Revealing Unreported OT Vulnerabilities from Public Discussions

Arslane Fawzi Halilou (✉) and Natalia Stakhanova (✉)

Department of Computer Science, University of Saskatchewan, Canada
kjc705@usask.ca, natalia@cs.usask.ca

Abstract. The convergence of Information Technology (IT) and Operational Technology (OT) has transformed industrial systems into interconnected, data-driven environments. While this shift enhanced real-time monitoring, automation, and decision-making, it has also expanded the attack surface of Industrial Control Systems (ICS), exposing them to significant cybersecurity risks. Although official databases such as MITRE's CVE and the NVD encompass large numbers of known vulnerabilities, evidence shows that vulnerabilities are often discussed in public sources (e.g., social media, blogs, forums) before their official disclosure. This lag poses a critical risk for OT systems, where applying patches is challenging due to legacy hardware and operational constraints. To address this, we propose a predictive framework that identifies undisclosed OT-related vulnerabilities by monitoring online sources such as mailing lists, news websites, and security podcasts. Our proposed framework filters content using device specifications and protocol information to isolate OT-relevant discussions. It then uses linguistic patterns from known vulnerabilities to detect vulnerability indicators. Experimental results demonstrate high accuracy for the proposed framework ranging from 86% to 99% in detecting signals of OT vulnerabilities in public discussions across multiple sources.

1 Introduction

No software is free of vulnerabilities. To identify these security flaws, traditional approaches rely on expert analysis and systematic examination of the software. Once a vulnerability is discovered, the first step is to confirm that it is a genuine security flaw and assess its impact to evaluate the severity of the vulnerability and its potential consequences. If the vulnerability is confirmed, it is documented and reported. Ideally, after responsible disclosure process, all software vulnerabilities should be officially documented and registered in authoritative databases such as the National Vulnerability Database (NVD) and the Common Vulnerabilities and Exposures (CVE) database. Unfortunately, the reality is different. Many vulnerabilities remain undisclosed, underreported, or are shared informally through public forums, news outlets, or technical discussions before they appear in official records. As a result, monitoring online public and unofficial sources to uncover vulnerabilities not present in official databases has become increasingly important.

The existing studies have focused on social media platforms, GitHub, cybersecurity blogs, and dark web forums to address the limitations of official vulnerability databases. The majority of these studies demonstrate that such public sources can serve as effective early warning systems for rapidly spreading cyberattacks [25,13,3,26,18,22,27]. Typically, these approaches rely on a precompiled dictionary of terms describing known attacks (e.g., malware, trojan) and emphasize the use of social media platforms to provide early alerts, assess the scale of attacks, and identify affected geolocations. A few efforts have also attempted to detect signs of undisclosed vulnerabilities using a predefined ontology indicative of vulnerabilities [21]. However, the robustness of these methods for detecting vulnerabilities not officially reported remains limited.

In this work, *we explore the potential of public sources to reveal undisclosed or underreported vulnerabilities.* This is particularly relevant in the operational technology (OT) domain, where the critical nature of systems, combined with challenges such as legacy hardware and strict operational constraints, makes patching particularly difficult.

We propose a framework that proactively monitors online sources to identify OT-related vulnerabilities, leveraging features derived from OT device specifications and linguistic patterns extracted using topic modeling techniques, i.e., BERTopic and Latent Dirichlet Allocation (LDA), commonly found in known vulnerability descriptions from official databases. We evaluate the robustness of the proposed framework across several online sources, including the Full Disclosure mailing list, the Ars Technica news site, and transcripts of the Security Now podcast, using a dataset of over 125,000 messages, articles, and podcast episodes.

Experimental results demonstrate high accuracy for the proposed ensemble-based framework, ranging from 86% to 99% across all three datasets. Precision and recall were particularly strong on the Full Disclosure dataset, reaching 87% and 83% respectively, highlighting the framework's effectiveness in detecting vulnerable OT content. Performance on the SecurityNow and Ars Technica datasets remained solid but showed a noticeable drop in precision, recorded at 56% and 67% respectively. A decline in recall was also observed for Ars Technica (46%) compared to SecurityNow (80%). While the lower performance on SecurityNow is expected due to its informal language, unstructured format, and frequent topic shifts, the relatively weaker results on Ars Technica are more surprising given its structured text. Further analysis of false positives revealed that Ars Technica articles often reference vulnerabilities indirectly or as part of broader discussions, which complicates the classification task.

These findings suggest that, regardless of the content source or the level of textual structure, similar language is consistently used to describe vulnerabilities. This linguistic consistency enables the model to effectively identify indicators of officially undisclosed vulnerabilities across diverse sources.

The rest of this paper is organized as follows. Section 3 gives an overview of related works and our contribution. In section 4, we explain our proposed

approach. Section 5 describes the experimental results obtained for the proposed model. Finally, this work is concluded in section 6.

2 Background

Traditionally, the vulnerability discovery process aims to identify security flaws in software systems. Once a vulnerability is discovered, the first step is verification, where the issue is tested to confirm that it is a genuine security flaw rather than a false positive. If the vulnerability is confirmed, an impact assessment is conducted to evaluate the severity of the vulnerability and its potential consequences. After this assessment, the vulnerability is documented and reported either internally, to the software vendor, or to public vulnerability databases, depending on the disclosure policy. Developers then work to create and test a patch that resolves the issue without introducing new problems. Once the fix is validated, it is deployed across affected systems. Finally, if appropriate, the vulnerability and its resolution are publicly disclosed, with an official CVE identifier and a CVSS (Common Vulnerability Scoring System) score.

Disclosure policy A vulnerability can be disclosed following a responsible disclosure policy, full disclosure, or private disclosure, depending on the context, the discoverer's intent, and the affected organization's practices. If *responsible disclosure* (also known as coordinated disclosure) policy is followed, the researcher privately notifies the affected vendor and allows time for a fix before making the details public. In contrast, *full disclosure* involves publicly revealing the vulnerability shortly after discovery, regardless of whether a patch is available. *Private disclosure* occurs when the information is shared only with the vendor and never made public, while *non-disclosure* refers to situations where the vulnerability is not reported at all, sometimes due to legal concerns, lack of incentives, or malicious intent.

3 Related work

Software vulnerabilities have been at the center of research for decades. The vulnerability discovery process is actively studied. In this paper, we focus on related work related to existing vulnerabilities that were discovered through traditional means but were never officially disclosed.

Vulnerability as early warning system. Public online sources have been widely examined to predict rapidly spreading cyberattacks [25,13,3,26,18,22,27]. Typically, these studies rely on official vulnerability repositories such as CVE or a precompiled dictionary of terms describing known attacks.

Many of these efforts leverage messages on the Twitter platform. For instance, Le et al. [17] proposed a model to collect tweets related to existing vulnerabilities. Quentin et al. [18] introduced an automated framework for the real-time detection and geolocation of ongoing security incidents, based on a predefined taxonomy of cybersecurity-related terms. Sabottke et al. [24] explored the use

of Twitter platform for early detection of exploits against known vulnerabilities contained in NVD database.

A similar approach was taken by Mittal et al. [21], who introduced CyberTwitter, a real-time monitoring system that analyzes the Twitter stream to extract information about emerging threats and vulnerabilities. Their system relies on a security ontology and is tailored to the organization's custom system profile. Another Twitter-based monitoring system, SYNAPSE, was proposed by Alves et al. [3]. Unlike broader approaches, SYNAPSE collects tweets exclusively from selected security-related accounts and filters the content to extract intelligence relevant to the assets of the monitored infrastructure. The potential of the Twitter platform for zero-day vulnerability detection was explored by Sauerwein et al. [26]. The study mapped tweets to different phases of the vulnerability lifecycle showing that vulnerabilities are discussed on Twitter before their official public disclosure. Alevizopoulou et al. analyzed Twitter platform for collection of IoT device relevant security tweets [1]. While Twitter proves valuable for early cybersecurity insights, its character limit presents a challenge for conveying detailed technical information [7].

Beyond Twitter, existing approaches such as DISCOVER [25] leveraged multiple sources including Twitter, blogs, and dark web forums to monitor terms potentially indicative of cyber threats (attacks) for early warning generation.

Several studies explored the possibility of assessing the likelihood of known software vulnerability exploitation based on data in public sources such as rating of vulnerabilities by Common Vulnerability Scoring System (CVSS) [6], dark web [2], CVE and Twitter [14].

Numerous studies also examined methods for predicting the severity of already discovered vulnerabilities as an alternative to manual severity assessments [10,19,31,28,30,11]. More recent work explores the use of large language models (LLMs) for CVSS classification [20,23].

Unlike these studies that rely on officially known vulnerabilities to determine their exploitability or access the scope of the exploitation, we propose a framework that proactively scans public sources to identify undisclosed and therefore not officially registered OT vulnerabilities.

Inconsistencies in Vulnerability Databases. The vast majority of vulnerability-related studies rely on official vulnerability repositories. However, numerous inconsistencies have been identified in these data sources. These include discrepancies in severity scores and vulnerability types [4], mismatches in software names and versions between the standardized NVD database and the unstructured CVE descriptions [9], differences in CVSS base metrics assigned by different organizations [15], inconsistencies between identical or semantically similar NVD entries [32], and inconsistencies in NVD's CPE tags [29].

4 Proposed approach

The ultimate goal of our approach is to predict whether a public discussion includes an indication of a security vulnerability. Accurate identification of public

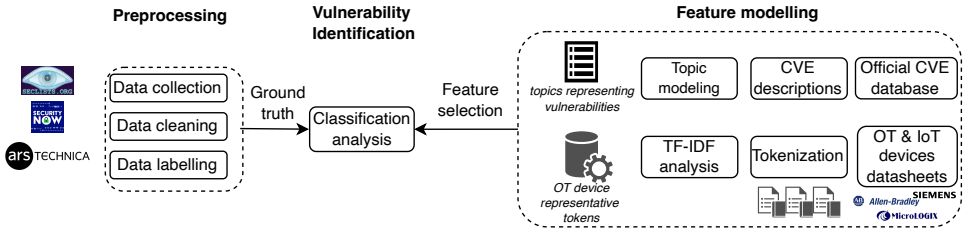

Fig. 1: Flow of the proposed framework

discussions that are likely to lead to official vulnerabilities is critical for a timely system response. To facilitate early identification of vulnerabilities specific to OT devices, we propose a framework that (1) identifies OT device-related content and (2) determines whether it describes a security vulnerability. Figure 1 illustrates the general flow of the proposed framework that consists of three main stages: data preprocessing, feature modelling, and vulnerability identification.

4.1 Preprocessing

Data collection. Collecting data from various sources is essential for effectively identifying potential vulnerabilities in OT devices. Public discussions related to cybersecurity can occur across a wide range of platforms, including social media, technical forums, security podcasts, research publications, vulnerability databases, and news sites. Each source offers unique perspectives, levels of technical detail, and timeliness. By aggregating information from diverse channels, the framework can capture a broader and more representative view of emerging threats, reduce blind spots, and improve the accuracy of both device-related content identification and vulnerability recognition.

For our analysis, we leveraged three sources of information: FullDisclosure mail list (from 2002 to 2024), Ars Technica news site (from 2006 to 2025), and SecurityNow podcast (from 2005 to 2025). To collect information across multiple sources, we designed custom crawlers to collect all available messages from mail list, all news articles (including those not related to security), and all public episodes of the podcast. The period wherein the data was collected differed according to the source: 27 dec 2024 to 6 Jan 2025 for FullDisclosure, 22 April 2025 to 25 April 2025 for Ars Technica, and 25 April 2025 for SecurityNow.

Data cleaning. Once the data was collected, it was preprocessed to remove duplicate entries, unparseable characters, and HTML tags. However, punctuation was intentionally preserved, as removing it could alter critical versioning information such as product versions, firmware identifiers, and hardware model numbers (e.g., CPU types) that are commonly associated with vulnerabilities.

For example, removing punctuation from the following article excerpt *"...the attacks targeted iPhones running iOS versions 15.7 through 16.0.3..."* would

fragment the software version numbers into arbitrary digits that could be misinterpreted as unrelated data, e.g., dates.

Data labelling. To establish ground truth, we labelled the extracted datasets. Labelling of OT-related content was done based on a precompiled dictionary of keywords known to indicate industrial OT devices, such as programmable logic controllers (PLCs), remote terminal units (RTUs), and supervisory control and data acquisition (SCADA) systems. The labelled data was manually verified afterwards to ensure its relevance.

Vulnerable content, on the other hand, was labeled based on the presence of an official CVE identifier in the text. A CVE identifier is a standardized string referring to a unique, officially disclosed vulnerability, formatted as CVE-Year-UniqueID, where the unique ID is typically four digits or more. The presence of such a string in messages, emails, or articles strongly suggests that the content refers to a vulnerability. Based on this criterion, any extracted content containing a CVE identifier was labeled as vulnerable; otherwise, it was labelled as not vulnerable. As with OT-related content, we manually reviewed the labeled data to avoid misclassifications particularly in Ars Technica articles, which sometimes mention CVE strings in contexts unrelated to actual vulnerabilities.

4.2 Feature modelling

A critical step in identifying vulnerable content is constructing a set of features that are representative of OT devices and their associated vulnerabilities. Since the descriptions we collected from the three sources may contain subjective language and may not consistently indicate vulnerabilities, we aimed to derive features that are more broadly representative of vulnerable OT content.

To achieve this, we leveraged two primary sources: specification documentation for OT devices and the CVE database.

To derive *features that can identify OT-related content* beyond device names, we collected datasheets and manuals for 65 devices used in industrial automation and process control. To ensure diversity and coverage, we selected devices from 12 of the top 50 global industrial automation companies, as ranked by Emerson [5]. Since OT devices share some functionalities with IoT devices, it is important to distinguish between the two to avoid overlap. Therefore, we also collected datasheets for 150 IoT devices not used in industrial settings, but commonly found in home, health, and fitness applications (e.g., smart lock, wearable devices, remote patient monitoring devices). The collected datasheets were converted to plain text. The resulting text was tokenized using whitespace as the delimiter, producing two sets of tokens for IoT and OT datasheets. To improve the quality of analysis, we filtered out tokens that were present in both sets, only retaining tokens exclusive to OT domain.

To assess the importance of tokens indicative of OT devices, we further employed Term Frequency–Inverse Document Frequency (TF-IDF) analysis, a widely used technique in natural language processing for evaluating the signif-

icance of terms across datasheets [16]. This technique combines the term frequency (TF) and the inverse document frequency (IDF).

Term Frequency (TF): is used to calculate the occurrence of the word in a datasheet. For a word t in a datasheet d, let $f_{t,d}$ be the number of times t appears in d and $\sum_{t' \in d} f_{t',d}$ the total number of terms in datasheet d.

$$\mathrm{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

The Inverse Document Frequency (IDF): reflect the importance of a word across multiple datasheets by calculating its occurrence across the entire corpus. For term t in a corpus D, let N be total number of datasheets in the corpus and n_t be the number of datasheets wherein term t appears. The number *1* in the denominator is used to avoid division by zero.

$$\mathrm{IDF}(t, D) = \log\left(\frac{N}{1 + n_t}\right)$$

Term Frequency–Inverse Document Frequency (TF-IDF): combines both TF and IDF to determine the significance of a word across a corpus.

$$\text{TF-IDF}(t, d, D) = \mathrm{TF}(t, d) \times \mathrm{IDF}(t, D)$$

Based on the TF-IDF scores, we selected the top 500, 1000, and 2000 tokens as candidate feature sets describing only OT devices.

To derive *features describing vulnerabilities*, we leveraged the CVE database, the official source of disclosed vulnerabilities. Since vulnerability descriptions in such databases are typically very brief (averaging at 42 words), we chose not to use TF-IDF for feature extraction. We employed topic modelling techniques: BERTopic and Latent Dirichlet Allocation (LDA). Topic modelling techniques are unsupervised, meaning they do not require labelled data to guide the analysis. These methods operate under the assumption that by uncovering hidden patterns in the data, they can identify semantically relevant words, which can then be used to annotate new texts. In our context, these techniques can help derive a set of distinguishing features representing descriptions of vulnerabilities.

For this analysis, we extracted all available CVE entries from the official CVE repository spanning the years 1999 to 2024. The descriptions of these CVEs were used to extract features that could help distinguish vulnerable content from benign. We deliberately chose not to restrict the scope to only OT-related CVE descriptions, both to compensate for the limited coverage of OT vulnerabilities in the CVE database and to enable the detection of vulnerabilities affecting OT devices beyond those explicitly disclosed.

For each CVE description, we removed common linking words (e.g., 'and', 'also', 'furthermore', 'moreover', 'but', 'however', 'although', 'so'), then tokenized the text using space as the token separator. This formed the full set of tokens. We refer to this set of tokens as 'All'. We then conducted topic modeling analysis using BERTopic and Latent Dirichlet Allocation (LDA) on this set.

Parameter	XGBoost	Random Forest
n_estimators	100	150
max_depth	5	10
class_weight	–	`balanced_subsample`
scale_pos_weight	N/P	–
random_state	42	42
n_jobs	Default (None)	-1 (all processors)

N and P denote the number of negative and positive samples, respectively.

Table 1: Model parameters for XGBoost and Random Forest algorithms

4.3 Identification of vulnerability description

To enable the identification of OT-related vulnerable content, our proposed system combines two sets of features: one derived from OT device documentation and the other from official vulnerability descriptions in the CVE database. These features are then used to train machine learning classifiers. In this work, we explore two classification algorithms: Random Forest and XGBoost.

Random Forest (RF) [12] is an ensemble learning method that constructs multiple decision trees during training and outputs the class selected by the majority of trees during inference. It is robust to overfitting and performs well with high-dimensional feature spaces. XGBoost (Extreme Gradient Boosting) [8] is a scalable and efficient gradient boosting framework that builds decision trees sequentially. Each new tree attempts to correct the errors made by the previous ones, and the model is optimized using a gradient descent algorithm. XGBoost is known for its high performance on classification tasks, especially in structured data settings.

To build a robust predictive model, we combine RF and XGBoost using a soft voting ensemble approach. RF reduces variance by averaging the predictions of multiple deep decision trees trained on bootstrapped subsets of the data, while XGBoost reduces bias through gradient boosting, iteratively improving performance by focusing on errors made by previous models.

The ensemble approach aggregates the predicted class probabilities from each model and selects the class with the highest average probability. Instead of selecting the final class based on the majority vote of predicted labels (as in hard voting), soft voting averages the predicted probabilities for each class across both models and selects the class with the highest average probability. The soft voting ensemble produced more stable, reliable, and generalizable results by compensating for the individual errors of each model, ultimately enhancing overall robustness.

5 Experimental results

5.1 Experimental setup

The framework was implemented using Python programming language with the following libraries: Py2pdf for conversion of datasheets to plain text, Scikit-learn for TF-IDF scores calculation, NLTK (Natural Language Toolkit) for BERTopic,

Source	# of documents	# unique documents	# unique related to OT devices	# unique with CVE
Full Disclosure	100,757	100,569 (99.8%)	855 (0.9%)	14,843 (14.6%)
Ars Technica	25,276	24,924 (98.6%)	7,567 (29.9%)	460 (1.8%)
Security Now	1,028	1,016 (98.8%)	935 (92%)	169 (16.4%)
Total	127,061	126,509 (99.6%)	9,357 (7.4%)	1,5472(12.2%)

Table 2: Data summary

LDA topic modeling and stop words removal, and finally Pandas and Pyarrow for data processing. All experiments were conducted using 10-fold cross validation. Table 1 states the parameters used for classification algorithms.

All experiments were conducted on a Linux machine running Pop OS, equipped with 134 GB of RAM, 20 CPU cores, and an MSI GeForce RTX 2060 GPU.

5.2 Overview of collected data

Table 2 gives an overview of collected data. Overall, we collected 127,061 messages across three sources. Most of them are unique. Among them, 12.2% contain CVE identifiers, and 7.4% are OT device-related. Of all three sets, Full disclosure dominates in volume with the least proportion of content with CVE mention (0.9%). Ars Technica, on the other hand, has minimal vulnerability coverage (1.7%) and moderate focus on OT related content (30%). Security Now podcast, unlike the other datasets, has the highest rates in terms of contents describing vulnerabilities and OT devices, despite its small size (which can be explained by the nature of the source).

Vulnerability-related features. The feature sets extracted using BERTopic and LDA techniques from CVE descriptions are presented in Table 3. Overall, we obtained 281,206 CVE entries from the official CVE repository. From this corpus, three sets of tokens were identified as the most relevant for detecting vulnerability-related content in text with 13,688, 1,679, and 206 tokens in All, BERTopic, and LDA sets. Interestingly, these token sets primarily consist of alphabetic characters, with alphanumerical characters only appearing in the set "All", and no presence of numeric characters across the three sets.

OT-related features. The results of the device datasheet analysis are presented in Tables 4 and 5. We collected a total of 200 manuals, evenly split between OT and IoT devices. OT device datasheets were generally longer, with the majority exceeding 5 pages in length (80.5%), whereas IoT datasheets were typically shorter, with 55.5% containing fewer than 5 pages. This is likely because OT device datasheets tend to include more detailed information and operational guidelines, as they are primarily intended for industrial use, unlike consumer IoT devices, which typically require simpler documentation. This richer content increases the likelihood of successfully distinguishing OT-related content from IoT-specific content.

Out of the 281,058 tokens collected from OT documents, 261,445 were found exclusively in OT sources. We retain this feature set for our experiments.

Features	Total tokens	Alphabetic tokens	Alphanumerical tokens
BERTopic	1,679	1,679 (100.00%)	0 (0%)
LDA	206	206 (100.00%)	0(0%)
All	13,688	11,174 (81.63%)	2514 (18.37%)

Table 3: Feature sets extracted by topic modelling techniques on CVE descriptions. Only alphabetic and alphanumerical tokens are present in these sets.

Device category	# of documents	Documents (<5 pages)	Documents (5-15 pages)	Documents (> 15 pages)	Unparseable	# of tokens
OT	200	27 (13.5%)	45 (22.5%)	116 (58.0%)	12 (6.0%)	281,058
IoT	200	111 (55.5%)	36 (18%)	53 (26.5%)	0 (0%)	107,295
Total	400	138 (34.5%)	81 (20.2%)	169 (42.2%)	12 (3%)	669,411

Table 4: Overview of extracted datasheets

5.3 Vulnerability identification results

To evaluate the framework, we conduct 3 sets of experiments: ① evaluation of OT-related features, ② evaluation of features for vulnerability content, ③ evaluation of ensemble approach for identifying potential vulnerability descriptions in OT products with a unified feature set.

Classification of OT-related content. Table 6 describes the performance of classifiers across three datasets (Full Disclosure, Ars Technica, SecurityNow).

Based on the results, both models performed similarly well overall, particularly on the Ars Technica and Full Disclosure datasets, achieving 96–98% accuracy. The Random Forest classifier outperformed XGBoost on less structured and noisier sources, namely Full Disclosure and SecurityNow, which aligns with RF's known robustness to noise. XGBoost, on the other hand, showed better performance than Random Forest when evaluated on the Ars Technica dataset.

Performance on the SecurityNow dataset was comparatively low, with RF achieving 92% accuracy and XGBoost ranging between 89% and 93%. This result is likely due to the nature of the data, as it consists of podcast transcripts where the text is less structured and speakers often use informal or ambiguous language, unlike the more formal writing found in the other sources.

Despite variations in model performance across datasets, the results support our initial hypothesis that *the inherent capabilities and specifications of OT devices can be effectively used to detect text describing industrial control systems.*

Classification of vulnerable content. Table 7 presents evaluation results on features derived by topic modelling techniques on CVE dataset.

As the results indicate, a different trend compared to the previous analysis is observed in this set of experiments. The XGBoost classifier generally outperformed RF on most datasets, particularly in terms of accuracy. However, the performance varied significantly across feature sets.

Document Length	Total tokens	OT tokens	IoT tokens	Tokens only in OT set	Common tokens
<5 pages	27,653	4,963 (17.9%)	17,727 (64.1%)	3,083 (11.2%)	1,880 (6.8%)
5-15 pages	55,661	20,531 (36.9%)	14,599 (26.3%)	16,203 (29.1%)	4,328 (7.8%)
>15 pages	586,097	255,564 (43.6%)	74,969 (12.8%)	242,159 (43.1%)	13405 (2.3%)
Total	**669,411**	**281,058** (42%)	**107,295** (16%)	**261,445** (39.1%)	**19,613** (2.9%)

Table 5: Description of tokens per datasheets.

Source	Number of Features	Random Forest			XGBoost		
		Accuracy	Precision	Recall	Accuracy	Precision	Recall
Full disclosure	500	**0.989**	**0.913**	**0.953**	0.970	0.723	0.987
	1000	0.987	0.894	0.941	0.976	0.766	0.987
	2000	0.984	0.875	0.930	**0.979**	**0.790**	**0.987**
Ars Technica	500	**0.992**	**0.996**	**0.976**	0.998	0.999	0.994
	1000	0.985	0.989	0.961	0.998	1.000	0.994
	2000	0.982	0.985	0.950	**0.998**	**1.000**	**0.994**
SecurityNow	500	0.920	0.920	1.000	0.892	0.935	0.948
	1000	0.920	0.920	1.000	0.904	0.933	0.965
	2000	**0.925**	**0.925**	**1.000**	**0.934**	**0.952**	**0.978**

Table 6: Experimental results for OT related content.

For the Full Disclosure dataset, XGBoost achieved the highest accuracy (94.65%) with the full token set, followed by BERTopic-derived features (92.61%) and LDA-derived features (89.33%). This aligns with the observation that larger feature sets (e.g., the full 13,688 tokens) yielded better performance compared to the reduced feature sets from LDA (206 tokens) and BERTopic.

Interestingly, the RF classifier exhibited different trend on Ars Technica and SecurityNow datasets. In Ars Technica, RF achieved higher accuracy than XGBoost (95.61% with LDA vs. 96.98% for XGBoost), but its precision was notably low (0.2815 for RF vs. 0.3529 for XGBoost with LDA). Similarly, in SecurityNow, RF struggled with recall (12.72% with the 'All' token set).

For Ars Technica, the RF classifier's low precision indicates a high rate of false positives, where benign articles were incorrectly flagged as vulnerable. Manual investigation showed that articles discussing CVEs in broader contexts (e.g., economics or politics) were commonly misclassified. For example, the article discussing spear-phishing campaigns targeting both the Trump and Biden presidential campaigns was falsely identified as describing a vulnerability. This ambiguity arises from the mixed signals in such articles terms associated with vulnerabilities may co-occur with unrelated topic specific language, confusing the model.

In SecurityNow, the RF classifier's low recall reveals a failure to detect actual vulnerable content, likely due to the dataset's informal structure (e.g., irregular grammar, abrupt topic shifts). Such noise disrupts RF's reliance on static, hierarchical decision boundaries.

XGBoost, while not perfect, demonstrated more balanced precision-recall trade-offs in these cases. It's superior performance in these cases stems from its iterative error correction and gradient-based optimization, which better handles sparse signals and nonlinear feature interactions. For instance, in SecurityNow,

Source	Features	Random Forest			XGBoost		
		Accuracy	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
Full disclosure	All	**0.908**	**0.626**	**0.905**	**0.947**	**0.749**	**0.948**
	BERTopic	0.895	0.589	0.904	0.926	0.678	0.933
	LDA	0.859	0.507	0.912	0.893	0.584	0.911
Ars Technica	All	0.934	0.207	0.907	**0.985**	**0.571**	**0.757**
	BERTopic	0.942	0.232	0.913	0.976	0.425	0.767
	LDA	**0.956**	**0.282**	**0.874**	0.970	0.353	0.744
SecurityNow	All	0.843	0.615	0.127	**0.871**	**0.640**	**0.544**
	BERTopic	**0.855**	**0.667**	**0.229**	0.863	0.602	0.572
	LDA	0.830	0.498	0.472	0.814	0.441	0.458

Table 7: Experimental results for vulnerable content.

XGBoost achieved a recall of 54.38% (vs. RF's 12.72%) with 'All' token set, demonstrating its adaptability to noisy data.

Beyond performance comparison, the results reveal that regardless of the source of content or the level of text structure, *similar language is consistently adopted to describe vulnerabilities.*

The ensemble approach. The final set of experiments focused on evaluating the framework assembled based on the conclusions drawn from previous analyses.

According to the results, the performance of the classification models for OT related content varied across Full disclosure, Ars Technica, and SecurityNow on different number of features without a clear optimal configuration. The classification of vulnerable content showed that 'All' and BERTopic features give comparable overall results across all three datasets. Since both feature sets involve trade-offs, we retained both for comparison purposes. The full vulnerability-related feature set is considerably larger than the set selected by the BERTopic modeling technique (13,688 vs. 1,679 features), suggesting that while topic modeling effectively reduces dimensionality, it may also exclude terms that contribute to improved detection performance in broader contexts. Based on these findings, we included the full set of features from the previous classifications to better evaluate the final model.

Table 8 illustrates the results obtained for the ensemble model. The results of the model show strong overall performance, particularly on the Full Disclosure dataset, where both precision and recall reach high values across all classifiers. The ensemble model consistently performs best, slightly outperforming XGBoost, while Random Forest tends to lag behind, lowering the overall performance of the ensemble in some cases.

For Security Now dataset, the model performed moderately well with a noticeable decrease in performance compared to Full Disclosure, specifically in precision results. This drop is likely due to the highly unstructured nature of the text in this dataset, which makes it more difficult for the model to accurately identify vulnerable content.

Source	Feature sets		Ensemble			Random Forest			XGBoost		
	Vuln.	OT	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Full Disclosure	All	500	0.996	0.848	0.833	0.986	0.465	0.489	0.996	0.835	0.859
		1000	**0.996**	**0.872**	**0.812**	0.986	0.487	0.472	**0.996**	**0.862**	**0.836**
		2000	0.992	0.828	0.802	**0.985**	**0.687**	**0.618**	0.993	0.878	0.813
	BERTopic	500	0.995	0.833	0.816	0.988	0.567	0.538	0.996	0.841	0.813
		1000	**0.996**	**0.859**	**0.812**	0.989	0.583	0.508	**0.996**	**0.857**	**0.824**
		2000	0.991	0.831	0.771	**0.986**	**0.687**	**0.659**	0.993	0.867	0.784
SecurityNow	All	500	**0.873**	**0.598**	**0.769**	**0.878**	**0.605**	**0.743**	**0.856**	**0.550**	**0.803**
		1000	0.864	0.556	0.760	0.866	0.570	0.764	0.850	0.538	0.752
		2000	0.861	0.556	0.802	0.864	0.579	0.805	0.860	0.566	0.776
	BERTopic	500	0.859	0.554	0.757	0.862	0.565	0.742	0.861	0.561	0.719
		1000	0.855	0.526	0.773	**0.861**	**0.572**	**0.765**	0.845	0.507	0.762
		2000	**0.871**	**0.569**	**0.772**	0.858	0.544	0.801	**0.860**	**0.559**	**0.750**
Ars Technica	All	500	0.992	0.270	0.400	0.993	0.259	0.339	0.992	0.227	0.365
		1000	0.993	0.235	0.370	0.995	0.304	0.260	0.990	0.204	0.408
		2000	**0.995**	**0.524**	**0.468**	**0.992**	**0.319**	**0.441**	**0.996**	**0.505**	**0.521**
	BERTopic	500	0.991	0.212	0.389	0.993	0.289	0.322	0.990	0.230	0.394
		1000	0.990	0.225	0.418	0.991	0.215	0.378	0.991	0.260	0.372
		2000	**0.996**	**0.677**	**0.452**	**0.994**	**0.482**	**0.387**	**0.996**	**0.655**	**0.471**

Table 8: Experimental results for the proposed model.

Interestingly, all models struggled to identify vulnerable OT content in Ars Technica despite being a more structured source. Classification models exhibited low precision and recall on this dataset, with average precision around 25% and recall around 39%.

These results suggest that the articles in structured news discuss vulnerability related content in a broader topic, introducing thereby unwanted noise. This eventually affects the results obtained for the models.

The pattern in Ars Technica is further illustrated in the results obtained from BERTopic-based features. While BERTopic helps structure noisy CVE descriptions into more coherent topics, it does not significantly improve detection of operational technology (OT) related vulnerabilities in Ars Technica.

The trend for Ars Technica, however, changes when 2000 tokens of the OT relted feature set is used. For this particular set, we noticed a much improved results but still lower than those of Full Disclosure even with the more formal structure of text used in Ars Technica. This suggests that a richer feature set allows better detection of vulnerable OT content within this dataset. The results also indicate that OT related content tends to often appear in the context of vulnerability discussions.

The ensemble model generally achieved a more consistent balance between precision and recall compared to both Random Forest and XGBoost across most datasets and feature sets. While XGBoost occasionally produced higher recall (notably on the Ars Technica dataset), the ensemble model consistently maintained superior or comparable performance in both metrics. Random Forest, on the other hand, exhibited notably lower precision, especially on noisier datasets which affected its reliability in detecting vulnerable OT-related content. These results suggest that the ensemble approach effectively mitigates individual model weaknesses and enhances robustness in vulnerability detection tasks.

Despite challenges for Ars Techncia dataset, the model shows promise. Its strong performance on unstructured sources like Full Disclosure and Security Now indicates its robustness in noisy environments.

6 Conclusion

The increasing convergence of IT and OT networks has introduced significant cybersecurity challenges, exposing OT Systems to previously unforeseen threats. Traditional vulnerability databases often fail to timely capture these emerging risks which necessitate proactive approaches to identify undisclosed OT vulnerabilities. Our framework addresses this gap by monitoring unofficial online sources such as news websites, mailing lists, and security podcasts while leveraging device specifications to filter OT-relevant discussions. By analyzing linguistic patterns against known vulnerability descriptions, we distinguish actual vulnerabilities from benign content. Our experimental results demonstrate the effectiveness of this approach across diverse datasets, including structured and unstructured sources. The Random Forest classifier exhibited superior robustness in noisy, less structured environments (i.e., Full Disclosure and Security Now), while XGBoost performed better in more organized contexts (i.e., Arstechnica). Notably, utilizing all CVE-derived features significantly improved detection accuracy, reinforcing the importance of comprehensive linguistic analysis in identifying vulnerability discussions. These findings also validate our hypothesis that OT device specifications based filtering can effectively isolate OT related vulnerabilities in public sources. Moreover, they highlight the consistency in vulnerability descriptions across different sources, regardless of their structure. By bridging the intelligence gap in OT security, our framework enables prediction of potential risks, enhancing thereby resilience of critical infrastructure against cyber threats that could compromise safety, operations, and environmental integrity.

References

1. Alevizopoulou, S., Koloveas, P., Tryfonopoulos, C., Raftopoulou, P.: Social media monitoring for iot cyber-threats. In: 2021 IEEE International Conference on Cyber Security and Resilience (CSR). pp. 436–441 (2021)
2. Almukaynizi, M., Grimm, A., Nunes, E., Shakarian, J., Shakarian, P.: Predicting cyber threats through hacker social networks in darkweb and deepweb forums. In: Proceedings of the 2017 International Conference of The Computational Social Science Society of the Americas. CSS 2017, Association for Computing Machinery, New York, NY, USA (2017)
3. Alves, F., Bettini, A., Ferreira, P.M., Bessani, A.: Processing tweets for cybersecurity threat awareness. Information Systems **95**, 101586 (2021)
4. Anwar, A., Abusnaina, A., Chen, S., Li, F., Mohaisen, D.: Cleaning the nvd: Comprehensive quality assessment, improvements, and analyses. IEEE Transactions on Dependable and Secure Computing **19**(6), 4255–4269 (2022)
5. Boyes, W., O'Brien, L.: The 50 largest automation companies around the world keep on keepin' on despite the recession. Control Magazine (December 2009)

6. Bozorgi, M., Saul, L.K., Savage, S., Voelker, G.M.: Beyond heuristics: learning to classify vulnerabilities and predict exploits. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 105–114. KDD '10, Association for Computing Machinery, New York, NY, USA (2010)
7. Burnap, P., Javed, A., Rana, O.F., Awan, M.S.: Real-time classification of malicious urls on twitter using machine activity data. In: 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 970–977 (2015)
8. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 785–794. KDD '16, Association for Computing Machinery, New York, NY, USA (2016)
9. Dong, Y., Guo, W., Chen, Y., Xing, X., Zhang, Y., Wang, G.: Towards the detection of inconsistencies in public security vulnerability reports. In: 28th USENIX Security Symposium (USENIX Security 19). pp. 869–885. USENIX Association, Santa Clara, CA (Aug 2019), https://www.usenix.org/conference/usenixsecurity19/presentation/dong
10. Edkrantz, M., Said, A.: Predicting cyber vulnerability exploits with machine learning. In: Scandinavian Conference on AI (2015), https://api.semanticscholar.org/CorpusID:12126104
11. Elbaz, C., Rilling, L., Morin, C.: Fighting n-day vulnerabilities with automated cvss vector prediction at disclosure. In: Proceedings of the 15th International Conference on Availability, Reliability and Security. ARES '20, Association for Computing Machinery, New York, NY, USA (2020)
12. Ho, T.K.: Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition. vol. 1, pp. 278–282. IEEE (1995)
13. Horawalavithana, S., Bhattacharjee, A., Liu, R., Choudhury, N., O. Hall, L., Iamnitchi, A.: Mentions of security vulnerabilities on reddit, twitter and github. In: IEEE/WIC/ACM International Conference on Web Intelligence. p. 200–207. WI '19, Association for Computing Machinery, New York, NY, USA (2019)
14. Huang, S.Y., Ban, T.: Monitoring social media for vulnerability-threat prediction and topic analysis. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). pp. 1771–1776 (2020)
15. Jiang, Y., Atif, Y.: Towards automatic discovery and assessment of vulnerability severity in cyber–physical systems. Array **15**, 100209 (2022)
16. Kadhim, A.I.: Term weighting for feature extraction on twitter: A comparison between bm25 and tf-idf. In: 2019 International Conference on Advanced Science and Engineering (ICOASE). pp. 124–128 (2019). https://doi.org/10.1109/ICOASE.2019.8723825
17. Le, B.D., Wang, G., Nasim, M., Babar, M.A.: Gathering cyber threat intelligence from twitter using novelty classification. In: 2019 International Conference on Cyberworlds (CW). pp. 316–323 (2019)
18. Le Sceller, Q., Karbab, E.B., Debbabi, M., Iqbal, F.: Sonar: Automatic detection of cyber security events over the twitter stream. In: Proceedings of the 12th International Conference on Availability, Reliability and Security. ARES '17, Association for Computing Machinery, New York, NY, USA (2017)
19. Manai, E., Mejri, M., Fattahi, J.: Helping cnas generate cvss scores faster and more confidently using xai. Applied Sciences **14**(20) (2024), https://www.mdpi.com/2076-3417/14/20/9231

20. Miranda, L., Senos, L., Menasché, D., Srivastava, G., Kocheturov, A., Ramchandran, A., Lovat, E., Limmer, T.: Learning cna-oriented cvss scores. In: 2024 IEEE 13th International Conference on Cloud Networking (CloudNet). pp. 1–5 (2024)
21. Mittal, S., Das, P.K., Mulwad, V., Joshi, A., Finin, T.: Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 860–867 (2016)
22. Nunes, E., Diab, A., Gunn, A., Marin, E., Mishra, V., Paliath, V., Robertson, J., Shakarian, J., Thart, A., Shakarian, P.: Darknet and deepnet mining for proactive cybersecurity threat intelligence. In: 2016 IEEE Conference on Intelligence and Security Informatics (ISI). pp. 7–12 (2016)
23. Qin, Y., Xiao, Y., Liao, X.: Vulnerability intelligence alignment via masked graph attention networks. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. p. 2202–2216. CCS '23, Association for Computing Machinery, New York, NY, USA (2023)
24. Sabottke, C., Suciu, O., Dumitraş, T.: Vulnerability disclosure in the age of social media: exploiting twitter for predicting real-world exploits. In: Proceedings of the 24th USENIX Conference on Security Symposium. p. 1041–1056. SEC'15, USENIX Association, USA (2015)
25. Sapienza, A., Ernala, S.K., Bessi, A., Lerman, K., Ferrara, E.: Discover: Mining online chatter for emerging cyber threats. In: Companion Proceedings of the The Web Conference 2018. p. 983–990. WWW '18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2018)
26. Sauerwein, C., Sillaber, C., Huber, M.M., Mussmann, A., Breu, R.: The tweet advantage: An empirical analysis of 0-day vulnerability information shared on twitter. In: Janczewski, L.J., Kutyłowski, M. (eds.) ICT Systems Security and Privacy Protection. pp. 201–215. Springer International Publishing, Cham (2018)
27. Shah, S., Madisetti, V.K.: Mad-cti: Cyber threat intelligence analysis of the dark web using a multi-agent framework. IEEE Access **13**, 40158–40168 (2025)
28. Shahid, M.R., Debar, H.: Cvss-bert: Explainable natural language processing to determine the severity of a computer security vulnerability from its description. In: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 1600–1607 (2021)
29. Thomas, R.J., Gardiner, J., Chothia, T., Samanis, E., Perrett, J., Rashid, A.: Catch me if you can: An in-depth study of cve discovery time and inconsistencies for managing risks in critical infrastructures. In: Proceedings of the 2020 Joint Workshop on CPS&IoT Security and Privacy. p. 49–60. CPSIOTSEC'20, Association for Computing Machinery, New York, NY, USA (2020)
30. Yamamoto, Y., Miyamoto, D., Nakayama, M.: Text-mining approach for estimating vulnerability score. In: 2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS). pp. 67–73 (2015)
31. Yin, J., Tang, M., Cao, J., Wang, H.: Apply transfer learning to cybersecurity: Predicting exploitability of vulnerabilities by description. Knowledge-Based Systems **210**, 106529 (2020)
32. Zhang, S., Cai, M., Zhang, M., Zhao, L., de Carnavalet, X.d.C.: The flaw within: Identifying cvss score discrepancies in the nvd. In: 2023 IEEE International Conference on Cloud Computing Technology and Science (CloudCom). pp. 185–192 (2023)