not-malicious*. We find 439,586 *not-malicious* and 63,903 *malicious*$_1$ IoCs, among which are 3,781 *malicious*$_5$ IoCs. Figure 3 shows the breakdown of artifacts per type. Our dataset predominantly yields FQDN, IPs, and URLs, with a substantial presence of hashes (we study the timeline distribution of *artifacts* in Appendix B-A). We see a good balance between *malicious*$_1$ and *not-malicious*.

VirusTotal provides a "first submission date" date for Hashes and URLs and a "last analysis date" date for FQDN and IPs. We present in Figure 4 the time difference (in days) between the date when a Hash or URL was posted in the forum and the date when it was first submitted to VirusTotal for Hashes and URLs. We only represent positive values, i.e.: Hashes and URLs that were unknown to VirusTotal at the time of appearing in the forum. We see that it takes about 1.5 years for AVs to detect up to 50% of the IoCs, and over 5 years to detect 80% of the IoCs. This long period serves as a strong motivation for our work. However, due to this long submission period, we see limited availability of scan results for *artifacts* in the year 2023 and we exclude these posts from what follows.

2) Characterizing Posts: Recall that the set PMτ includes all posts that contain IoCs detected by $\geq \tau$. Looking at PM1 and PM5, we see 41,129 posts containing *malicious*$_1$ IoCs, whereby 2,327 out of those contain *malicious*$_5$ IoCs. We also see 77,449 containing *not-malicious* IoCs. Figure 5 displays the number of *malicious*$_5$, *malicious*$_1$, and *not-malicious* posts per year. We see an initial increase of *malicious*$_5$ and not *not-malicious* posts in the forums creation phase and a slight peak



a) Number of *artifacts*/Thread b) Number of *artifacts*/Post c) Number of *artifacts*/Thread d) Those known to OSINT

Figure 2. A Cumulative Distribution (CD) of how *artifacts* are distributed in threads and posts in the forum.

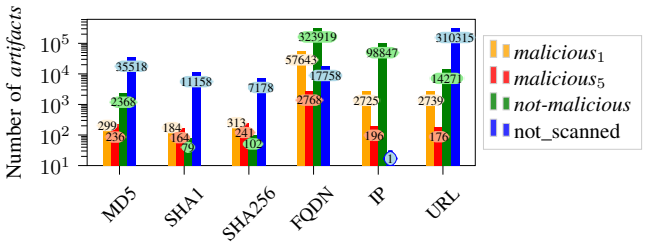


Figure 3. An overview of all *artifacts* extracted from our dataset, labeled as $malicious_1$, $malicious_5$, *not-malicious* or unknown ("not_scanned)."

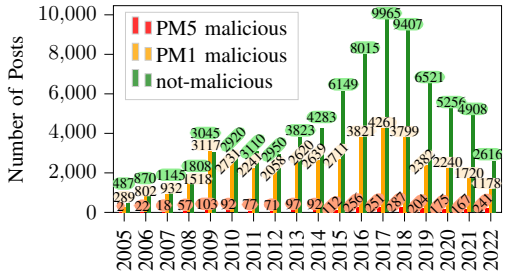


Figure 5. Visualizing the annual distribution of malicious posts of PM1, malicious posts of PM5, or *not-malicious* posts.

in the years 2016 - 2018. Note that the drop towards most recent years might be due to 'Unknown' *artifacts* that have not made it to VirusTotal just yet, due to the long submission time effect discussed earlier in §IV-B1. We exclude those posts that contain both *unknown* and *not-malicious artifacts* at the same time, to avoid potential noise that may stem from the *unknown*.

C. Classification

We next evaluate our classification choices (c.f., §III-D).

1) Algorithm Selection: For our analysis, we test different classifiers on PM1 to evaluate the most suitable one to address our problem, including: "Random Forest" (RF), "Logistic Regression" (LR), "Gradient Boosting" (GB), "MLPClassifier" (MC), "GausianNB" (GNB) and "LinearSVC" (SVC). We consider time dependencies (c.f., §III-D2) over 5 years, $y = [2022, 2021, 2020, 2019, 2018]$, comparing the mean F1-score. Details of the analysis are shown in Appendix B-B. Our selected feature set can reasonably predict if a post contains a $malicious_1$ IoC, achieving the best results with RF (F1-score of 0.739).

A key observation here is that some types of IoCs are harder to predict individually. Predicting Hashes individually provides slightly better results (F1-score 0.791). While GB shows the best results for URLs (F1-score 0.595), RF is comparably good (F1-score 0.592). The predictions for IPs and FQDN are significantly harder with LR only achieving an F1-score of 0.428 for IPs and an F1-score of 0.382 for FQDN. This could be due

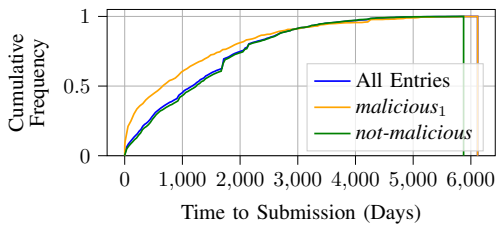


Figure 4. Time elapsed between the discussion date (post timestamp) and the first submission date (VirusTotal) of URLs and Hashes.

to different reasons. We find discussions mentioning Hashes usually provide a good explanatory context, which benefits the knowledge gained through embeddings. Furthermore, URLs and Hashes are not as volatile regarding their malicious status, as FQDN and IPs. We discuss this further in §VII.

2) Evaluation of Training Data Fine-Tuning: In this section, we apply three strategies described in §III-D to enhance the training set. Our goal is to measure the effect of each strategy on the training process and assess how well they enhance our model's predictions.

Confidence selection for training. We use the best-performing algorithm, Random Forest, to build two classifiers: C1 trained on set PM1, and C5 trained on set PM5. While the classifiers are trained on different sets, we use the same set PM1 for testing to offer a fair evaluation and comparable results, aligned with existing works, where a single malicious vote for an *artifact* makes it suspicious [49].

We run both C1 and C5 on years $y = [2013-2022]$ (in total, 88,314 posts), using the samples from $< y$ as training data to predict *artifacts* of year y. Figure 6 shows the classification performance focusing on the predictions for "malicious" as a positive class. We see good overall results for classifier C1 with a reasonable mean F1-score of 0.757, which represents a reliable prediction while keeping false positives and false negatives low. C5 does not have a high F1-score as the recall is very low. In other words, it only detects a few positives and plenty of false negatives. A reason for this could be the strong imbalance between $malicious_5$ and *not-malicious* posts, also visible in Figure 5. Instead, C5 has a high precision (>0.91) — if it classifies an artifact as $malicious_1$, it is very likely correct. However, C1 detects as much as 99% of the posts flagged as malicious by C5. An important difference between C1 and C5 lies in the false negatives, where C1 performs better. In particular, C1 correctly detects ≈82% of PM5, while C5 only detects ≈20% of these. As a result, C5 ends up with more false negatives. Since IoC Stalker aims to be an initial scanning

system whereby predictions undergo further investigation, we err on the side of having false positives over false negatives, and thus we use C1 for the remaining steps.

Balancing strategies. We compare the four balancing strategies discussed in §III-D3. We see that applying any of the four strategies improves the classification results, whereby the four strategies differ less than 1% in their values with a mean of ≈ 0.75, and a variance of $\approx 7.57 \cdot 10^{-7}$. Thus, we select classic random oversampling [48] due to its simplicity and since it is straightforward to apply without human validation for the misclassified items. We report the performance of the different balancing strategies in Appendix B-C.

Time Dependencies. We pick timeframes $t = [1, ..., 20]$ and predict years $y = [2013, ..., 2022]$ to determine the most suitable timeline for each sub-dataset. Timeframe 20 represents the full historical data available. For each timeframe, we average the F1-score of all years and present the results in Figure 7, offering more details about the highest values in Appendix B-D. Our results indicate, that while there is an ideal timeframe for each type of IoC, using fewer data is beneficial when results are comparable. We observe that seven years ($t = 7$) is the optimal timeframe, while any window >1 does not make a significant difference in the F1-score (difference less than 0.4%). It follows that when resources or historical data are limited, it is sufficient to focus the analysis on the recent 2 years.

We observe different optimal parameters for each type of IoC. This led us to investigate if the performance improves when we exclude certain types from the training set while keeping them in the testing set using optimal timeframes per type (cf. Appendix B-D). We report this in §V-C.

Overhead. To put things in perspective, we look at the average number of posts appearing per day, for which we have ground truth to build yearly models (88,314 posts). We compare the number of posts with those IoC Stalker flags as positive (TP + FP). Specifically, we see an average of 24.2 daily new posts with IoCs. By leveraging IoC Stalker, only ≈8 posts (7.8 on average) would require further investigation, whereby a subsequent analysis would confirm that ≈6 samples are actual malicious IoCs and 2 are false positives. As a result, the overhead placed over subsequent analysis systems is significantly reduced by approximately 68% (from an average of 24.2 to 7.8). This reduction allows for more focused and efficient investigations while maintaining a high level of coverage with a loss of ≈2 posts per day (FNs). Refer to §II to gain perspective of how this performance compares to existing methods.

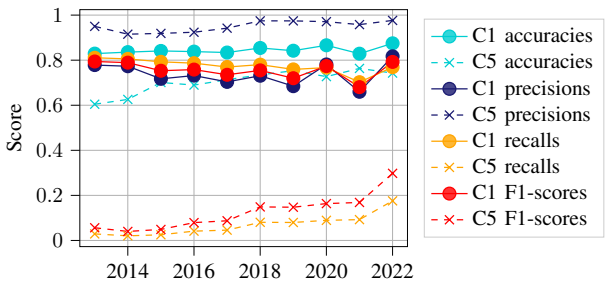


Figure 6. Classification performance of C1 and C5, for "malicious" class in different years (x-axis).

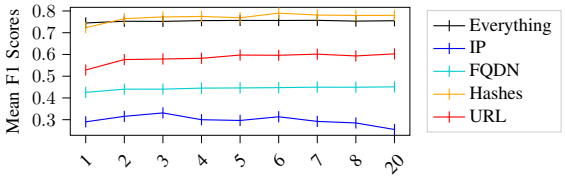


Figure 7. Mean F1-scores across datasets (legend) and timeframes (x-axis).

Takeaway. We conclude that there is a clear preference for a confidence threshold when generating the training data, and we show that applying time window sizes, as well as the balancing strategies is beneficial to enhance the training set, however, comparable results can be achieved with even a relatively small time window. This finding is valuable when historical data is limited. Similarly, balancing strategies offer minimal impacts on the results.

V. IoC Stalker

We build IoC Stalker using the best-performing setting derived from our evaluation (i.e., training over A1 with random oversampling and a timeframe of size 7). Next, we validate each of the RQs we present. To answer RQ1, we show that our algorithm can predict, from the context, if a post contains a malicious *artifact* (§V-A). To answer RQ2, we show how our model is able to identify IoCs before OSINT (§V-B). To answer RQ3, we demonstrate how context can be transferred across IoC types (§V-C). Finally, we perform an analysis on the different classifier predictions to better understand how the model performs (§V-D).

A. Prediction from context

We evaluate the yearly development of our classification, to analyze how it performs across years. While we use our best-performing parameters, results offer a similar progression to C1 in Figure 6 with slightly better performance. This improvement is mostly attributed to the optimal timeframe (i.e., $t = 7$). Figure 11 in Appendix C-A depicts the results of the classifier trained with our final settings.

Performance. We observe a good performance, increasing over time, except for a slight drop in 2021. The accuracy consistently remains higher than other metrics across all years. The F1-score stays relatively steady accross the years around its mean of 0.76, with a drop in 2021 to a minimum of 0.68 but reaching its maximum of 0.8 in the year after. The recall is generally higher than the precision, except for 2020 and 2022 — around the drop in the recall in 2021. This means that for those years the model was more conservative in predicting cases as malicious, thereby ensuring that those it does predict as malicious are highly likely to be true positives. The performance drop of the entire dataset in 2021 is slightly noticeable for FQDN and IPs. Instead, URLs show a performance increase compared to the previous two years. Looking at the FPs and FNs holistically, we have not observed

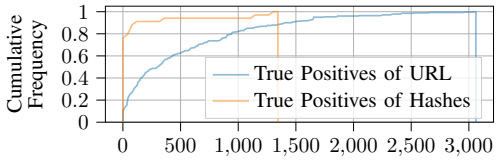


Figure 8. The time difference in days (x-axis) of an IoC versus VirusTotal "first submission date" ($\geq$ 1 day)

any particular reason for the performance drop. Judging by the significant increase in 2022, we attribute the dip to an organic concept drift [44], [50]. Finally, we note that our model predicts IoCs better when combined than when predicting them individually. We attribute this benefit to the transfer of knowledge as we discuss in §V-C.

Feature importance. We average yearly values and summarize the 10 dimensions of each sentence embedding vector for the post content, the initial post's content, and the thread headline to show the importance of each embedding. Table IV shows that the content and text-based features are significantly more important than Metadata-based features. The number of *artifacts* in a post is the most important feature, followed by the sentence embedding content of the post, the thread's first post, and the thread's headline. Apart from the author's role, which seems to have the least impact, the metadata is comparably important. These findings show the benefit of considering the post content (sum. of post-content embeddings, and text-based features: 93.2%) rather than features related to the author.

B. Prediction Ahead of OSINT

To answer RQ2, we focus on the true positives of our classifier. First, we compare our findings directly with our OSINT source (VirusTotal), and then with the most recent largest CTI release in the literature [51].

1) Comparison with VirusTotal: We perform a yearly analysis over $y = [2013 - 2022]$, comparing the time when we detect an IoC and the time when VirusTotal is first aware of it. Recall that VirusTotal only provides "first submission date" for Hashes and URLs, therefore we do not conduct comparisons for IPs and FQDN as these might not be accurate. We note, however, that our system can be deployed to crawl and detect posts in real-time, i.e., when IoCs are posted, and thus timely information for these IoCs could be obtained. Our system detects 21,159 malicious posts containing 37,317 malicious IoCs (383 Hashes, 33,086 FQDN, 1,894 IPs and 1,954 URLs).

Table IV
THE FEATURE IMPORTANCE AVERAGE OVER ALL YEARS. THE 10 EMBEDDING VALUES HAVE BEEN SUMMARIZED.

Feature	Importance
Post-content Features	$\sum$ **0.5794**
1) Post sentence embeddings	0.2178
2) Thread headline sentence embeddings	0.1642
3) Thread's 1st post sentence embeddings	0.1974
Text-based Features	$\sum$ **0.3527**
4) Total nr. of characters in the post	0.0511
5) Number of *artifacts* in a post	**0.3016**
Metadata-based Features	$\sum$ **0.068**
6) Author's community reputation	0.0142
7) Author's activity rank	0.0123
8) Author's role	0.0047
9) Author's number of total posts	0.0184
10) Days between author registration and post creation	0.0184

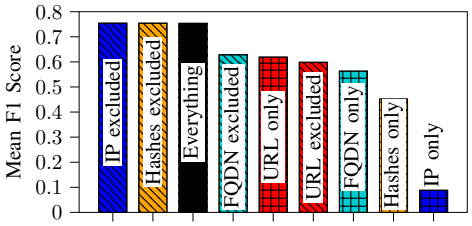


Figure 9. Ablation study over different types of IoCs. Showing the mean F1-score across all timeframes for each dataset.

As *artifacts* can be mentioned multiple times across posts, we remove duplicates and remain with 87 Md5, 54 SHA1, 175 SHA256, 6,135 FQDN, 1,013 IPs, and 1,161 URLs.

Figure 8 shows the time difference between Hashes and URLs detected by our system, compared to VirusTotal, where the time difference is at a day later. We see how our system detects some IoCs up to 3.7 and 8.4 years before VirusTotal for Hashes and URLs respectively. On average, we see how we detect 34 unique malicious Hashes 107 days in advance, and 281 unique URLs 536 days in advance. In summary, we detect IoCs $\approx$ 490 days before VirusTotal.

2) Comparison with CTI-Lense dataset [51]: We compare our approach with a recent dataset released by Jin et al. [51]. It provides data from seven publicly available OSINT sources (AlienVault OTX, Hail a TAXII, IBM X-Force Exchange, Cyware, PickupSTIX, Unit42, and Limo from Anomali) and three repositories (JamesBrine, DigitalSide, and Mitre Attack) containing a total of 6,363,065 objects within the time period of October 31, 2014, to April 10, 2023, which largely coincides with the time-span of our analysis. We extract the IoCs from their dataset and compare them with ours. Table V shows the overlap. We see 402 IoCs posted in the forum before they were included in the OSINT feeds of CTI-Lense. Also, our method discovers 8,123 IoCs that are not reported in this dataset, with a significant subset of Hashes and URLs ($\approx$ 21.3%) posted in the forum before they were known to VirusTotal.

C. Knowledge Transfer across IoC Types

We test how the classifier performance changes in the absence of certain types of IoCs in the training set. This analysis is motivated by the poor performance offered by some types of IoCs (§IV-C2) and to validate RQ3. For this analysis, we use the `Everything` dataset slice as a baseline, as well as learning strategies `<IoC_name>_only` and `<IoC_name>_excluded` as defined in §III-D4.

Figure 9 shows the performance in each strategy. Interestingly, we see that our learning pipeline transfers informative features across different types, enabling transfer learning over IoCs that initially were hard to predict. For instance, learning

Table V
COMPARISON BETWEEN OUR FINDINGS, THE DATASET PROVIDED BY [51] AND VIRUSTOTAL (VT). NUMBERS WITHOUT DUPLICATES. (*) MEAN NUMBER OF DAYS BEFORE OSINT.

Type	Hashes	URLs	IPs	FQDN	Total
IoCs detected before [51]	15 (631*)	14 (1.7k*)	138 (2K*)	235 (1.4k*)	402
IoCs not contained in [51]	291	1143	852	5,837	8,123
Not in [51] & detected before VT	32	274	-	-	-

in the absence of IPs in the training set yields performances as good as when considering every IoC type in the training set (0.75 F1). IPs are very volatile and the quality of their ground truth is detrimental. Furthermore, training with `URL_only` (0.62 F1) or `fqdn_only` (0.56 F1) is still beneficial to predict other types even in the absence of their ground truth. These findings offer valuable insights to devise cost-effective labeling strategies, as practitioners can completely exclude certain types from the pool of samples to label, like Hashes, that require complex and expensive static and dynamic analysis methods to determine if a sample is malicious.

D. Investigating Classifier Predictions

We investigate the reasons behind existing misclassifications. First, we check whether shorter posts provide less explanation, and would therefore be harder to classify. Then, to further understand the classification behavior, we perform a qualitative analysis of C1 and C5 predictions. We select a random sample of 200 posts per classifier (i.e., 50 TPs, 50 FPs, 50 TNs, and 50 FNs) and report our observations.

1) Evaluating Context: For each IoC-type and all IoC-types together (i.e., `Everything`) we calculate the min., max., avg., and median post-length for all classification groups TPs, FPs, TNs, and FNs. We consider the length that remains after preprocessing the post (c.f. §III-C1), and remove the *artifacts* that were masked to investigate only the textual content of the post. Overall, we find no strong evidence that misclassification is related to a shorter post length as we detail next. We start by analyzing the average length in Table VI for `IPs` and `Everything`. [2] Zero-length posts account for ≈2% of our dataset. We see that a median post length of 169 is enough to provide context to detect *artifacts* for `Everything`. Interestingly, we observe longer lengths in `IPs` when compared to `Everything`. As noted in Figure 9, including IPs resulted in the worst performance for knowledge transfer, while excluding them led to the best outcomes. We also observe that TPs are in general larger than FPs, suggesting that the detection of malicious posts could benefit from a longer context. However, this is not the case for benign posts, where TNs and FNs are shorter than FNs. Due to these nuances, we also perform a cursory manual inspection. We observe that the top longest posts of `IPs` have concise, yet valid contextualized explanations that come together with a dump of technical strings with no semantic value. Based on all this, we conclude that the length of a post does not generally justify existing misclassifications, and we identify that further insights into the reasons explaining existing misclassifications require dedicated qualitative analysis, as we perform next.

2) Evaluating C1: We look into the 200 samples to study how the context and content influence the performance of the model. When the context is descriptive and aligns with the *artifact's* status, the model performs well. If little context is available, distinctive keywords can help the classifier to make the right decision. In the absence of any content, the model can still make correct decisions, based on other features like first-post content or thread title. For example, we found a thread asking for proxies to send spam on social networks. Someone replied with a link to a malicious proxy shop, without any other content, yet IoC Stalker properly detected it as malicious. Confusing, ambiguous, or incomplete content poses challenges for the classifier, as well as when the described intention does not align with the *artifact's* status, i.e., an IoC discussed benignly (see Appendix C-B5). This reflects the complexity of subtle threat indicators. This confusion mirrors potential human error and highlights areas where human oversight could fail similarly, suggesting that certain classification tasks are inherently complex and not solely a machine limitation. Appendix C-B offers a deeper analysis of representative threads.

3) Evaluating C5: Similar findings from C1 are observed for C5. However, TPs explicitly talk about malicious activity in the presence of hacking-related keywords, detailed explanations and tutorials, explicit questions, and malware advertisements. We also observe similar discussions in the set of FPs, except the IoCs were not flagged as malicious by existing OSINT at the time of writing. In FNs, we find less explicit malicious discussions, with fewer unambiguous hacking-related keywords, but we observe a significant amount of content where the malicious intent is subtle to a human reader. This is most likely as C5 was trained with a more reliable threshold τ (i.e., using PM5) therefore misclassified gray behavior (i.e., 95% of PM1).

VI. Case Study

We investigate how our system performs in the early warning of IoCs through a case study. Table VII lists malicious products and services that would have been discovered by our method before VirusTotal, and compares them with the CTI-Lense dataset [51]. We select cases based on how early they were detected, and we next discuss three of them.

Spectre RAT. We find a post where a person shares a "Spectre RAT" project, a new Botnet that allegedly allows features such as accessing browser data or stealing bank information. The IoC is the URL of an online file-sharing platform. The file was created on 2022-03-31 (now expired), and was posted in the forum shortly after (on 2022-04-06). The first submission of the URL to VirusTotal was on 2022-07-15 (3.5 months after), and it is flagged as malicious. We note that the serving IP address is also classified as malicious. This example shows the potential of our system in early detecting malicious IoCs (Req. 2) — the submission to VirusTotal is more than 3 months after the discussion in a post where the actor shared the new malware.

Table VI
Investigating discussion context length

	`Everything`				`IPs`			
	TPs	TNs	FPs	FNs	TPs	TNs	FPs	FNs
min	0	0	0	0	1	0	0	0
max	218,470	76,856	47,731	41,469	43,497	83,108	47,731	35,331
mean	1,071	361	600	400	2,948	1,395	1,588	2,013
median	301	169	291	183	803	551	676	637

[2]Note that our post-processed dataset includes zero-length posts, which can be either correctly or wrongly classified based on the contents of the first post and headline of the thread, as we show in the Appendix C-B5.

Table VII
IOCS DETECTED BEFORE VIRUSTOTAL. ONLY THE IP IS INCLUDED IN [51] OVER 3.5 YEARS LATER (CF. §V-B2).

Name	Type	Undetected for	Description
Spectre RAT	URL	≈3 months	URL to download malware
Mars Stealer V8	URL	>2 weeks	URL to download malware
8x.14x.5x.18x	IP	-	IP related to phishing.
Predator Pain v12	Hash	>4.5 months	Stealer (keylogger)
Allow.exe	Hash	2 weeks	Malicious code injection
Install Exchange	Domain	≈1.5 months	Shop dedicated to PPI
Backdoor.exe	Hash	9 days	Unauthorized remote access

Install exchange. This case shows the advert of an install exchange shop, a criminal service (Pay Per Install, or PPI), popular in the underground economy [52]. We find FQDN and IP IoCs posted on 2022-04-23. The "first submission date" of the domain to VirusTotal is on 2022-06-02 (almost 1.5 months later). Although the domain is now unregistered, the serving IP is still flagged as malicious by VirusTotal at the time of writing, since it belongs to a hosting provider which is being abused for malicious activities.

Mars Stealer V8. We see a download URL in a post where an actor is offering a cracked version of "Mars Stealer V8", an information-stealing Trojan. The URL was posted in the forum on 2022-05-01. The first submission to VirusTotal was on 2022-05-16 and detected as "personal network storage" and "filesharing." The malware operated undetected for 15 days since the time of posting.

VII. DISCUSSION

Our work is designed to complement existing IoC detection solutions to improve CTI. We next discuss our limitations and then highlight our main takeaways.

Limitations. We rely on prior work to extract *artifacts* from text [13], and hence we inherit their limitations. Extracting IoCs from text is a long-established area of research [15] and we rely on an extensively-evaluated state-of-the-art tool [13].

Second, due to their dynamics, the OSINT label for some *artifacts* (e.g., IPs or URLs) might change over time [2], [43]. The historical information of our data should be contextualized with the corresponding historical ground truth from OSINT. VirusTotal, however, does not provide such historical info for IPs and FQDN (c.f., §IV-B1). This limitation might bias our classification (e.g., an IP that was malicious 5 years ago and nowadays is classified as benign), but it would not impact our system when deployed in real-time with continuous data collection.

Third, we observe a gap between when an IoC is published and when it is scanned by OSINT (c.f., §V-B). Since our system relies on OSINT to build ground truth, we might exclude recently discussed IoCs or miss those that are not on the radar of the OSINT community. In an ideal setting, our system would not necessarily have to fully rely on OSINT. Instead, our system could benefit from a deployment where a dedicated team of analysts was analyzing IoCs posted in the forums to build ground truth tailored to our domain. Unfortunately, we lack these resources but we argue that the use of OSINT is enough for our purpose, and our evaluation shows that learning from context can effectively help in getting ahead of the arms race against miscreants.

Fourth, the application of ML in the context of cybersecurity requires robust tools [17]. We consider an adversary who is not aware of our monitoring system, and thus might not attempt to attack the classifier (e.g., injecting or removing specific content to evade detection, or modifying the semantics to tamper with sentence embeddings). While this limits our ML pipeline in an adversarial setting, we believe that an adversary that takes due care to bypass such detection would also take care of not publishing IoCs that can be directly queried on VirusTotal or any other OSINT intelligence. Also, altering sentences in an adversarial fashion may not only confuse our method but human readers as well. This might not be desired if the author wants to reach as many users as possible, e.g.: when selling a product. Furthermore, large amounts of confusing and misleading content would decrease the quality of the forum content, and users may shift to other platforms. On the technical side, building sentence embeddings robust against adversarial attacks is currently an active area of research [53], studying its use in underground forums and how it may affect IoC Stalker is part of our future work. Despite this limitation, a system such as ours can also help in hindering communication and deterring the effective sharing of *artifacts* in underground forums in the presence of adversaries aware of our system. Furthermore, we note that forum administrators might act if they believe there are malicious actions in users' posts.

Finally, we limit our analysis to a single forum due to restrictions on VirusTotal's API. Our approach would not suffer this limitation should we have the budget for a private API, as CTI companies have. Despite this, we show how our method detects IoCs related to severe criminal activities like Trojans, keyloggers, or phishing-related IPs, 490 days before OSINT. We conclude that one single forum suffices to answer our RQs, generalizing across the data we observe. Also, we work with features that have been used in research on similar underground hacking communities [9], [54]. Furthermore, one important finding is that features derived from the discussions are more significant than author-related features (cf. §V-A). Thus, we posit that our method could work on other platforms hosting similar discussions.

Overall, our system can be viewed as a complementary tool for CTI, with early-warning capabilities. As such, and despite the benefits, our tool inherits limitations from the CTI realm as discussed above.

Takeaways This is the first work that fulfills all three requirements (§II) for IoC detection in underground forums:

Req. 1 (threat-less zero-shot learning). As demonstrated in §V-A, we are able to predict malicious *artificats* just from the context they appear in. In fact, we show in Table IV the importance textual features have in the detection, concluding that they can be strong indicator for the appearance of IoCs. We thus fill a gap from previous work, which either relies on characteristics from the *artifact* (e.g., URL patterns) [21], [22], [24], or do not use advanced NLP models such as sentence

transformers [1], [19], [20], [25]–[27].

Req. 2 (early detection). We can detect IoCs months/years before they appear in OSINT as shown in §V-B. Recall that our training process relies on time-invariant features. Thus, IoC Stalker helps in detecting malicious IoCs in the early stages of modern cyberattacks, i.e., when tools or services used in the attacks are traded in cybercriminal communities [6], [9], [16]. Our case study further supports this, showing examples of IoCs representing malware that could have been detected weeks or months in advance with our method.

Req. 3 (generalizable approach). We show in §III-D our system can improve the detection of IOC-types that are hard to predict, by relying on the combination of other IoCs. This means that the system is robust and it is able to learn from the noisy context of some IoCs. Furthermore, we show in §V-C that different types of IoCs can be predicted while not being part of the training set. This property allows to easily extend a system like ours to include other sources of CTI [1], [55]. Indeed, even in the absence of *artifacts* in discussion, e.g., when the actual trading or sharing of the IoCs could happen through DMs or alternative channels, our system would still be able to detect posts that advertise malicious products or services. We see value in flagging those services since this would assist CTI analysts to dig further.

Finally, we expect our method to offer an increased performance in a real-world monitoring environment, which could easily be integrated with existing OSINT feeds.

VIII. Related Work

Detecting malicious *artifacts* is a key aspect of CTI that has been prevalent in the cybersecurity literature [18]. This information is often given in OSINT repositories. Existing work studied and compared various public and commercial threat intelligence data sources [2], [3], showing that the ecosystem needs reliable and robust mechanisms to update these feeds. For such a purpose, in §II we listed related work that use the context and content, as well as features from the *artifacts* directly, to detect whether a given *artifact* is malicious or not [1], [19]–[29].

We review further literature that relies on features from the *artifacts* directly (mostly, for malicious URL detection). These provide the tools for IoCs investigation from online social media, to detect activities such as spamming [56]–[58], phishing [59], or other malicious activities [60]–[63]. Also, *artifacts* have been investigated to gain information about the context in which they appear, e.g., to profile users and potentially malicious behavior through evaluating the posted links [9], [64]. Finally, we see research focusing on the automatic extraction of IoCs from content obtained from unstructured, security-related text [13], [15].

Looking at the extraction of CTI beyond IoCs from hacking-related sources [6], [65]–[67], we see prior work extracting CTI from tutorials, source code, or attachments. Paladini et al. perform a measurement correlating threat reports and hacker forums [8]. Unlike our work, their approach does not predict actionable CTI in the form of IoCs. However, their results highlight the value of hacker forums as early threat indicators and the importance of proactively monitoring them for potential cyberattack detection, which motivates our work. Samtani et al. [68] extract and analyze them using a set of seeding keywords and snowball sampling to identify posts of interest. Other work classifies posts of interest through a set of keywords and bag-of-words [69]. While we have not considered these types of CTI *artifacts*, we note that our method would potentially identify such posts of interest, provided that we can gather ground truth for a subset of these. As shown in §II, closely-relevant related work fails to meet the requirements we set to address in this paper.

IX. Conclusion

In this paper, we propose a method to detect posts containing IoCs in underground hacking forums. This approach allows to identify IoCs at the initial stages of modern cyberattacks, when the tools or services are being shared, advertised or traded by cybercriminals. We conducted our study on a prominent English-Russian underground forum and considered four different types of IoCs: Hashes (MD5, SHA1, and SHA256), IPs, URLs, and FQDN.

For the first time, we design a method able to fulfill three key requirements for IoC detection in underground forums. First, we detect posts containing IoCs with a reasonably high F1-score of $\approx$0.8 through threat-less zero-short learning, meaning to train our classifier we only considered the context of the post. We find that Post-content and Text-based features are significantly more important than the metadata-based features from the author. Second, our method allows discovery of IoCs that stayed unknown to OSINT for an average of $\approx$ 490 days. In our case study, we showcased IoCs that our approach detected prior to their appearance in OSINT. Our detailed analysis of recent IoCs confirmed that they often surface in forum discussions shortly after their creation. Finally, since certain types of IoCs are harder to predict individually than others, we provided a method that benefits from the combination with other IoCs through transfer learning. Furthermore, we demonstrated how our classifier could predict different types of IoCs although those were not part of the training set. This means the approach can be easily extended to other types of IoCs beyond this analysis.

Acknowledgments

We thank the reviewers and anonymous shepherd for their feedback. This work was supported by project TED2021-132900A-I00 funded by MICIU/AEI/10.13039/501100011033/ and the European Union-NextGenerationEU/PRTR, and PID2022-143304OB-I00 funded by MICIU/AEI/10.13039/501100011033/ and by the ERDF, EU. G. Suarez-Tangil has been appointed as 2019 Ramon y Cajal fellow (RYC-2020-029401-I) funded by MICIU/AEI/10.13039/501100011033 and ESF Investing in your future. Special thanks to Andrei Costin for providing valuable feedback on this paper while visiting the University of Jyväskylä.

REFERENCES

[1] Z. Li and X. Liao, "Understanding and analyzing appraisal systems in the underground marketplaces." in *NDSS*, 2024.

[2] Á. Feal, P. Vallina, J. Gamba, S. Pastrana, A. Nappa, O. Hohlfeld, N. Vallina-Rodriguez, and J. Tapiador, "Blocklist babel: On the transparency and dynamics of open source blocklisting," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1334–1349, 2021.

[3] V. G. Li, M. Dunn, P. Pearce, D. McCoy, G. M. Voelker, and S. Savage, "Reading the tea leaves: A comparative analysis of threat intelligence," in *28th USENIX security symposium (USENIX Security 19)*, 2019, pp. 851–867.

[4] M. Motoyama, D. McCoy, K. Levchenko, S. Savage, and G. M. Voelker, "An analysis of underground forums," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, 2011, pp. 71–80.

[5] S. Pastrana, D. R. Thomas, A. Hutchings, and R. Clayton, "Crimebb: Enabling cybercrime research on underground forums at scale," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1845–1854.

[6] E. Nunes, A. Diab, A. 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 *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2016, pp. 7–12.

[7] L. Allodi, "Economic factors of vulnerability trade and exploitation," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 1483–1499.

[8] T. Paladini, L. Ferro, M. Polino, S. Zanero, and M. Carminati, "You might have known it earlier: Analyzing the role of underground forums in threat intelligence," in *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses*, 2024, pp. 368–383.

[9] S. Pastrana, A. Hutchings, A. Caines, and P. Buttery, "Characterizing eve: Analysing cybercrime actors in a large underground forum," in *Research in Attacks, Intrusions, and Defenses: 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings 21*. Springer, 2018, pp. 207–227.

[10] J. Hughes, S. Pastrana, A. Hutchings, S. Afroz, S. Samtani, W. Li, and E. Santana Marin, "The art of cybercrime community research," *ACM Computing Surveys*, vol. 56, no. 6, pp. 1–26, 2024.

[11] K. Thomas, D. Huang, D. Wang, E. Bursztein, C. Grier, T. J. Holt, C. Kruegel, D. McCoy, S. Savage, and G. Vigna, "Framing dependencies introduced by underground commoditization," in *Annual Workshop on the Economics of Information Security (WEIS)*, 2015.

[12] A. De La Cruz Alvarado and S. Pastrana, "Understanding crypter-as-a-service in a popular underground marketplace," in *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2024, pp. 85–90.

[13] J. Caballero, G. Gomez, S. Matic, G. Sánchez, S. Sebastián, and A. Villacañas, "The rise of goodfatr: A novel accuracy comparison methodology for indicator extraction tools," *Future Generation Computer Systems*, vol. 144, pp. 74–89, 2023.

[14] Virustotal, "Virustotal," https://www.virustotal.com/gui/home/upload, [Online] Last accessed: September, 21 2023.

[15] X. Liao, K. Yuan, X. Wang, Z. Li, L. Xing, and R. Beyah, "Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 755–766.

[16] R. Van Wegberg, S. Tajalizadehkhoob, K. Soska, U. Akyazi, C. H. Ganan, B. Klievink, N. Christin, and M. Van Eeten, "Plug and prey? measuring the commoditization of cybercrime via online anonymous markets," in *27th USENIX security symposium (USENIX security 18)*, 2018, pp. 1009–1026.

[17] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and don'ts of machine learning in computer security," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 3971–3988.

[18] N. Sun, M. Ding, J. Jiang, W. Xu, X. Mo, Y. Tai, and J. Zhang, "Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 3, pp. 1748–1774, 2023.

[19] J. Gharibshah, T. C. Li, A. Castro, K. Pelechrinis, E. E. Papalexakis, and M. Faloutsos, "Mining actionable information from security forums: the case of malicious ip addresses," *From Security to Community Detection in Social Networking Platforms*, pp. 193–211, 2019.

[20] P. Dewan and P. Kumaraguru, "Towards automatic real time identification of malicious posts on facebook," in *2015 13th Annual Conference on Privacy, Security and Trust (PST)*, 2015, pp. 85–92.

[21] C. Liu, L. Wang, B. Lang, and Y. Zhou, "Finding effective classifier for malicious url detection," in *Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences*, ser. ICMSS 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 240–244. [Online]. Available: https://doi.org/10.1145/3180374.3181352

[22] F. Alkhudair, M. Alassaf, R. Ullah Khan, and S. Alfarraj, "Detecting malicious url," in *2020 International Conference on Computing and Information Technology (ICCIT-1441)*, 2020, pp. 1–5.

[23] R. Islam, B. Treves, M. O. F. Rokon, and M. Faloutsos, "Linkman: hyperlink-driven misbehavior detection in online security forums," in *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ser. ASONAM '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 270–277. [Online]. Available: https://doi.org/10.1145/3487351.3488323

[24] J. Chen, Z. Hu, and Z. Qian, "Research on malicious url detection based on random forest," in *2022 14th International Conference on Computer Research and Development (ICCRD)*, 2022, pp. 30–36.

[25] J. Gharibshah, T. C. Li, M. S. Vanrell, A. Castro, K. Pelechrinis, E. E. Papalexakis, and M. Faloutsos, "Inferip: Extracting actionable information from security discussion forums," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ser. ASONAM '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 301–304. [Online]. Available: https://doi.org/10.1145/3110025.3110055

[26] J. Gharibshah, E. E. Papalexakis, and M. Faloutsos, "Ripex: Extracting malicious ip addresses from security forums using cross-forum learning," in *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part III 22*. Springer, 2018, pp. 517–529.

[27] J. Gharibshah and M. Faloutsos, "Extracting actionable information from security forums," in *Companion Proceedings of The 2019 World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 27–32. [Online]. Available: https://doi.org/10.1145/3308560.3314197

[28] A. Sapienza, A. Bessi, S. Damodaran, P. Shakarian, K. Lerman, and E. Ferrara, "Early warnings of cyber threats in online discussions," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017, pp. 667–674.

[29] H. Shin, W. Shim, S. Kim, S. Lee, Y. G. Kang, and Y. H. Hwang, "#twiti: Social listening for threat intelligence," ser. WWW '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 92–104. [Online]. Available: https://doi.org/10.1145/3442381.3449797

[30] K. Allix, T. F. Bissyandé, J. Klein, and Y. Le Traon, "Are your training datasets yet relevant? an investigation into the importance of timeline in machine learning-based malware detection," in *International Symposium on Engineering Secure Software and Systems*. Springer, 2015, pp. 51–67.

[31] K. Turk, S. Pastrana, and B. Collier, "A tight scrape: Methodological approaches to cybercrime research data collection in adversarial environments," in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2020, pp. 428–437.

[32] M. Campobasso, P. Burda, and L. Allodi, "Caronte: crawling adversarial resources over non-trusted, high-profile environments," in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2019, pp. 433–442.

[33] Google, "What is cloud translation?" https://cloud.google.com/translate/docs/overview, [Online] Last accessed: October, 14 2022.

[34] S. Pastrana and G. Suarez-Tangil, "A first look at the crypto-mining malware ecosystem: A decade of unrestricted wealth," in *Proceedings of the Internet Measurement Conference*, 2019, pp. 73–86.

[35] C. Smutz and A. Stavrou, "Malicious pdf detection using metadata and structural features," in *Proceedings of the 28th annual computer security applications conference*, 2012, pp. 239–248.

[36] K. Yuan, H. Lu, X. Liao, and X. Wang, "Reading thieves' cant: automatically identifying and understanding dark jargons from cybercrime marketplaces," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 1027–1041.

[37] D. Seyler, W. Liu, Y. Zhang, X. Wang, and C. Zhai, "Darkjargon. net: A platform for understanding underground conversation with latent meaning," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 2526–2530.

[38] Y. Jin, E. Jang, Y. Lee, S. Shin, and J.-W. Chung, "Shedding new light on the language of the dark web," 4 2022. [Online]. Available: https://arxiv.org/abs/2204.06885v2

[39] L. Zhou, A. Caines, I. Pete, and A. Hutchings, "Automated hate speech detection and span extraction in underground hacking and extremist forums," *Natural Language Engineering*, vol. 29, no. 5, pp. 1247–1274, 2023.

[40] V. Ghafouri, V. Agarwal, Y. Zhang, N. Sastry, J. Such, and G. Suarez-Tangil, "Ai in the gray: Exploring moderation policies in dialogic large language models vs. human answers in controversial topics," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, ser. CIKM '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 556–565. [Online]. Available: https://doi.org/10.1145/3583780.3614777

[41] H. Face, "all-mpnet-base-v2," https://huggingface.co/sentence-transformers/all-mpnet-base-v2, 07 2022.

[42] L. McInnes and J. Healy, "Umap: Uniform manifold approximation and projection for dimension reduction," 02 2018.

[43] S. Zhu, J. Shi, L. Yang, B. Qin, Z. Zhang, L. Song, and G. Wang, "Measuring and modeling the label dynamics of online {Anti-Malware} engines," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 2361–2378.

[44] L. Yang, W. Guo, Q. Hao, A. Ciptadi, A. Ahmadzadeh, X. Xing, and G. Wang, "{CADE}: Detecting and explaining concept drift samples for security applications," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2327–2344.

[45] A. Guerra-Manzanares, M. Luckner, and H. Bahsi, "Android malware concept drift using system calls: detection, characterization and challenges," *Expert Systems with Applications*, vol. 206, p. 117200, 2022.

[46] S. T. Jan, Q. Hao, T. Hu, J. Pu, S. Oswal, G. Wang, and B. Viswanath, "Throwing darts in the dark? detecting bots with limited data using neural data augmentation," in *2020 IEEE symposium on security and privacy (SP)*. IEEE, 2020, pp. 1190–1206.

[47] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Folleco, "An empirical study of the classification performance of learners on imbalanced and noisy software quality data," in *2007 IEEE International Conference on Information Reuse and Integration*, 2007, pp. 651–658.

[48] T. imbalanced-learn developers, "Randomoversampler," https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.RandomOverSampler.html, 05 2024.

[49] M. Lindorfer, M. Neugschwandtner, and C. Platzer, "Marvin: Efficient and comprehensive mobile app classification through static and dynamic analysis," in *2015 IEEE 39th annual computer software and applications conference*, vol. 2. IEEE, 2015, pp. 422–433.

[50] S. H. Bach and M. A. Maloof, "Paired learners for concept drift," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 23–32.

[51] B. Jin, E. Kim, H. Lee, E. Bertino, D. Kim, and H. Kim, "Sharing cyber threat intelligence: Does it really help?" 2024.

[52] J. Caballero, C. Grier, C. Kreibich, and V. Paxson, "Measuring Pay-per-Install: The commoditization of malware distribution," in *20th USENIX Security Symposium*. San Francisco, CA: USENIX Association, Aug. 2011. [Online]. Available: https://www.usenix.org/conference/usenix-security-11/measuring-pay-install-commoditization-malware-distribution

[53] J. R. Asl, P. Panzade, E. Blanco, D. Takabi, and Z. Cai, "Robustsentembed: Robust sentence embeddings using adversarial self-supervised contrastive learning," *preprint arXiv:2403.11082*, 2024.

[54] R. Overdorf, C. Troncoso, R. Greenstadt, and D. McCoy, "Under the underground: Predicting private interactions in underground forums," *arXiv preprint arXiv:1805.04494*, 2018.

[55] J. Cabrero-Holgueras and S. Pastrana, "A methodology for large-scale identification of related accounts in underground forums," *Computers & Security*, vol. 111, p. 102489, 2021.

[56] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@spam: the underground on 140 characters or less," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, ser. CCS '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 27–37. [Online]. Available: https://doi.org/10.1145/1866307.1866311

[57] A. H. Wang, "Don't follow me: Spam detection in twitter," in *2010 international conference on security and cryptography (SECRYPT)*. IEEE, 2010, pp. 1–10.

[58] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on twitter," in *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, vol. 6, no. 2010, 2010, p. 12.

[59] P. Yang, G. Zhao, and P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," *IEEE access*, vol. 7, pp. 15 196–15 209, 2019.

[60] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from urls," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417418306067

[61] M. Korkmaz, O. K. Sahingoz, and B. Diri, "Detection of phishing websites by using machine learning-based url analysis," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2020, pp. 1–7.

[62] C. Do Xuan, H. D. Nguyen, and V. N. Tisenko, "Malicious url detection based on machine learning," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, 2020.

[63] M. Bitaab, H. Cho, A. Oest, Z. Lyu, W. Wang, J. Abraham, R. Wang, T. Bao, Y. Shoshitaishvili, and A. Doupé, "Beyond phish: Toward detecting fraudulent e-commerce websites at scale," in *2023 IEEE Symposium on Security and Privacy (SP)*, 2023, pp. 2566–2583.

[64] B. Treves, M. R. Masud, and M. Faloutsos, "Urlytics: Profiling forum users from their posted urls," in *2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2022, pp. 510–513.

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

[66] N. Arnold, M. Ebrahimi, N. Zhang, B. Lazarine, M. Patton, H. Chen, and S. Samtani, "Dark-net ecosystem cyber-threat intelligence (cti) tool," in *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 2019, pp. 92–97.

[67] 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)*, 2017, pp. 3648–3656.

[68] S. Samtani, R. Chinn, and H. Chen, "Exploring hacker assets in underground forums," in *2015 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 2015, pp. 31–36.

[69] J. Gharibshah, E. E. Papalexakis, and M. Faloutsos, "Rest: A thread embedding approach for identifying and classifying user-specified information in security forums," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 14, 2020, pp. 217–228.

[70] D. Nevado-Catalán, S. Pastrana, N. Vallina-Rodriguez, and J. Tapiador, "An analysis of fake social media engagement services," *Computers & Security*, vol. 124, p. 103013, 2023.

APPENDIX A
ETHICS

The data we collect is available through a gatekeeper. We have not informed the gatekeeper of our collection process as this action would result in our account being banned and this study would not be possible. This has important Ethical implications, which we discuss next. First, there is a risk to the privacy of the users of the forum. We make no attempt at deanonymization in cases where the users use pseudonyms, and we do not focus on the real-life identities of the people involved in cases where the users use their legal names. Second, there is a risk of disrupting the normal operation of the site if our crawler is very aggressive. Our methodology

ensures that our crawling will not disrupt the normal operation of the site through a human-like crawl. Third, there is a risk that our study would attract the interest of someone falling into the pathways to cybercrime. We make sure to omit references to any unknown IoC studied in this work, and we do not publicly share the name of the forum. We have discussed these risks with the Institutional Review Board (IRB) of a former institution of a co-author where the data collection started. We obtained ethics approval to collect data and perform our analysis with application number LRS-19/20-17377.

APPENDIX B
EXTENDED EVALUATION

A. Characterizing IoCs

In §IV-B1 we study the annual distribution of *artifacts*. Our dataset predominantly yields domains, IPs, and URLs, with a substantial presence of hash values as well. When looking into the *artifacts*' distribution over the years, Figure 10 shows that while there has been an initial increase of mentioned IoCs in 2005-2009 due to the forums' creation phase, we see an even appearance of $malicious_1$ IoCs over the years with slight increases and decreases. We observe the same for *not-malicious*, and *not-scanned* artifacts, and additionally see a slight jump in 2013, where these two types tripled compared to the previous year. For $malicious_5$ IoCs we observe similar developments, only in 2008 there is a higher increase of $malicious_5$ compared to the other types in the same period.

B. Algorithm Selection

In §IV-C1, we evaluate the best classifier for our prediction including Random Forest (RF), Logistic Regression (LR), Gradient Boosting (GB), MLPClassifier (MC), GaussianNB (GNB), and LinearSVC (SVC) on PM1 to identify the most effective method for detecting suspicious Indicators of Compromise (IoCs). The analysis considers time dependencies (cf. §III-D2) and is conducted over five years $y = [2022, 2021, 2020, 2019, 2018]$. Table VIII summarizes the results, showing the mean F1-score over the 5 years.

When predicting posts containing $malicious_1$ IoCs, RF performs the best overall, achieving an F1-score of 0.739, whereas GB was most effective for URLs with an F1-score of 0.595.

We note that some *artifact* types, like Hashes, are easier to predict individually (F1-score of 0.791 with RF), while predicting IPs and Domains is more challenging, with significantly lower F1-scores of 0.428 and 0.382, respectively. This may stem from the volatile nature of domains and IPs compared to the more stable nature of URLs and Hashes.

C. Balancing Dataset

In Table IX we show the results of the different balancing strategies we evaluated in §IV-C2. We show that any of them improves the classification outcome with minimal variance among them (less than 1% difference with a mean of ≈ 0.75, and a variance of $\approx 7.57 \cdot 10^{-7}$). Due to its simplicity of application without the need for human validation of the misclassified items, we pick classic random oversampling as the preferred method.

Table VIII
MEAN F1-SCORES OF THE TESTED CLASSIFIERS.

Mean of years 2018-2022					
Classifier	Everything	IP	FQDN	Hashes	URL
GNB	0.228	0.378	0.197	0.643	0.371
MC	0.629	0.124	0.251	0.535	0.485
SVC	0.344	0.048	0.166	0.694	0.476
GB	0.722	0.180	0.303	0.750	**0.595**
LR	0.636	**0.428**	**0.382**	0.579	0.571
RF	**0.739**	0.160	0.356	**0.791**	0.592

Table IX
COMPARISON OF DIFFERENT BALANCING STRATEGIES.

Method	Value
Dataset without balancing	0.7513
1) Random Oversampling	0.7551
2) Re-adding Malicious	0.7568
3) Re-adding All	0.7546
4) Oversampling & Re-adding Malicious	0.7548

D. Time dependencies

The highest values for each with the corresponding timeframe size t are reported in Table X. We see that seven years ($t = 7$) is the optimal timeframe for the entire dataset. When resources are limited, it is sufficient to focus the analysis on the recent 2 years. For Hashes, ($t = 6$) years of historical data bring the best results, while when considering the most recent 2 years and therefore 1/3 of the data the performance loss will be $\approx 2.5\%$, or when using the most recent 3 years (1/2 of the data) the loss will be $\approx 1.7\%$. The entire historic data ($t = 20$) is the ideal training data size for URLs and FQDN, while the performance loss when focusing on the 2 most recent years is $\approx 2.6\%$ for URLs, and $\approx 1\%$ for FQDN. The ideal timeframe when checking IPs is ($t = 3$) years. The performance loss of $\approx 1.6\%$ for $t = 2$ and $\approx 4.1\%$ for $t = 1$ indicates that using just slightly fewer data results in a relatively large loss in performance.

Table X
MEAN VALUE (v) OF ACCURACY (A), PRECISION (P), RECALL(R), AND F1-SCORE (F1) ACROSS THE YEARS. DISPLAYING THE MAXIMUM VALUE (v) AND THE CORRESPONDING TIMEFRAME (T).

	everything		ip		fqdn		Hashes		url	
	t	v	t	v	t	v	t	v	t	v
a	4	0.85	6	0.80	2	0.83	6	0.70	7	0.80
p	4	0.75	6	0.49	2	0.68	20	0.75	7	0.76
r	20	0.77	3	0.27	20	0.35	6	0.87	20	0.51
f1	7	0.76	3	0.33	20	0.45	6	0.79	20	0.60

APPENDIX C
IOC DETECTION

A. Prediction from context

Figure 11 shows the performance of our prediction in §V-A. The overall good performance of the dataset is improving over time, with a noticeable decrease only in 2021. The Accuracy is higher than other metrics each year. The F1-score fluctuates around a mean of 0.76, with a dip to 0.68 in 2021 but recovering to 0.8 the following year. Apart

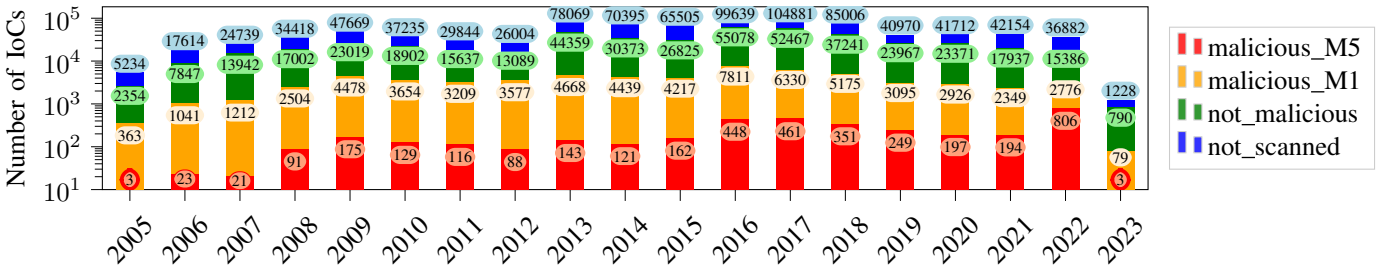


Figure 10. Evolution of total IoCs posted per year.

from the years 2020 and 2022, the recall is higher than the precision, indicating the model was more conservative when making predictions. Despite examining false positives (FPs) and false negatives (FNs), no specific cause for the 2021 dip was identified; however, the recovery in 2022 suggests the influence of organic concept drift [44], [50].

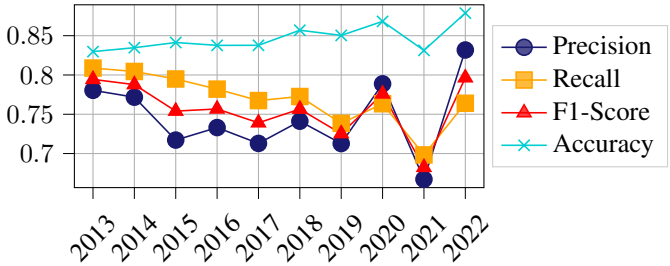


Figure 11. Classification performance scores (y-axis) across the years (x-axis) for malicious as positive class, with PM1 as training set, using random oversampling and a timeframe of size 7.

B. *In-depth analysis of Example Threads from C1 Predictions*

We provide illustrative examples of some of the cases that have led to the findings described in §V-D2.

1) **Example 1 - Windows privilege escalation:** We investigate posts of a thread about Windows Privilege Escalation. When good context is provided, the model correctly detects malicious posts and realizes that the absence of any context means *not-malicious*. The model got confused when the post content was describing malicious intention, but the artifacts were benign. Further misclassification occurred when a post only contained a vulnerability name but no further explanation. The context of the thread headline, and the first post talking about malicious interaction, led the model to decide on malicious activity, although the artifacts were benign. A human investigator would have made the same mistakes. There were no FNs in this thread. Despite the FP posts being wrongly classified, there were replies in the context of a thread where the headline and first post are related to actual malicious behavior, which provides valuable insights for detecting and pinpointing malicious activity.

True Positives. Most posts in the thread are TPs. The first post gives instructions and shares tools to conduct privilege escalation in Windows (a tutorial). The post contains URLs flagged as malicious from which the tools can be downloaded. A human reader would have flagged this post as malicious as well. We find that some replies contribute to the tutorial by sharing other malicious URLs. For example, one post writes "Windows readfile 0day" containing a file-sharing URL. The URL itself was not known to VirusTotal at the time of our scan, but the domain was. Although the text description is short, the post is properly marked as malicious by IoC Stalker. Due to the keyword "0day", a human analyst would have decided to classify the post as malicious.

False Positives. Some posts were falsely classified as malicious. One post shares a link to a privilege-escalation cheat sheet. While the post content and the intent are malicious, the *artifact* itself is not. Another post in this thread contains the name of a vulnerability in Microsoft Defender , and two links with further information about the vulnerability (Github and Microsoft). Missing explanation in the post leads to wrong classification. The post could have been flagged correctly as *not-malicious* by investigating the benign domains. Furthermore, one post was misclassified because it shares a project link with a brief description about increasing privileges through the Windows Kernel Cryptography Driver and the vulnerability name. A human investigator would have also further investigated the *artifact*.

True Negatives. One post containing only an image link, with no text was correctly classified as *not-malicious*. Despite the thread's headline and first post, the model correctly realized that the absence of information means the IoC is *not-malicious*.

2) **Example 2 - Thread containing dark keywords:** In this thread, a person is advertising a newly developed RAT (Remote Access Trojan). Although posts included dark keywords, the classifier correctly classified TPs, because they contained sufficient explanation and context. A FP post consisted of short incomplete sentences, containing keywords and forum slang which led to translation errors. This made it very difficult to understand and classify for both a machine and a human. There were neither TNs nor FNs in this thread.

True Positives. The initial post advertises the new RAT and its features, then finally shares the link to download the malware. The post does not contain the word "trojan" yet provides sufficient context to understand the malicious intention. A follow-up post containing an updated version and the download link is also correctly classified. A human investigator would have decided the same.

False Positives. One post is falsely classified as malicious. The author is answering the previous discussion, expressing gratitude for the program. The remaining content is challenging to understand as it is a brief description containing dark

keywords and translation errors. A human investigator would have difficulties understanding it. This probably led to a wrong decision for the classifier as well.

3) **Example 3 - Which RAT to use:** We investigate a thread whose creator asks for advice on which RAT to use. There is no *artifact* contained in the initial post. The classifier was correctly able to distinguish the types of answers looking for advice that does not include IoCs (TN), and offering malware that contains IoCs (TP). It had, however, difficulties identifying the IoC among the benign-looking expressions of opinion about RATs. There were no FPs in this thread.

True Positives. One person is sharing information by providing the RAT's name together with a filesharing URL. Although the post provides little context, the model correctly classified the post as malicious. A human would have decided the same.

True Negatives. In a brief reply, someone expressed the need for two specific types of RATs. The model correctly classified the contained artifacts as *not-malicious*, a human would have done the same.

False Negatives. In another reply, a person shares their opinion on a RAT, whereby the malicious URL is the RAT's official website. The context does not express malicious intent explicitly, as it discusses the latest news regarding the RAT. However, a human would have understood from the context that the shared link probably contains malicious content.

4) **Example 4 - Initial post with short context:** We investigate two cases where the initial posts in their respective threads lead to confusion due to missing context.

False Negative. A thread was opened with the intention of "Selling a crypto database" (as mentioned in the title). While the first post contained a malicious domain, it was written with so little information that the model was not able to detect it as malicious.

False Positive. In a first post, a user is referencing a TV show (*"Who is watching Mr. Robot?"*), posing a question about the method used to search for a Trojan in one of the episodes (including some technical details of its functioning). The *artifact* is the link to a streaming service to watch the series, but it was wrongly classified as malicious due to the context.

5) **Example 5 - Posts missing context:** Recall from Section V-D2 that zero-length posts can be correctly classified. Here, we give two examples of replies that post *artifacts* without context and how this influences the classifier's decision in the existence of the headline and first post.

False Negative. In a thread, a person is discussing how to boost likes on social media, which is a common malicious activity traded in hacking forums [70]. However, it did not provide details on how these were obtained, thus lacking, at the eyes of the classifier, a malicious purpose. As such, one of the replies is misclassified as FN, since it only contains a malicious link with no context.

False Positive. A thread starts with a post that describes vulnerabilities for a web content management system. In the same thread, one reply shares a benign link to an OSINT repository, without further description. Thus, this *artifact* has been misclassified due to the contents of the first post and thread headline.