Universidade Federal do Rio de Janeiro -- UFRJ

Instituto de Computação -- IC

Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais – NCE

Programa de Pós-Graduação em Informática -- PPGI

Felipe Adrian Moreno Vera

Automated Detection of Vulnerability Exploitation in Underground Hacking Forums

Rio de Janeiro

2024

Universidade Federal do Rio de Janeiro -- UFRJ

Instituto de Computação -- IC

Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais – NCE

Programa de Pós-Graduação em Informática -- PPGI

Felipe Adrian Moreno Vera

Automated Detection of Vulnerability Exploitation in Underground Hacking Forums

Dissertação de Mestrado submetida ao Programa de Pós-graduação em Informática do Instituto de Computação (IC) e do Insti-tuto Tércio Pacitti de Aplicações e Pesquisas Computacionais (NCE), da Universidade Fed-eral do Rio de Janeiro (UFRJ). Como parte dos requisitos necessários à obtenção do título de Mestre em Informática.

Supervisor: Ph.D. Daniel Sadoc Menasché

Rio de Janeiro

2024

CIP - Catalogação na Publicação

M843a

Moreno Vera, Felipe Adrian
Automated detection of vulnerability exploitation in underground hacking forums / Felipe Adrian Moreno Vera. -- Rio de Janeiro, 2024.
66 f.

Orientador: Daniel Sadoc Menasché.
Dissertação (mestrado) - Universidade Federal do Rio de Janeiro, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Programa de Pós-Graduação em informática, 2024.

1. Processamento de linguagem natural. 2. Cibersegurança. 3. Fóruns online. 4. Mineração de dados. 5. Segurança da informação. I. Menasché, Daniel Sadoc, orient. II. Título.

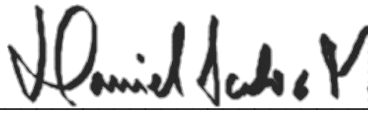
Elaborado pelo Sistema de Geração Automática da UFRJ com os dados fornecidos pelo(a) autor(a), sob a responsabilidade de Miguel Romeu Amorim Neto - CRB-7/6283.

Automated Detection of Vulnerability Exploitation in Underground Hacking Forums

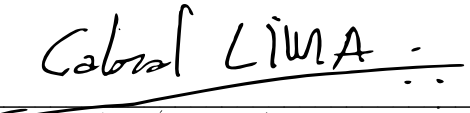
Felipe Adrian Moreno Vera

Dissertação de Mestrado submetida ao Programa de Pós-graduação em Informática do Instituto de Computação (IC) e do Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais (NCE), da Universidade Federal do Rio de Janeiro (UFRJ). Como parte dos requisitos necessários à obtenção do título de Mestre em Informática.

Aprovada em 2 de dezembro de 2024 por:

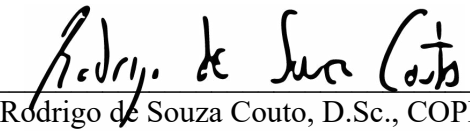


Prof. Daniel Sadoc Menasché, Ph.D., PPGI/UFRJ (Presidente)



Prof. Josefino Cabral Melo Lima, Docteur, PPGI/UFRJ

Prof. Claudio Miceli de Farias, D.Sc., PESC/UFRJ e PPGI/UFRJ



Prof. Rodrigo de Souza Couto, D.Sc., COPPE/UFRJ

Este trabalho é dedicado ao apoio e orientações recebidas por familiares, companheiros alunos e professores colaboradores. Também aos meus familiares que sempre estão me apoiando nas minhas decisões e me motivando a continuar.

ACKNOWLEDGEMENTS

Os agradecimentos principais são direcionados aos alunos e professores que estiveram envolvidos e contribuíram com este trabalho. Além disso, agradecemos à Universidade de Cambridge, por compartilhar conosco os dados do CrimeBB, usados nesta proposta como estudo de caso. Agradecemos também às entidades de desenvolvimento científico, incluindo **Coordenação de Aperfeiçoamento de Pessoal de Nível Superior** (CAPES – 315110/2020-1), **Conselho Nacional de Desenvolvimento Científico e Tecnológico** (CNPq – E-26/211.144/2019) e **Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro** (FAPERJ – E-26/201.376/2021) que ajudaram neste período de Mestrado em Informática na **Universidade Federal do Rio de Janeiro** (UFRJ), no **Programa Pós-Graduação em Informática** (PPGI).

RESUMO

Este trabalho propõe uma abordagem baseada em aprendizado de máquina para identificar e classificar a exploração de vulnerabilidades, o escopo e o impacto de softwares maliciosos por meio do monitoramento de fóruns de hackers clandestinos. O volume crescente de postagens discutindo a exploração de vulnerabilidades exige uma abordagem automatizada para processar tópicos e postagens que possam acionar alarmes com base em seu conteúdo. Para ilustrar o sistema proposto, utilizamos o conjunto de dados CrimeBB, composto por dados extraídos de vários fóruns clandestinos, e desenvolvemos um modelo de aprendizado de máquina supervisionado capaz de filtrar tópicos que citam CVEs e rotulá-los como prova de conceito (PoC), armamento (Weaponization), exploração (Exploitation), entre outros. Aplicamos técnicas de aprendizado de máquina e processamento de linguagem natural para pré-processar os textos e, em seguida, treinamos diversos modelos lineares e não lineares. Para avaliar a eficácia e a precisão dos modelos e comparar os resultados, utilizamos as métricas de exatidão (accuracy), precisão (precision) e recuperação (recall). Além disso, para maior compreensão e explicação sobre o motivo de o modelo diferenciar entre as classes, utilizamos métodos de explicação de modelos para determinar a relevância das palavras nas predições. No geral, este trabalho destaca as diferenças entre as naturezas das postagens rotuladas e sua importância na análise de vulnerabilidades.

Palavras-chave: Cíber segurança; fóruns online; mineração de dados; processamento de linguagem natural; segurança da informação; modelos de linguagem; interpretabilidade.

ABSTRACT

This work proposes a machine learning-based approach to identify and classify vulnerability exploitation, the scope, and the impact of malicious software through the monitoring of underground hacker forums. The growing volume of posts discussing vulnerability exploitation demands an automated approach to process topics and posts that may trigger alarms based on their content. To illustrate the proposed system, we used the CrimeBB dataset, which contains data extracted from various underground forums, and developed a supervised machine learning model capable of filtering topics that cite CVEs and labeling them as proof of concept (PoC), weaponization, exploitation, among others. We applied machine learning and natural language processing techniques to preprocess the texts and then trained several linear and non-linear models. To evaluate the models' effectiveness and accuracy and compare results, we used metrics such as accuracy, precision, and recall. Additionally, to gain a better understanding and explanation of why the model can differentiate between classes, we employed model explanation methods to determine the relevance of words in predictions. Overall, this work highlights the differences in the nature of the labeled posts and their importance in vulnerability analysis.

Keywords: Cybersecurity; online forums; data mining; natural language processing; information security; large language models; Generative Pre-trained Transformer; interpretability.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
EPSS	Exploit Prediction Scoring System
GPT	Generative Pre-trained Transformer
NLP	Natural Language Processing
NVD	National Vulnerability Database

CONTENTS

1 INTRODUCTION

In this chapter, we describe our main motivations for developing our work. We will also define the problem we intend to address and the objectives we aim to achieve in this work.

1.1 CONTEXT AND MOTIVATION

The global impact of cybercrime on corporations and government organizations is significant, with investments in countermeasures reaching about 2 trillion dollars over the years (ANDERSON et al., 2019; MORROW, 2019). By exploiting vulnerabilities in computer software and hardware, various internet ecosystems can be easily deceived or attacked. This poses a threat to end-users, companies, and the overall stability of the internet. Therefore, early detection of weaponization and attempted exploitation is critical in defending against such attacks.

Exploiting vulnerabilities involves using a weaponized exploit to attack a target, allowing the attacker to exploit a vulnerability to gain unauthorized access to a system or steal sensitive information. Although public databases such as [1]ExploitDB continuously provide updated information on how to exploit vulnerabilities, underground hacking forums still offer exclusive and more current information on the availability and development of exploits and on tentative use of these exploits in the wild. These forums are documented in recent studies (BASHEER; ALKHATIB, 2021; CAMPOBASSO; ALLODI, 2022; PASTRANA; THOMAS et al., 2018).

In particular, monitoring underground hacking forums is crucial in detecting and neutralizing the exploitation of vulnerabilities in the wild and identifying the main hacking users. Certain forums also provide information on the development of new exploits, the latest prices, and instructions on how to make attacks Fully UnDetectable (FUD) (ABLON; LIBICKI; GOLAY, 2014). In this context, understanding what users are discussing in these forums helps in tracking exploit prices, usage, demand, and main targets.

The development of a cyberattack involves several stages, including weaponization and exploitation. While most of the literature has focused on weaponization (HANKS et al., 2022; ALLODI, 2017), i.e., building exploits for vulnerabilities, exploitation in the wild has received less attention. Exploitation refers to the actual use of a weaponized exploit to attack a target or gain unauthorized access to a system or sensitive information (BASHEER; ALKHATIB, 2021). This lack of attention is partly due to the involvement of sensitive

[1] https://www.exploit-db.com/

data and strict non-disclosure agreements in studying exploitation in the wild.

In this study, we utilize the CrimeBB dataset, a compilation of data scraped from various underground forums, provided by the Cambridge Cybercrime Centre (PASTRANA; THOMAS et al., 2018). In our work (MORENO-VERA et al., 2023), we have proposed a machine learning approach, leveraging the CrimeBB dataset, to classify forum posts discussing Common Vulnerabilities and Exposures (CVE) into categories like PoC and Exploitation, achieving F1 scores above 90%. Ground truth information was not available at CrimeBB, and a major effort in previous works involved the manual labeling of posts and threads. we also apply an unsupervised learning approach to infer the topics based on the thread content (MORENO-VERA, 2023), containing many words such as "server" or "tool" could belong to Proof-of-concept or Weaponization discussion. In addition, a new extension to CrimeBB, namely PostCog, provides (noisy) ground truth labels for a subset of CrimeBB posts. For further analysis, we also integrate the PostCog extension into the dataset, allowing for enhanced labeling for each post type, intent, and crime type. Notably, terms like "fud" (fully undetectable) and "pm" (private message) emerged as significant indicators of exploitation. Those terms already appeared as relevant in (MORENO-VERA et al., 2023), and our new results leveraging the PostCog extension serve to reinforce some of our previous findings.

1.2 OBJECTIVES

In this section, we will outline the objectives of this work concisely. Our primary goal is to investigate methods for predicting discussions related to cybercrime. To achieve this, we utilize the CrimeBB dataset and propose a methodology for performing this task. The following are the objectives established in this work:

1.2.1 General objective

The general objective of this work is to assess and evaluate various text-based models for predicting discussions related to cybercrime in posts and threads using the *CrimeBB* dataset and the PostCog extension. To achieve this goal, we will explore, analyze, and present the findings of our study.

1.2.2 Specific objectives

In particular, we list the specific objectives briefly mentioned in the main objective:

- We propose a methodology to explore and analyze the *CrimeBB* dataset, which contains 117,365,492 posts from 37 underground-forum websites. Our aim is to

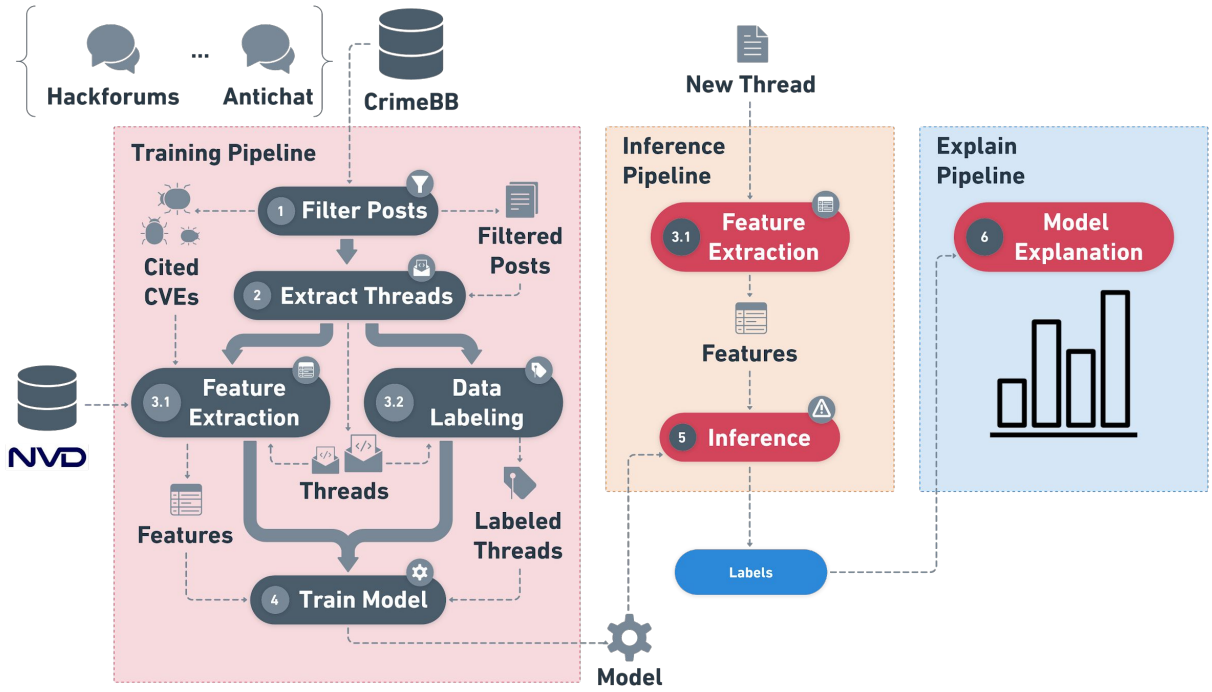


Figure 1 – Proposed framework composed of two main steps: (1) Threads and posts pre-processing including feature extraction and labeling; (2) three-class classifier model to predict new threads content; (3) generate model explanations to understand predictions.

provide insights into the behavior and patterns within the data. The primary focus of our study is the *HackForums* forum.

- We propose and present a text-based model that can efficiently classify the discussion about cybercrime in thread-post content while considering the behavior and distribution of the analyzed data.
- We apply explanation methods to understand and analyze our model's predictions, aiming to identify relevant words that provide insights into the discussions within a forum.
- We use Generative Pre-trained Transformers (GPT) to analyze and obtain new information from posts and their content. We aim to evaluate the performance of GPT models when classifying unstructured information such as forum content.

1.3 METHODOLOGY

The proposed methodology, illustrated in Figure 1, consists of three primary tasks: (i) data preparation, labeling, feature extraction, and training a predictive model; (iii) predicting new samples; and (iii) explain models. In the data preparation phase (steps 1-3), we collect data from both NVD and CrimeBB, filtering the CrimeBB posts to retain

only those that reference CVE identifiers and extracting all cited CVEs. We then retrieve the threads containing these posts and automatically generate a set of features for each thread. Next, using labels provided by PostCog, expert annotations, and GPT, we train a machine-learning model to classify the content of discussion threads (step 4). In the inference phase (step 5), we apply the trained model to classify the content of new thread posts, predicting the most relevant labels based on their text content. Finally, we apply explanation techniques (step 6) to analyse predictions and obtain insights from inferences.

1.4 CONTRIBUTIONS

We have already published insights and results obtained from experiments: (i) We study the content of threads in HackForums, analyzing the CVSS and EPSS of CVE codes within posts; this work is published at IEEE CSR: *IEEE International Conference on Cyber Security and Resilience (CSR) '23*. (ii) We use topic modeling to analyze and infer what is being discussed within threads and posts without annotations, this work is published at IEEE ICTC: *IEEE International Conference on Information and Communication Technology Convergence (ICTC) '23*. (iii) We analyze the PostCog extension and GPT outputs, compare them with ensemble models, and apply model explanations, this work will be presented at *The International Symposium on Cyber Security, Cryptology and Machine Learning (CSCML) '24*.

- Felipe Moreno-Vera, Daniel S. Menasché, and Cabral Lima. "Beneath the Cream: Unveiling Relevant Information Points from CrimeBB Underground Forums with Its Ground Truth Labels". In *The International Symposium on Cyber Security, Cryptology and Machine Learning* (CSCML '24), December 2024, Be'er Sheva, Israel, (MORENO-VERA; MENASCHE; LIMA, 2024).

- Felipe Moreno-Vera, Mateus Nogueira, Cainã Figueiredo, Daniel S. Menasché, Miguel Bicudo, Ashton Woiwood, Enrico Lovat, Anton Kocheturov, and Leandro Pfleger de Aguiar. "Cream Skimming the Underground: Identifying Relevant Information Points from Online Forums". In *IEEE International Conference on Cyber Security and Resilience* (CSR '23), August 2023, Venice, Italy, (MORENO-VERA et al., 2023).

- Felipe Moreno-Vera. "Inferring Discussion Topics about Exploitation of Vulnerabilities from Underground Hacking Forums". In *International Conference on Information and Communication Technology Convergence* (ICTC '23), October 2023, Seoul, South Korea, (MORENO-VERA, 2023)

1.5 DOCUMENT STRUCTURE

The remainder of this paper is structured as follows. In Chapter 2 provides important concepts and definitions for a better understanding of this document. Chapter 3 discusses related work. Chapter 4, describes the dataset used, the exploratory analysis made, and our methodology. Chapter 5 presents our main experiments and results, and Chapter 6 concludes.

2 BACKGROUND

In this chapter, we describe the foundational concepts of machine learning, deep learning, and their application in natural language processing. Through a comprehensive description of key methods, techniques, and models, we aim to provide and define the basic concepts for understanding the following chapters.

2.1 MACHINE LEARNING AND DEEP LEARNING

Machine learning and deep learning are two powerful branches of artificial intelligence that have transformed how we tackle complex problems across diverse fields. Fundamentally, both approaches aim to enable computers to learn from data, allowing them to make predictions or decisions without explicit programming for each specific task (BENGIO, 2007). The combination of machine learning and deep learning has driven remarkable advancements across a broad spectrum of applications, including self-driving cars, virtual assistants, medical diagnostics, urban perception analysis, and recommendation systems.

Machine learning includes a variety of algorithms and techniques, ranging from classical methods like linear regression and decision trees to more advanced approaches such as support vector machines and random forests. It enables computers to learn patterns and relationships from data, allowing them to make informed predictions and decisions. Moreover, Deep learning has become a powerful approach for handling complex and high-dimensional data, especially in fields like natural language processing (NLP), where text data presents unique challenges. At its core, deep learning relies on neural networks inspired by the structure of human neurons. These networks are composed of layers of interconnected neurons that process and transform data, learning representations of language directly from raw text inputs (GOODFELLOW; BENGIO; COURVILLE, 2016). In NLP, models like recurrent neural networks (RNNs) handle sequential data, capturing temporal dependencies, while transformers revolutionize language processing by leveraging self-attention to understand context across entire sentences or documents. Deep learning in NLP can be applied through various types of learning: supervised (using labeled text for classification or translation), unsupervised (finding patterns without labels, topic modeling), and semi-supervised (Generative Pre-trained Transformer). These learning types encompass different tasks and problem formulations, enabling deep learning models to excel in various NLP applications, from sentiment analysis to language generation.

2.1.1 Learning type

A learning type refers to the general approach or paradigm that defines how a machine learning algorithm learns from data. The primary types of learning in machine learning include:

- **Supervised Learning**: In supervised learning, the model is trained on a labeled dataset, where each data point is associated with a specific label or target outcome. The system learns to associate input data with output labels by minimizing a loss function, which improves its predictive accuracy over time. Examples of supervised learning tasks include text classification, machine translation, Part-of-Speech (POS) Tagging, and Named Entity Recognition (NER).
- **Unsupervised Learning**: Unsupervised learning trains a model on an unlabeled dataset, where the input data has no associated labels or targets. Instead, the model learns to uncover patterns, structures, or relationships within the data, such as identifying clusters or latent features. Examples of unsupervised learning tasks include clustering, topic modeling, word embedding, and dimensionality reduction.
- **Semi-supervised/self-supervised Learning**: Semi-supervised learning merges aspects of both supervised and unsupervised learning. In this approach, the system is trained on a dataset that contains a small number of labeled examples alongside a larger set of unlabeled data. The model uses the labeled data to guide its learning while also utilizing the unlabeled data to improve its performance and generalize better. Examples of semi-supervised/self-supervised learning tasks include text generation, sentence embedding, and text summary.

2.2 NATURAL LANGUAGE PROCESSING (NLP)

Natural Language Processing (NLP) is a field of artificial intelligence (AI) focused on enabling machines to understand, interpret, and generate human language. The need for NLP arises from the fact that human language is inherently complex, rich in nuance, and often ambiguous, making it a challenging area for machines to process effectively. Figure 3 summarizes the embedding results from the different techniques.

2.2.1 Text processing techniques

In Natural Language Processing (NLP), data preprocessing is critical for improving the accuracy and efficiency of text analysis. Here's a breakdown of some essential text-cleaning steps and why they are so important:

Removing stopwords

Stopwords are commonly used words in a language (like “the”, “is”, “and”, “in”, etc.) that typically do not carry significant meaning. Removing stopwords helps reduce noise in text data, allowing models to focus on more meaningful words. By filtering out these high-frequency, low-value terms, the dimensionality of the text data decreases, improving both model performance and computational efficiency. For example, in a sentence like, “The cat is on the mat”, removing stopwords yields “cat mat”, which is more meaningful for the model.

Removing emojis

Emojis are pictorial icons commonly used in digital communication to convey emotions, actions, or objects. Emojis can either introduce noise or, depending on the context, provide sentiment information. For general NLP tasks where the focus is on words, removing emojis can simplify the data. However, in sentiment analysis, they can sometimes provide valuable insights if processed correctly.

Removing punctuation

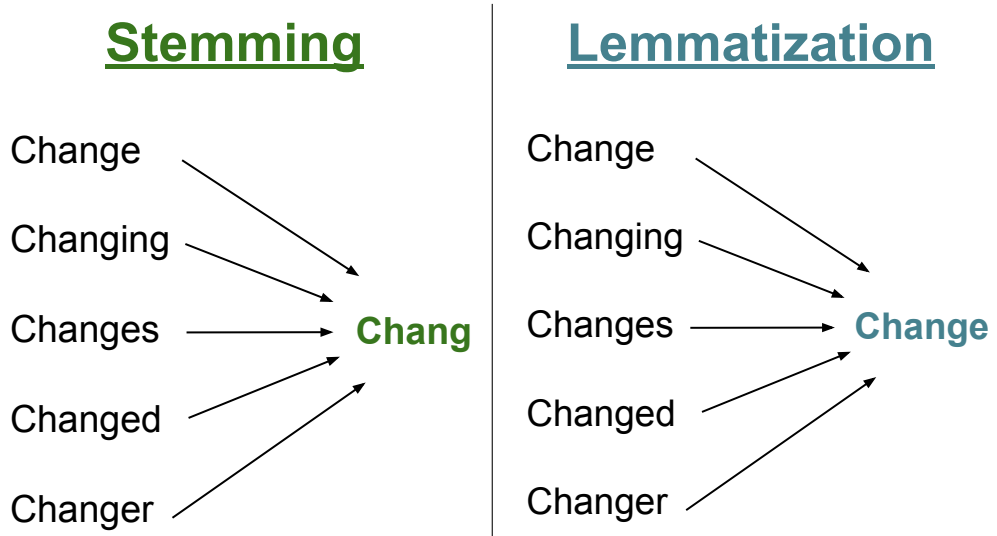
Punctuation marks (like “.”, “,”, “!”, “?”) are symbols used to structure sentences and clarify meaning but are often not needed for NLP models. Punctuation can be redundant in NLP tasks focused solely on word content, such as text classification or topic modeling. Removing it reduces text complexity and improves tokenization, making it easier to analyze. For example, for the text “Hello, world!” removing punctuation yields “Hello world”, which is simpler to process.

Removing special characters

Special characters (like “#”, “@”, “$”, and “&”) are symbols that may not convey relevant information for typical NLP tasks. Special characters are often unnecessary in text analysis and can add noise, especially in data from social media, where symbols are common. Cleaning them out improves tokenization and reduces the vocabulary size, making models easier to train. For example, text such as “Hello @user! Check this out: #NLP #ML” can be reduced to “Hello Check this out NLP ML” by removing special characters.

Reduce words

Stemming and Lemmatization are both Natural Language Processing (NLP) techniques used to reduce words to their root forms. However, they approach this goal differently: (i) Stemming is a rule-based process that cuts off the end of words in order to reduce them



(a) Stemming (b) Lemmatization

Figure 2 – NLP reduce word techniques: (a)Stemming and (b) Lemmatization.

to a base or "stem" form. This method is often crude and may not produce actual words. (ii) Lemmatization is a more sophisticated method that reduces words to their base or dictionary form, called a "lemma". It takes into account the word's meaning and part of speech, resulting in more accurate root forms.

Figure 2 show both methods and their reduction approach. In general, lemmatization is more accurate than stemming as it considers the context of the word. On the other hand, Stemming is faster but may create non-existent words, while lemmatization is slower but yields real words.

By removing stopwords, emojis, punctuation, special characters, and reducing words, text preprocessing makes NLP models more efficient, easier to analyze, and better at generalization.

2.2.2 Early approaches for text-embedding

Text processing began with simple tasks like tokenization, the process of splitting a sentence into smaller units, like words or phrases. These small pieces of text, called "tokens", formed the building blocks for deeper understanding. However, early methods struggled to capture the richness of language. Simple rules-based models could detect basic patterns, but they fell short when faced with the complexities of human communication, such as sarcasm or ambiguity. One of the most influential techniques was bag-of-words (BoW) (HARRIS, 1954), where text was transformed into a set of word frequencies using a Document-Term Matrix (DTM), where rows correspond to documents and columns represent the terms (words or tokens) that appear across all the documents.

Bag-Of-Words (BoW) helped machines to identify important words, but it ignored word order, which made understanding context difficult. Further, TF-IDF (Term Frequency-

Corpus

Document 1	I like cats
Document 2	cats are the best, they are awesome
Document 3	also dogs are nice

Document-Term Matrix

Words	I	like	cats	are	the	best	they	awesome	also	dogs	nice
Document 1	1	1	1	0	0	0	0	0	0	0	0
Document 2	0	0	1	2	1	1	1	1	0	0	0
Document 3	0	0	0	1	0	0	0	0	1	1	1

Doc2Vec

Vectors	x1	x2	x3	x4	x5	x6	x7	...
Document 1	0.35	0.86	1.82	3.48	1.05	10.15	8.63	...
Document 2	0.84	0.45	3.45	4.49	2.64	2.87	13.97	...
Document 3	0.39	1.0	0.98	7.92	5.14	6.19	20.98	...

Bag-Of-Words (1-2-gram)

Words	I	like	cats	I like	like cats	are	the	best	they	awesome	cats are	the best	they are	...
Document 1	1	1	1	1	1	0	0	0	0	0	0	0	0	...
Document 2	0	0	1	0	0	2	1	1	1	1	1	1	1	...
Document 3	0	0	0	0	0	1	0	0	0	0	0	0	0	...

TF-IDF (1-2-gram)

Words	I	like	cats	I like	like cats	are	the	best	they	awesome	cats are	the best	they are	...
Document 1	0	0	0.47	0.62	0.62	0	0	0	0	0	0	0	0	...
Document 2	0	0	0.33	0	0	0.56	0.43	0	0	0	0.43	0.43	0.13	...
Document 3	0	0	0	0	0	0.35	0	0	0	0	0	0	0	...

Figure 3 – Example of the methods Bag-Of-Word, TF-IDF, and Doc2Vec. The first two methods use a pre-constructed from a Document-Term Matrix (DTM). Doc2Vec method embed paragraph or sentence into numerical vector representations.

Inverse Document Frequency) (SALTON; BUCKLEY, 1988) can be seen as an enhancement over the bag-of-words (BoW) model. TF-IDF helps to address a limitation of the BoW model: it reduces the impact of commonly occurring words (such as “the”, “is”, and “or”), which might be frequent across all documents but don’t provide much meaningful insight. In summary, while BoW only focuses on the frequency of words within a document, TF-IDF goes a step further by adjusting the word frequencies based on their importance across the entire corpus of text.

Bag-of-Words (BoW)

The Bag-of-Words model represents a text (document) as an unordered collection of words, disregarding grammar and word order. Each document is represented as a vector of word counts or frequencies. The equation for BoW can be written as:

Let D be a set of documents, and $V = w_1, w_2, \ldots, w_k$ be a vocabulary of size k. For a given document $d \in D$ the BoW representation v_d is a vector where each entry represents the frequency of a word in the document:

$$v_d = [freq(w_1, d), freq(w_2, d), \ldots, freq(w_k, d)] \tag{2.1}$$

Where $freq(w_i, d)$ is the number of times word w_i appears in document d.

TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF is a statistical measure used to evaluate the importance of a word in a document relative to a corpus. It is computed by multiplying Term Frequency (TF) and Inverse Document Frequency (IDF):

- **Term Frequency (TF)**: Measures how frequently a term appears in a document:

$$TF(w,d) = \frac{freq(w,d)}{\sum_{i \in d} freq(w_i,d)} \tag{2.2}$$

- **Inverse Document Frequency (IDF)**: Measures how important a word is across the entire corpus:

$$IDF(w,D) = log(\frac{|D|}{1+|d \in D : w \in d|}) \tag{2.3}$$

Where $freq(w,d)$ is the frequency of word w in document d, and the denominator is the sum of frequencies of all words in document d, $|D|$ is the total number of documents in the corpus and $|d \in D : w \in d|$ is the number of documents containing word w. Finally, TF-IDF is the product of the TF and IDF.

2.2.3 Machine learning and word-embeddings

However, the true revolution in text processing came with the advent of deep learning. Neural networks, inspired by the way human brains work, began to change the game. The introduction of word embeddings, like Word2Vec (MIKOLOV et al., 2013) and GloVe (PENNINGTON; SOCHER; MANNING, 2014), allowed words to be represented as dense vectors in multi-dimensional space. These vectors captured the semantic meaning of words based on their context, enabling machines to understand that "king" and "queen" were more closely related than "king" and "car".

Later, Doc2Vec (LE; MIKOLOV, 2014) expanded on this concept by creating vector representations not just for individual words but also for longer texts, such as sentences, paragraphs, or documents. This allows machines to understand the meaning and context of longer text units in a similar way that Word2Vec captures word relationships. In summary, while Word2Vec and GloVe focus on learning word-level embeddings by capturing the semantic relationships between words, Doc2Vec extends this concept to entire documents or paragraphs, further enhancing the machine's ability to understand and analyze text.

Word2Vec

Word2Vec uses neural networks to learn distributed vector representations of words. It typically uses two approaches: (i) Continuous Bag of Words (CBOW) and (ii) Skip-gram.

For the Skip-gram model, the model learns embeddings for words based on context. The objective is to predict surrounding words given a target word. For a given target word w_t, the model tries to predict the context words $w_1, w_2, \ldots, w_m$. The equation for Skip-gram is:

$$J(\theta) = -\sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} log(p(w_{t+j}|w_t)) \tag{2.4}$$

Where w_t is the target word, w_{t+j} are the context words, T is the total number of words in the training corpus, m is the size of the context window, and $p(w_{t+j}|w_t)$ is the probability of the context word given the target word.

In Continuous Bag of Words (CBOW), the model learns to combine the context words and their embeddings to predict the target word.The objective is to predict the target word given the surrounding context words within a fixed-size window. The equation for CBOW is:

$$J(\theta) = -\sum_{t=1}^{T} log(p(w_t|w_{t-m}, \ldots, w_{t+m})) \tag{2.5}$$

Where w_t is the target word, $w_{t-m}, \ldots, w_{t+m}$ is the context of the target word, which includes $2m$ words before and after w_t, T is the total number of words in the training corpus, m is the size of the context window. In both cases, θ represents the parameters (weights) of the neural network.

The context is typically represented by averaging the word embeddings of the context words, and the target word is predicted using a softmax function over the entire vocabulary. This approach differs from Skip-gram, where the task is to predict surrounding context words given a target word.

Doc2Vec

Doc2Vec extends Word2Vec to represent entire documents as vectors. It learns a vector representation for each document in addition to the word vectors. There are two main models for Doc2Vec: (i) Distributed Memory (DM) and (ii) Distributed Bag of Words (DBOW).

For Distributed Memory (DM), the model learns a vector for a document d and a vector for each word in the context. The objective is to predict the context words given the target word and the document vector. The equation for DM is:

$$J(\theta) = -\sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} log(p(w_{t+j}|w_t, d)) \tag{2.6}$$

Where w_t is the target word, w_{t+j} are the context words, d is the document vector, T is the total number of words in the training corpus, m is the size of the context window, and $p(w_{t+j}|w_t, d)$ is the probability of the context word given the target word and the document vector.

In Distributed Bag of Words (DBOW), the model skips the word vectors and directly learns a fixed vector for each document, which is used to predict the surrounding words within a certain window. The equation for DBOW is:

$$J(\theta) = -\sum_{d=1}^{N} \sum_{i=1}^{T} log(p(w_i, d)) \tag{2.7}$$

Where N is the total number of documents in the corpus, T is the size of the context window, $p(w_i, d)$ is the probability of the word w_i in the context given the document d. In both cases, θ represents the parameters (weights) of the neural network.

2.2.4 Deep learning and transformers

Despite significant advancements, there was still room for improvement. Transformers, a new model architecture, transformed NLP by utilizing self-attention mechanisms to capture relationships between words, regardless of their distance. This innovation led to powerful models like BERT (DEVLIN et al., 2018) and GPT (RADFORD; NARASIMHAN, 2018), which improved contextual understanding and the ability to generate coherent, human-like text.

GPT (Generative Pre-trained Transformer)

GPT is a causal language model that uses the autoregressive approach, meaning it predicts the next token given the previous tokens in the sequence. GPT uses only the decoder part of the Transformer architecture and performs left-to-right prediction.

Autoregressive Language Modeling (ALM): Given a sequence of tokens $w_1, w_2, \ldots, w_t$, the model's goal is to predict the next token w_{t+1} given the context (i.e., all previous tokens). The equation of ALM is:

$$J(\theta) = -\sum_{t=1}^{T} log(p(w_{t+1}|w_1, w_2, \ldots, w_t)) \tag{2.8}$$

The prediction is based on the previous tokens' context, and the objective is to maximize the likelihood of predicting the correct next word in the sequence. Where T is the total number of tokens in the sequence and $p(w_{t+1}|w_1, w_2, \ldots, w_t)$ is computed using the transformer model, and θ represents the parameters (weights) of the neural network.

2.3 FINAL CONSIDERATIONS

This chapter presents, describes, and summarizes the main methodologies, pipelines, and models commonly applied in text processing using natural language processing. From machine learning algorithms and linear models to deep learning models, to describe learning types used for textual information as input to obtain relevant characteristics. In addition, we complement the text-processing and evaluation concepts by providing and detailing the most used methods such as Bag of Words, TF-IDF, and Doc2Vec.

3 RELATED WORKS

In this chapter, we provide an overview of the related literature and background relevant to the main themes of our work.

3.1 BLACKHAT FORUMS

Blackhat forums comprise unstructured posts. All posts include their content, author, and subject. Blackhat forums provide a way for researchers as well as badly-intentioned users to trade knowledge about hacking. These forums supply information ranging from beginner hacking skills to functional hacking tools that anyone can easily get access, sometimes for free. The so-called *script-kiddies*, i.e., home users with limited computing skills, for instance, can leverage those tools to initiate cyberattacks. One of our goals is to distinguish between research activity that poses a potential threat against exploitation in the wild, wherein criminals chat about threats. Despite the fact that blackhat forums provide a rich source of information about vulnerabilities, curating and using data from those forums is a daunting task. Among the challenges, we focus on the following.

1. **Timeliness.** Due to their public, democratic, and distributed nature, blackhat forums are candidates to be the first places where vulnerabilities are studied and discussed. Hackers typically find blackhat forums to be fertile ground to discuss exploits and weaponization strategies. However, early filtering and detection of activity about a given vulnerability are non-trivial. Although one can effectively find, in retrospect, information at the blackhat forums about well-known vulnerabilities, e.g., after they have been curated by NIST, detecting relevant activity at the blackhat forums before systems are compromised poses significant challenges.

2. **Report confidence and impact.** The early detection of activity about vulnerabilities is challenging, partially due to a tradeoff between timeliness and report confidence. Early messages about a given vulnerability typically discuss trials to develop proof-of-concept (PoC) weapons, which may turn to be ineffective, being challenging to determine their impact on vulnerability risk.

3. **Matching posts against vulnerabilities.** Posts at blackhat forums are unstructured. It is non-trivial to match those posts to recently reported vulnerabilities, for which little information may be available. Although some posts explicitly cite the vulnerability to which they refer to, others may only cite the targeted products or vendors.

In summary, for blackhat forums to provide useful and timely data about vulnerabilities, it is imperative to overcome the challenges associated with assessing the quality of the information provided in the posts, as well as matching those against vulnerabilities. To that goal, in this work, we employ an automated text mining approach to explore the broad spectrum of data available in the blackhat forums and extract relevant events early in the lifecycle of vulnerabilities.

3.1.1 Why blackhat forums for vulnerabilities lifecycle analysis?

The information gathered from blackhat forums and TI feeds can serve as *ground truth* for models that predict exploitability, such as EPSS and Expected Exploitability (SUCIU et al., 2022; JACOBS; ROMANOSKY et al., 2021). These platforms can also improve the *matching between vulnerabilities and exploits*, which is often non-trivial, by identifying CVEs that are not yet matched to corresponding exploits. Additionally, blackhat forums and TI feeds can be used to *find new exploits* before they are reported in public databases such as ExploitDB. The presence of a CVE citation in a blackhat forum can also motivate an *increase in the CVSS temporal score*, as it may indicate increased risk associated with the vulnerability. Finally, blackhat forums can be used to automatically identify *resource development activity*, i.e., activity where adversaries establish infrastructure, accounts, or capabilities to support their goals. By analyzing data from blackhat forums, one can gain insights into the development of these resources, identifying new attack methods and vulnerabilities.[1]

3.2 NLP AND THREAT INTELLIGENCE (TI)

The use of Natural Language Processing (NLP) for the analysis of hacker forums has been considered in a number of previous works (RAHMAN et al., 2021; PASTRANA; HUTCHINGS et al., 2019; PASTRANA; THOMAS et al., 2018; MORENO-VERA et al., 2023). In this work, we complement such a body of literature by focusing on discussions about software vulnerabilities within CrimeBB forums, which has been previously considered for the analysis of eWhoring (PASTRANA; HUTCHINGS et al., 2019) and other cybercrimes (PASTRANA; THOMAS et al., 2018; DEGUARA et al., 2022). Our research aims to analyze discussions about software vulnerabilities within CrimeBB forums. Threat Miner (DEGUARA et al., 2022) is a system to identify threats based on hacker forums, classifying notifications or reports as "good" if they represent a cyber threat that can be linked to a known CVE. Other NLP applications are sentiment classification based on the words used in threads and posts, such as identifying aggressive language (CAINES et

[1] See Mitre Att&ck Develop Capabilities https://attack.mitre.org/matrices/enterprise/

al., 2018a), intention and content type (CAINES et al., 2018b), and search for specific ideologies content such as misogyny, anti-semitism and racism (CHUA; WILSON, 2023),

3.3 EXPLOITATION OF VULNERABILITIES IN THE WILD

The analysis of the exploitation of vulnerabilities in the wild poses significant challenges. While attackers actively target vulnerabilities to compromise systems, often sharing techniques and information on online forums, researchers must understand these exploits to build effective defense strategies (LIANG et al., 2018). Manual methods, such as reverse engineering and fuzzing, are particularly useful in uncovering new vulnerabilities and weaknesses, helping to strengthen defenses against threats, whether discovered in controlled environments or exposed on blackhat forums (SUTTON; GREENE; AMINI, 2007).

Collaboration and information sharing play a crucial role in combating vulnerabilities. Platforms and databases, such as the Common Vulnerabilities and Exposures (CVE) system and the National Vulnerability Database (NVD), provide standardized information about vulnerabilities, including their severity and available patches (CHEN et al., 2016). Furthermore, coordinated disclosure practices, such as responsible disclosure and bug bounty programs, facilitate the reporting and fixing of vulnerabilities (EDKRANTZ; TRUVÉ; SAID, 2015). Other applications rely on finding shodan activities (BADA; PETE, 2020) or highlight hacker activities and data breaches (FANG et al., 2019). In addition, it is possible to study the social interactions between users and their criminal/non-criminal behavior (MAN; SIU; HUTCHINGS, 2023; PASTRANA et al., 2018), criminal and non-criminal trades, transactions, buys, and sells (BUTLER, 2020; ALLODI, 2017; SIU; COLLIER; HUTCHINGS, 2021), and analyze the maturity of CVE, CVSS, and EPSS software shared (SUN et al., 2021; MORENO-VERA, 2023; MORENO-VERA et al., 2023).

3.4 FINAL CONSIDERATIONS

In this chapter, we have thoroughly reviewed the body of work relevant to our study, focusing on three primary areas: blackhat forums, natural language processing (NLP) and threat intelligence, and the exploitation of vulnerabilities in the wild. CrimeBB, a dataset provided by the Cambridge Cybercrime Centre (CCC), has emerged as the most commonly used source for such analyses, offering extensive data on cybercriminal discussions. Through this comprehensive review, we identified several applications derived from textual analysis, including sentiment analysis, detection of aggressive language, trade and transaction tracking, identification of criminal activities, detection of racism, and detailed CVE analysis, among others.

This review has guided the definition of our own research path, which is centered on classifying and understanding discussions within threads and posts in underground forums. Unlike prior studies that often emphasize broad cybercrime themes, our approach focuses specifically on the vulnerabilities discussed and the associated risks of CVEs referenced within comments and user interactions. By concentrating on CVE citations, our research aims to provide deeper insights into emerging threats, emphasizing the potential risk each vulnerability poses as perceived by cybercriminals. This novel focus on granular vulnerability analysis enables a more targeted approach to understanding underground forum discussions, with implications for enhancing threat intelligence and cybersecurity practices.

4 EXPLORATORY ANALISYS OF THE *CRIMEBB* DATASET

In this chapter, we present our methodology and detail the steps involved, we also present the findings of our exploratory analysis, including statistics and insights obtained from the CrimeBB dataset and the PostCog extension.

4.1 DATASET DESCRIPTION

In this section, we provide a comprehensive overview of the datasets utilized in this study, detailing their sources, characteristics, and relevance to our research objectives. We discuss each dataset's structure, including the type and volume of data, as well as any preprocessing steps applied to ensure consistency and suitability for our analysis.

4.1.1 National Vulnerability Database (NVD)

The National Vulnerability Database (NVD) [1] is a comprehensive repository of information about software vulnerabilities and security issues. It is maintained by the National Institute of Standards and Technology (NIST) in the United States. The NVD dataset provides detailed information about known vulnerabilities in various software products, including operating systems, applications, libraries, and hardware.

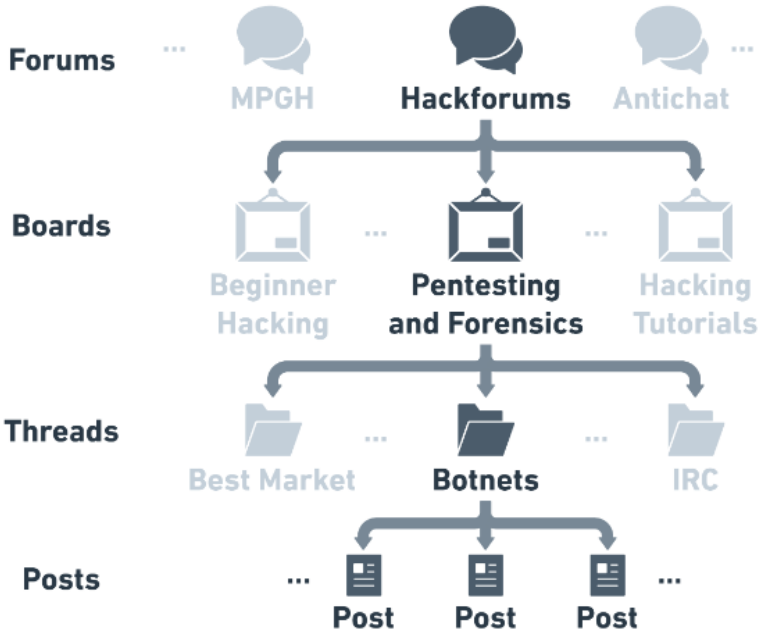
4.1.2 CrimeBB and PostCog

Cambridge Cybercrime Center (CCC) makes available 38 underground forums collected in the CrimeBB dataset (PASTRANA; THOMAS et al., 2018). The PostCog framework (PETE et al., 2022) extends CrimeBB providing a web application to navigate through CrimeBB data, to find keywords and to label posts following a crowd sourcing approach.

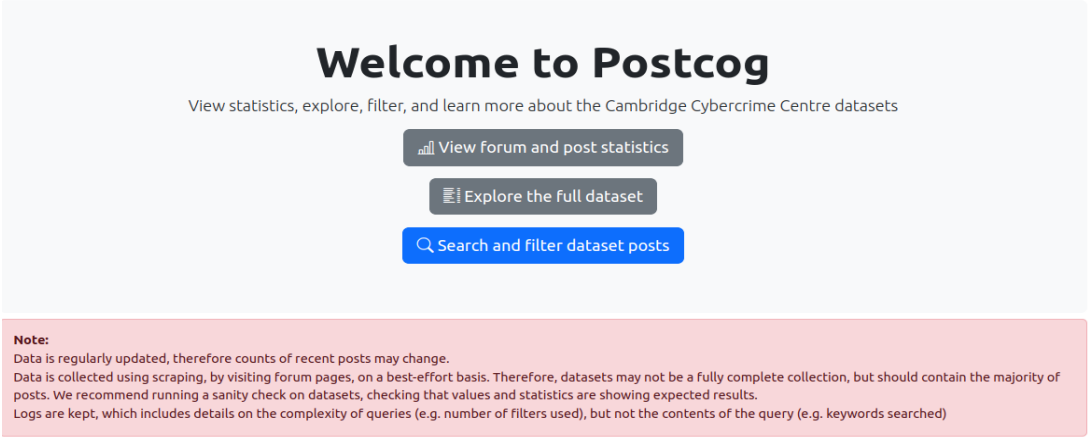
CrimeBB. Most posts in online forums consist primarily of plain text, though they may also incorporate additional elements such as images, videos, or attachments. By using the CrimeBot, CCC identifies and annotates various elements within posts, including images, video (via iframes), snippets of source code, external website links, internal thread links, and attachments (PASTRANA; THOMAS et al., 2018).

The CrimeBB dataset comprises hierarchical data based on websites, containing around 45Gb of textual information. The structure of underground forums is composed

[1] https://nvd.nist.gov/



(a)



(b)

Figure 4 – (a) CrimeBB dataset, showing the hierarchical composition of websites, boards, threads, and posts. (b) PostCog framework, we can navigate through crimeBB data using this framework.

of forums or "websites", followed by boards related to "topics", threads corresponding to "questions", and posts corresponding to "answers" (see Figure 4 (a)). As of August 28, 2024, CrimeBB had 6,739,073 users interacting on 37 websites, 10,600,580 discussion threads, and 117,365,492 posts (see Table 1).

PostCog. PostCog (PETE et al., 2022) is a web application providing curated data from CrimeBB underground forums. Figure 4 (b) shows the Welcome page of the PostCog framework, we are able to use this data set due to an agreement and an NDA contract assigned. Postcog is in its second phase of development. The initial prototype was

constructed using NodeJS, ExpressJS, and PostgreSQL. Following insights derived from tests with users, the technology stack was expanded to integrate ReactJS and ElasticSearch. Recently, PostCog allows users to search for words by forum, subforum, date, and NLP tags. Results can be exported as CSV files for external applications to analyze.

Table 1 – CrimeBB forums statistics extracted from PostCog. Forums are divided into boards, and boards are divided into threads. Each thread contains a number of posts. Data was downloaded on August 28, 2024.

Forum	**#Boards**	**#Threads**	**#Posts**	**First post**	**Recent post**
Hack Forums	212	4,301,893	42,686,891	2007-01-27	2024-05-24
Zismo	39	546,832	12,194,525	2010-05-26	2024-05-04
MPGH	770	918,439	12,193,797	2005-12-26	2024-05-26
Blackhatworld	112	1,017,226	12,132,290	2005-10-31	2024-05-20
Nulled	169	687,522	9,546,230	2013-04-02	2024-05-16
lolzteam	292	577,642	6,196,005	2013-03-10	2019-09-01
Cracked	163	419,517	3,911,032	2018-04-03	2024-04-17
OGUsers	58	244,766	3,608,306	1990-01-01	2019-04-09
UnKnoWnCheaTs	248	182,667	2,837,509	2002-11-02	2024-05-24
Antichat	80	254,810	2,642,161	2002-05-29	2024-03-15
V3rmillion	40	456,262	2,459,519	2016-02-02	2019-11-11
Raidforums	88	114,450	1,231,126	2015-03-20	2022-02-20
Elhacker	53	212,081	987,039	2002-08-21	2024-05-28
Probiv	168	123,023	909,007	2014-11-05	2024-04-25
Breached	72	34,412	737,922	2022-03-16	2023-03-19
Hackers Armies	53	42,548	468,880	2009-06-01	2024-04-01
Forum Team	201	44,404	433,901	2017-10-31	2024-03-26
BreachForums	76	28,800	331,357	2023-05-12	2024-05-14
Indetectables	72	32,274	328,539	2006-02-20	2024-05-19
XSS Forum	49	48,718	310,796	2004-11-13	2023-04-27
Dread	446	75,122	294,596	2018-02-15	2020-01-09
Runion	19	16,867	240,632	2012-01-11	2020-01-05
Offensive Community	71	119,251	161,492	2012-06-30	2018-12-11
Underc0de	73	27,054	95,723	2010-02-10	2024-05-26
The Hub	62	11,286	88,753	2014-01-09	2019-08-09
Ifud	65	11,827	72,851	2012-05-10	2022-12-19
PirateBay Forum	33	11,526	60,678	2013-10-23	2020-12-03
OnniForums	27	3,542	45,094	2023-02-08	2024-05-24
Torum	11	4,346	28,485	2017-05-25	2019-08-07
Safe Sky Hacks	50	12,963	27,018	2013-03-28	2019-01-23
Kernelmode	11	3,606	26,815	2010-03-11	2019-11-29
Freehacks	228	5,106	26,471	2013-07-27	2023-04-23
Deutschland im Deep Web	43	4,075	20,185	2018-11-22	2020-06-04
GreySec	28	2,232	11,925	2015-06-10	2022-01-04
Garage for Hackers	47	2,329	8,710	2010-07-06	2018-10-13
Stresser Forums	17	708	7,069	2017-04-09	2018-04-09
Envoy Forum	93	454	2,163	2019-07-06	2019-08-09
Total	4,339	10,600,580	117,365,492		

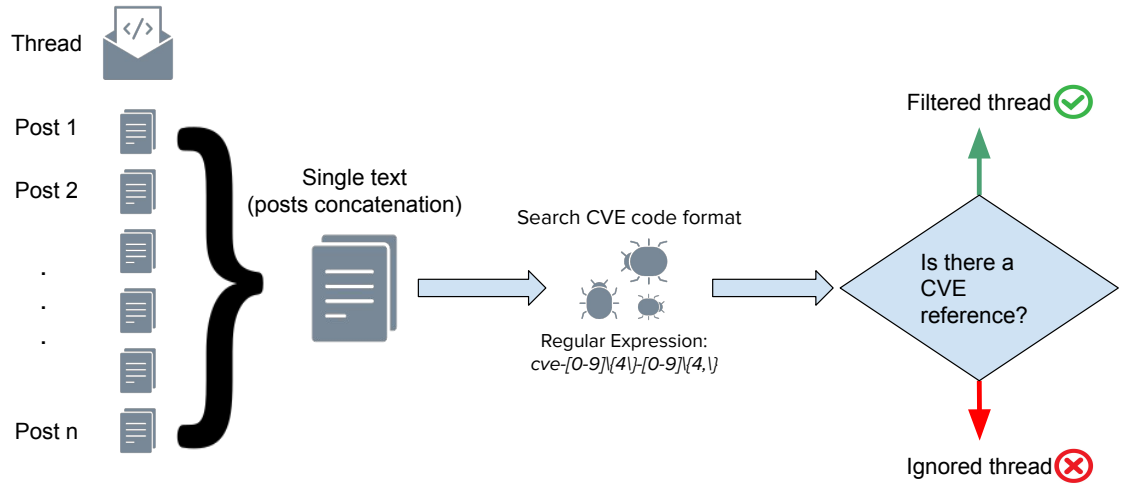


Figure 5 – Post concatenation by thread: If at least one post cites a CVE code, we take all others posts from the same thread as one text sample. Otherwise, the complete thread is ignored and excluded from the dataset. This is why we don't use all labeled threads.

4.2 UNDERGROUND FORUMS ANALYSIS

To produce the dataset, we consider the steps described in Figure 1. First, we use the PostCog framework to search and download relevant posts, threads, and boards from CrimeBB. We also obtained the post-creation date, username, and three labels assigned by PostCog based on the post-content such as intention, post type, and crime type (discussed). We divided this process into three sub-steps: (i) threads processing; (ii) labeling target classes; (iii) text processing; and (iv) tokenization and feature extraction

4.2.1 Thread processing

We employ a filtering process to identify all posts referencing at least one Common Vulnerabilities and Exposures (CVE) code. We concatenate and merge all posts with their corresponding parent thread. Then, by using case-insensitive regular expression `cve-[0-9]{4}-[0-9]{4,}` (slightly more specific than `cve(-id)?(?i)` used in (ALLODI, 2017) and refined in (MORENO-VERA et al., 2023)). Figure 5 presents the thread processing process. We search and filter posts referring to vulnerabilities by their CVE identifiers, ignoring threads-posts that do not cite any CVE reference. Across all CrimeBB forums, we found about 15,645 posts citing 2,218 unique CVEs (11,420 CVEs citations in total) under 6,496 discussion threads. Then, we concatenate each of those posts contained in a thread, along with the thread title.

4.2.2 Labeling target

We consider three types of labeling targets: (i) PostCog labeling; (ii) Expert manual labeling; and (iii) ChatGPT labeling.

PostCog labels

We obtain three sets of labels from PostCog, these labels are only available for the HackForums threads-posts:

1. **Post type**, which refers to the content of the post, and is retrieved from (CAINES et al., 2018b);
2. **Intention**, which refers to user intention (e.g., negative, positive, or neutral) and is retrieved from (CAINES et al., 2018b);
3. **Crime type**, which refers to the criminal activity identified within the thread, and is obtained from (SIU; COLLIER; HUTCHINGS, 2021).

These labels are currently available only for the HackForum site; therefore, most of our analysis focuses on it.

Expert annotations in HackForum

We leverage the manual labeling by experts reported in (MORENO-VERA et al., 2023).[2] We asked experts to label a subset of 1,037 threads in the CrimeBB dataset using the HackForum website. The experts used the following code book to manually label the threads:[3]

1. **Exploitation:** (i) mention a well-known hacker group; (ii) reference cryptocurrencies and keywords like bitcoin, exploitation, and attack (in the context of attacks in the wild); (iii) discuss approaches to make exploits fully undetectable; (iv) involve markets of exploits.
2. **Proof-of-Concept (PoC):** (i) contain keywords such as PoC, tutorial, or guide (in the context of producing tools in a lab or controlled environment); (ii) provide a tutorial on building a PoC; or (iii) discuss vulnerabilities without using exploits in the wild.

[2] In (MORENO-VERA et al., 2023) we already had all expert labels considered in this work, but we used only three of them (exploitation, PoC and weaponization).

[3] Accounting for slang and abbreviations that are typical in those communities is left as a subject for future work.

Crime type labels	
Labels	Samples
Not criminal	2,307
Bots/Malware	604
Sql Injection	208
Credentials	41
VPN/proxy	34
DDoS/booting	12
Spam/marketing	7
CurrencyXchange	4
Identity fraud	2
eWhoring	1

(a)

Intention labels	
Labels	Samples
Neutral	2,184
Other	494
Positive	197
Gratitude	170
Aggression	53
Negative	37
PrivateMessage	30
Moderate	28
Vouch	27

(b)

Post type labels	
Labels	Samples
InfoRequest	912
Comment	909
Other	494
OfferX	490
Exchange	137
RequestX	137
Tutorial	76
Social	65

(c)

Expert annotations labels	
Labels	Samples
Weaponization	397
PoC	242
Others	190
Exploitation	102
Warning	55
Help	41
Scam	10

(d)

Table 2 – We present the 3,220 threads and set of labels obtained from the PostCog framework: (a) Crime type labels correspond to cybercrime types discussed in a post, (b) Intention labels refer to the intention expressed within a post, and (c) Post type labels correspond to the content of the post. In addition, in (d) we present the 1,037 HackForum threads annotated by experts.

3. **Weaponization:** (i) contain keywords like vulnerability and exploit (in the context of weaponization); (ii) discuss the availability of fully functional or highly mature exploits, providing references or source code.

4. **Warning:** contains advice or warning about the detection (e.g., by some company) of exploitation of some vulnerability.

5. **Help:** contains keywords such as help or coding referring to users asking help to exploit some vulnerability or execute some code of general purpose.

6. **Scam:** contain information about transactions, or selling an exploit or vulnerability that does not work or has already been published.

7. **Others:** Other discussion not related to any above.

In Table 2, we summarize the number of samples for each class for each set of labels analyzed in this work. Note the imbalance of samples for each set of labels: “not criminal”, “infoRequest” and “neutral” are the most representative classes for crime type, post type, and intention, respectively.

Crime type labels	
Labels	Samples
Not criminal	2,307
Cybercrime activities	875
Cybercrime support services	38

(a)

Intention labels	
Labels	Samples
Sentiment	2,418
Other	494
Expression	280
Intensity	28

(b)

Post type labels	
Labels	Samples
Communication/Interaction	1050
Requests	1049
Other	494
Offer/exchanges	627

(c)

Expert labels	
Labels	Samples
Malicious activity	509
Informal	297
Others	190
Support and assistance	41

(d)

Table 3 – We present the new labels assigned by ChatGPT, (a) crime type labels were grouped into three new labels, (b) intention labels were grouped into four new labels, (c) post type labels were grouped into four new labels, and (d) expert annotations in HackForums were grouped into four new labels.

ChatGPT labeling

We ask ChatGPT [4] to label our threads using prompts given the context and content of each thread. We ask each set of labels to summarize: (1) post type, (2) crime type, (3) intention, and (4) expert annotations in HackForums. One of our aims was to reduce the number of classes and to have a representative set of up to four labels for each of the above four dimensions. We use ChatGPT 3.5-turbo and perform a few-shot learning (AHMED; DEVANBU, 2022), by giving five samples of input and output for each class of each set of labels (crime type, post type, intention, and expert annotations). Below we give the prompt template example used for crime type labeling:

```
I have a list of possible labels {not criminal, bots/malware, ...} related to cyber criminal activites.

I want to perform two tasks: the first one is to group the list of possible labels into a smaller list of lab

The second task is to set a new label using the new smaller list of labels given an input raw {text} and thei

Based on the following samples { (text 1, label 1, new label 1), \dots, (text 5, label 5, new label 5)}, I wo

Please return in a list of tuples: [ ( "input", {text},   "new label", {new label}), ...]
```

Consequently, we execute this prompt to obtain a new group of labels from each set of labels and the new labels for each sample. In Table 3, we present the new set of labels, classes, and the number of samples assigned by the ChatGPT prompt. In all cases, we significantly reduce the number of classes, while preserving a high imbalance in crime type and intention labels. From this, ChatGPT learns to classify threads using the following criteria:

[4] OpenAI ChatGPT: https://www.openai.com/chatgpt

1. **Crime type:** recategorized into “not criminal” defined as activities that are not considered criminal, “cybercrime activities” defined as illegal activities related to cybercrime, and “cybercrime support services” defined as services that facilitate cybercrime activities.

2. **Post type:** recategorized into “communication/interaction” defined as forms of interaction or content type, “requests” defined as seeking information or services, “offer/exchanges” defined as providing services or trading, and “others”.

3. **Intention:** recategorized into “sentiment” defined as emotions or attitudes, “expression” defined as ways of communicating or expressing oneself, “intensity” defined as levels of strength or forcefulness, and “others”.

4. **Expert annotations in HackForum:** recategorized into “malicious activity” defined as content including about weaponization, exploitation, or scam, “informational” defined as content including PoC, alerts, tutorials, or warnings, “support and assistance” defined as content asking help, and “others”.

4.2.3 Text feature extraction

We will divide our preprocessing into two steps: (i) nlp preprocessing, (ii) language evaluation, and (iii) feature extraction.

NLP pre-processing

We implement a text preprocessor that helps identify and keep relevant words to consider. We implement a library to preprocess text and evaluate language in order to facilitate our dataset preparation. We must be careful to select which word should be filtered. To do this, we filter the following characters:

- **Stopwords:** These are the words in any language which does not add much meaning to a sentence. Some samples of stopwords are pronouns, adverbs, articles, etc.

- **Punctuations:** These are symbols that you add to a text to show the divisions between different parts of it, such as Periods, commas, semicolons, question marks, apostrophes, and parentheses.

- **Special Characters:** These Are the symbols used in writing, typing, etc., that represent something other than a letter (outside the 26 letters used in US English) or number, such as §, à, é, î, œ, ü, ñ, etc.

- **Emojis:** These are a form of pictorial language used to express an idea, also called “digital images". These symbols denote an emotion or an action. It can be an image or textual composed symbol such as ":)", ":C", ":-)", etc.

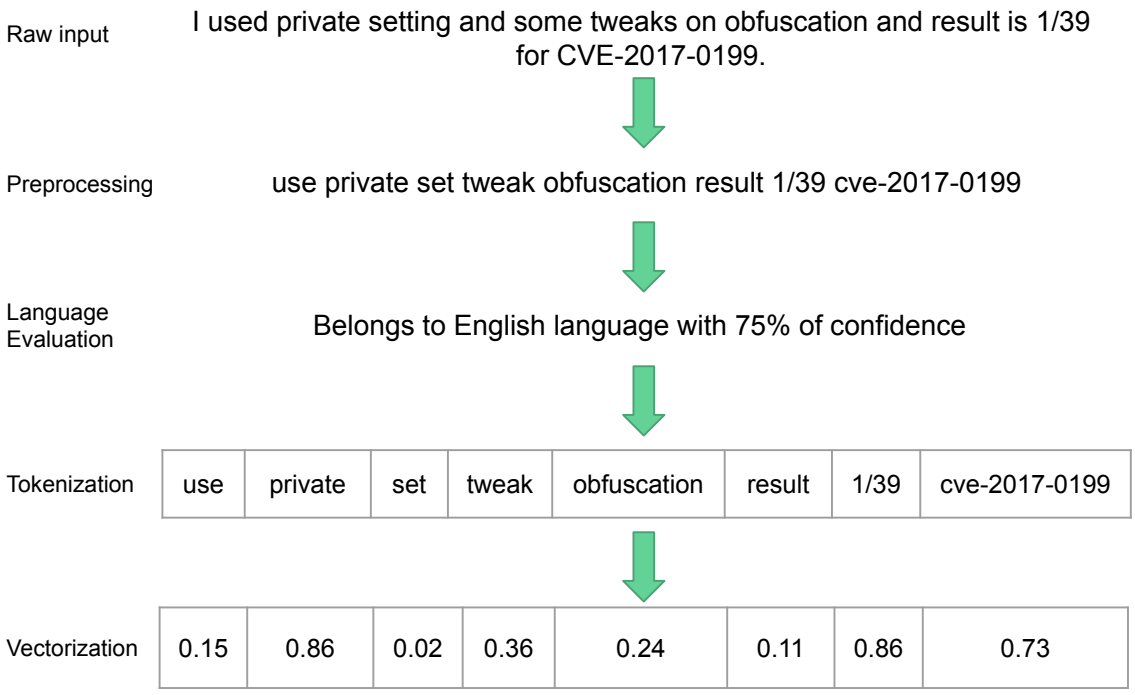


Figure 6 – Text preprocessing pipeline, we show all steps from raw input until vectorization. Note that we only keep and lemmatize words to their basic root form but not all words. This post was taken from the HackForums website.

After filtering out these characters from the raw input, we proceed to identify the language to which the input text belongs. Subsequently, we choose to convert all words using lemmatization instead of stemming. This choice is justified because, in text classification, lemmatization facilitates the creation of high-quality, contextually accurate, and semantically meaningful feature representations. This, in turn, contributes to improved classification accuracy, reduced noise, and enhanced generalization across various types of text data.

Language evaluation

We define the **Indicator Language Function** (ILF) denoted by $\mathbb{I}_{ilf}$. In Equation 4.1 we define our ILF, taking two parameters the word w and the language L to eval:

$$\mathbb{I}_{ILF}(w, L) = \begin{cases} 1, & \text{if } w \in L \\ 0, & \text{Otherwise} \end{cases} \tag{4.1}$$

Moreover, we define the **Language Ratio Function** (LRF) for a set of words and languages. This function allows us to identify and calculate what percentage of words within a phrase, paragraph, or text, in general, belongs to a specific language L. In Equation 4.2 we define our LRF, taking two parameters the text t with n words and the language L

to eval:

$$\text{Ratio}_{LRF}(t, L) = \frac{1}{\text{n}} \times \sum_{\substack{i=1 \\ \text{w}_i \in t}}^{n} \mathbb{I}_{ILF}(w_i, L) \tag{4.2}$$

Finally, the **Determine Language Function** (DLF) is defined using the previously calculated Language Recognition Factor (LRF). This function determines the most probable language to which the input text belongs. In Equation 4.3, we define our DLF, taking a text t as a parameter. This text is evaluated in a set of languages $\mathbb{L}$ (English and Russian language by default):

$$\text{Language}_{DLF}(t) = \max_{\forall L \in \mathbb{L}} \text{Ratio}_{LRF}(t, L) \tag{4.3}$$

This step helps us to identify the main language within a thread (we only focus on the English language. Otherwise, we filter it). After preprocessing the thread-posts concatenations to identity languages, we identified 3,220 thread-posts in English, 1,977 thread-posts in Russian, 1,223 thread-posts in Spanish, 21 thread-posts in Portuguese, and 4 thread-posts in Deutsch. After this step, we proceed to perform text embedding, converting text into vectors.

Tokenization & Feature Extraction

In this step, we perform the feature extraction from textual information. After pre-processing all textual information and performing the tokenization of words, we will use text encoding-based methods such as Bag-Of-Words (BoW) (HARRIS, 1954), Term Frequency - Inverse Document Frequency (TF-IDF)(SALTON; BUCKLEY, 1988), and Doc2VEc (MIKOLOV et al., 2013). Figure 6 shows all the text processing pipeline followed to perform feature extraction.

4.3 EMPIRICAL FINDINGS

In this section, we report empirical findings from CrimeBB forums, including users activity, risks, exploit prices, delays, and CVSS and EPSS scores from NVD.

4.3.1 Users activity

Figure 7 (a) shows how activity varied across users. In particular, Trillium is the most active user, typically announcing novelties in the Trillium Security MultiSploit Tool and FUD artifacts. Trillium cited more than 175 CVEs. The other users appearing in threads citing CVEs span a less diverse set of CVEs. In particular, it is challenging to relate real users to virtual accounts, as real users may create multiple virtual accounts,

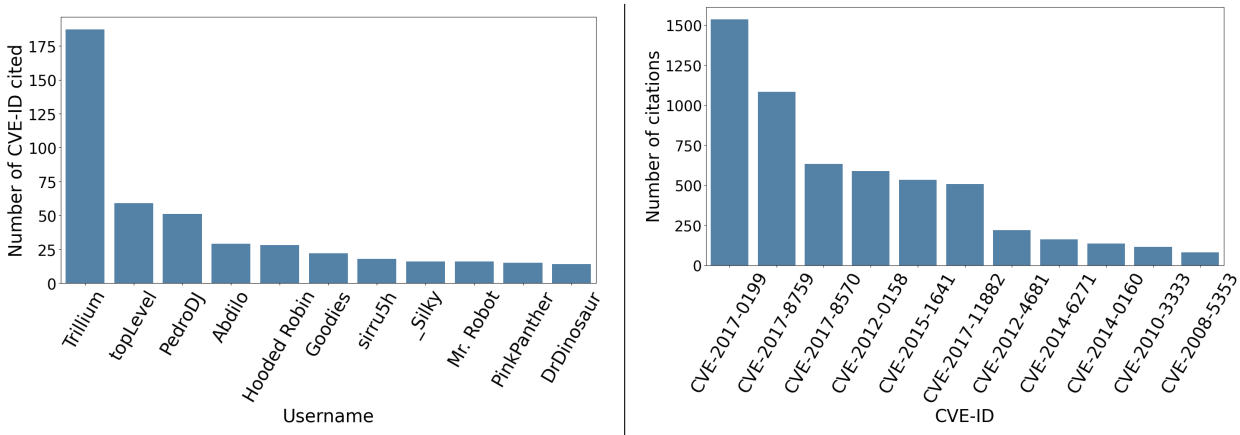


Figure 7 – Statistics: (a) Users activity: most users that cite CVEs cite less than 50 unique CVEs. A notable exception is Trillium. (b) The popularity of CVEs on Hackforums.

e.g., to increase privacy or voting power. Indeed, the forums at CrimeBB do not use reputation for admission control purposes. This is in stark contrast to the Russian market, which requires explicit admission by market administrators, conditioned on the user being active in related communities, incentivising users to keep their accounts active over longer periods of time (ALLODI, 2017). It is worth noting that in this work we focus exclusively on threads referring to vulnerabilities. For a broader analysis of the reputation of users across CrimeBB forums, we refer the reader to (PARACHA; ARSHAD; KHAN, 2023).

4.3.2 Risks

Figure 7 (b) shows the most cited CVEs at Hackforums. It indicates that the number of citations sharply drops over the top 10 vulnerabilities. Among the top vulnerabilities, we have CVE-2017-0199, CVE-2017-8570, CVE-2017-8759, and CVE-2017-11882. For those four vulnerabilities, we find exploits on GitHub, as cited by NVD. Those four vulnerabilities, together with CVE-2010-3333 and CVE-2012-0158, are related to Microsoft products. A number of the vulnerabilities cited at CrimeBB are also cited in other relevant sources. This is the case, for instance, of CVE-2012-0158, known as Microsoft MSCOMCTL.OCX Remote Code Execution Vulnerability, and CVE-2015-1641, which are found at CISA's Known Exploited Vulnerabilities (KEV) Catalog. [5] For other vulnerabilities, such as CVE-2008-5353, CVE-2012-4681, CVE-2010-0094, and CVE-2011-3544, there are exploits available at ExploitDB or Metasploit toolkit [6]. Those vulnerabilities are related to the Java Runtime Environment. Other notable vulnerabilities include CVE-2014-0160 and CVE-2014-6271, which correspond to Heartbleed and Shellshock.

[5] See also https://www.cisa.gov/binding-operational-directive-22-01
[6] https://www.exploit-db.com/exploits/16293

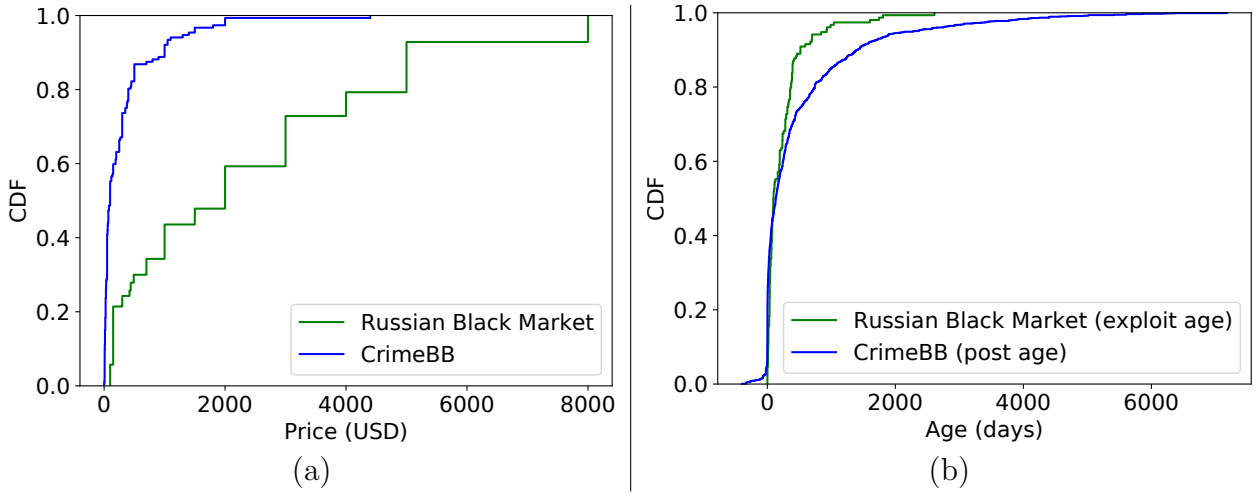


Figure 8 – Russian Market vs CrimeBB: (a) CDF of hacking tools prices: prices at CrimeBB are relatively low compared to the Russian market – some prices correspond to subscriptions, and others to repackaging and FUD. Price statistics: CrimeBB (Min: 1, Median: 100, Max: 4400), Russian market (Min: 100, Median: 2000, Max: 8000). (b) CDF of the difference in days between CrimeBB citation and NVD publish date. Negative values correspond to citations to CVEs that occurred before NVD published the corresponding vulnerability. Age statistics: CrimeBB (Min: -396, Median: 132, Max: 7181), Russian market (Min: 1, Median: 95,5, Max: 2610)

4.3.3 Prices

Moving on, we will now examine the prices of artifacts referenced by users in CrimeBB forums. Figure 8 (a) displays the CDF of prices in dollars, and for comparative purposes, we also present the CDF of prices of exploits reported at the Russian market analyzed in (ALLODI, 2017). The minimum, median, and maximum values in CrimeBB forums were 1 USD, 100 USD, and 4,400 USD, respectively, while the corresponding values in the Russian market were 100, 2,000, and 8,000 USD, respectively. Additionally, we observed that over 80% of the references to hacking tools corresponded to prices under 1,000 USD. The higher prices in the Russian market compared to CrimeBB forums can be attributed to the fact that the Russian market requires explicit admission by market administrators. As a result, users in the Russian market tend to discuss more mature artifacts, which are consequently more expensive.

On the other hand, in CrimeBB forums, it was noticed that users often propose the repackaging of existing exploits, such as under new FUD versions, or offer subscriptions to websites that tend to be cheaper than exploits (VALEROS; GARCIA, 2020). Despite the differences in prices, some similarities were also observed. The maximum prices did not exceed USD 8,000 on both platforms, the majority of prices were below USD 2,000, and roughly 20% of the prices were close to USD 100. These figures suggest that participating in these forums can be financially rewarding, with rewards aligned with most bug bounty programs that offer up to USD 3,000 for a critical bug. However, these amounts are still

far from the million-dollar bug bounties that have been reported in the literature.[7]

4.3.4 Delays

Next, we consider the delays between the publication of vulnerabilities and posts appearing at the forums. For CrimeBB forums, we compute the postage as the difference between the day of the post and the day on which the corresponding vulnerability was published at NVD, `PostAge = PostPubDate - CVEPubDate`. Similarly, the exploit age reported by (ALLODI, 2017) is the difference between the day on which an exploit was published at the Russian market and the day on which the corresponding vulnerability was published at NVD, `ExplAge = ExplPubDate - CVEPubDate`. Figure 8 (b) shows the CDF of `PostAge` and `ExplAge`, for CrimeBB forums and the Russian market, respectively.

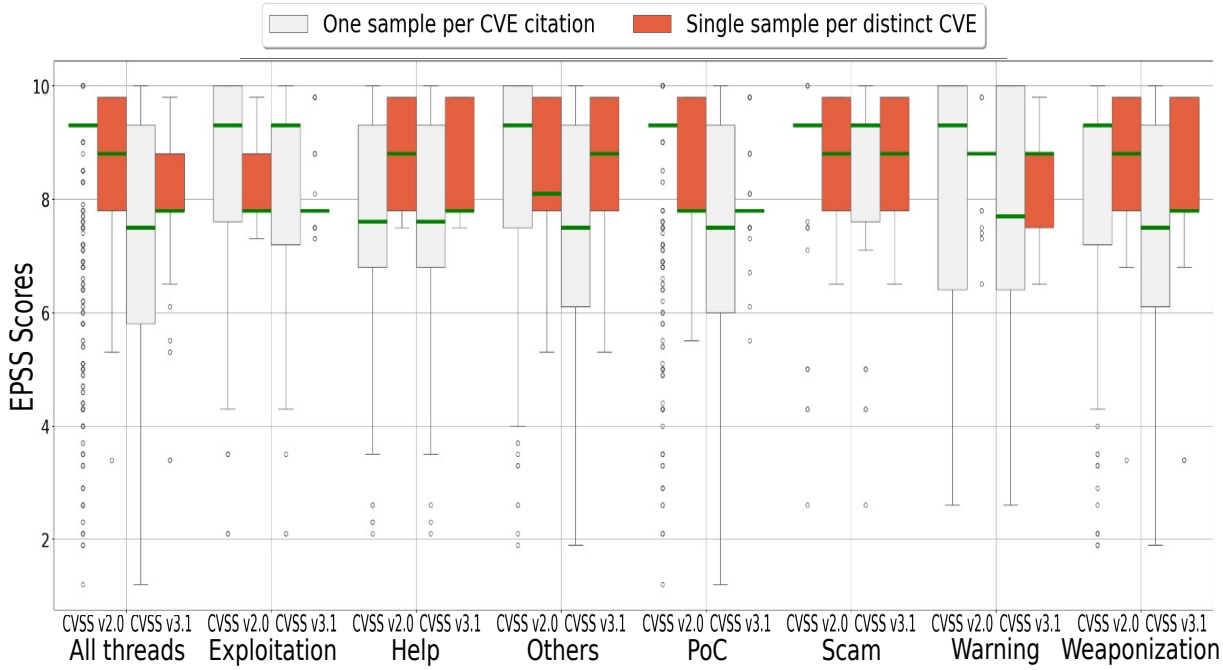
Note that more than 50% of exploits discussed in CrimeBB are about vulnerabilities that were disclosed over the previous 69 days before being cited at CrimeBB. Considering that 50% of Industrial Control Systems (ICS) are not patched 60 days after vulnerability disclosure(WANG et al., 2017), the use of blackhat forums is imperative to estimate risks associated with vulnerabilities. Among the similarities between CrimeBB forums and the Russian black market, we observe that roughly 60% of the activity occurs very close to CVE publish date. Discussion tends to phase out for virtually all vulnerabilities 6 years after they are released.

We observe that in the Russian market, we have only positive age values, whereas under CrimeBB we have a small fraction of negative values. This is explained by the different nature of the two forums, as discussed in the previous section: whereas CrimeBB also counts with messages querying about vulnerabilities and discussing strategies to produce proof-of-concept weapons, the Russian market contains mostly discussion of mature exploits to be sold at higher values, and typically being released only after the CVE has already been published at NVD. Fully functional or high-maturity exploits are rarely produced before the vulnerability publish date, i.e., `ExplPubDate` is larger than `CVEPubDate`. Discussions about vulnerabilities, however, can initiate before they are released, as CVE identifiers are announced to the public before they are published at NVD.

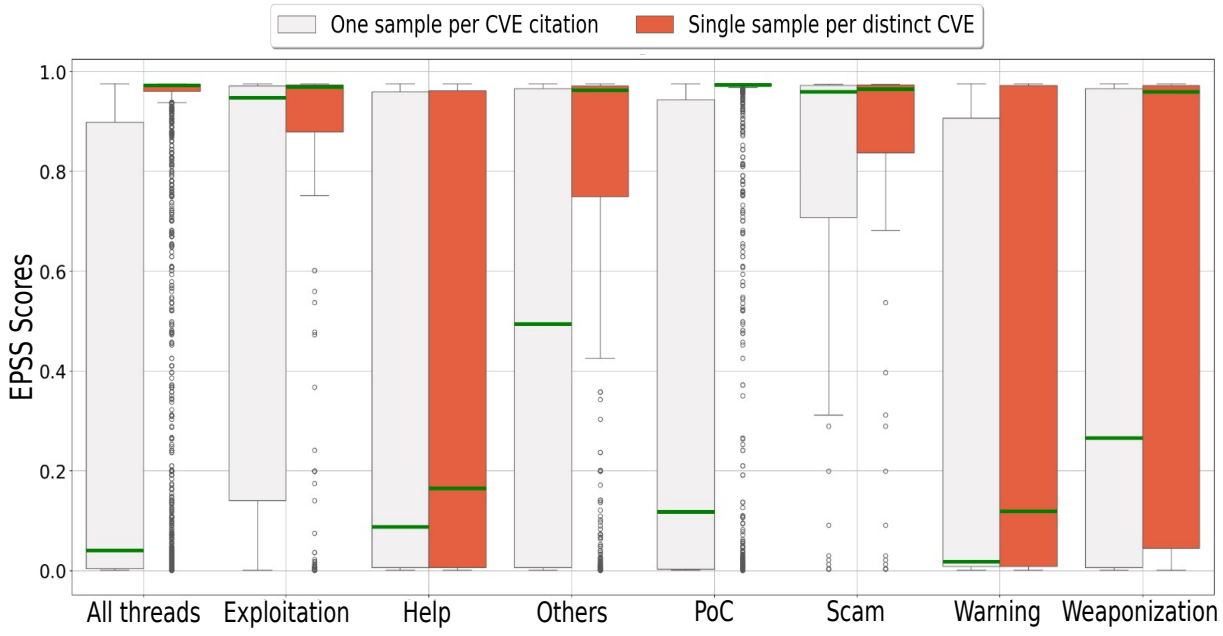
NVD data

We use data from NVD to determine properties of the considered vulnerabilities, such as severity level (CVSS [8]). CVSS ranges between 0 and 10, and values above 8 correspond to high risk. In particular, NVD provides a brief description of each vulnerability, together with its publish date, products affected, and external resources. Note that CVSS

[7] https://portswigger.net/daily-swig/million-dollar-bug-bounties-the-rise-of-record-breaking-payouts
[8] https://nvd.nist.gov/vuln-metrics/cvss



(a) CVSS



(b) EPSS

Figure 9 – Distribution of CVSS and EPSS scores across different classes annotated by experts. Note that 91% of posts refer to CVEs whose CVSS score is higher than the mean CVSS across all NVD CVEs (not shown in the figure).

accounts for a temporal subscore to capture the maturity level of exploits. The level ranges between PoC, fully functional, and high maturity. In particular, CVSS does not account for exploitation in the wild. Expected Exploitability, in contrast, exclusively targets fully functional and high maturity exploits, referred to, in this work, as weaponization, and EPSS [9] targets exploitation. In this work, we generalize the scope of CVSS and EPSS, by

[9] https://www.first.org/epss/

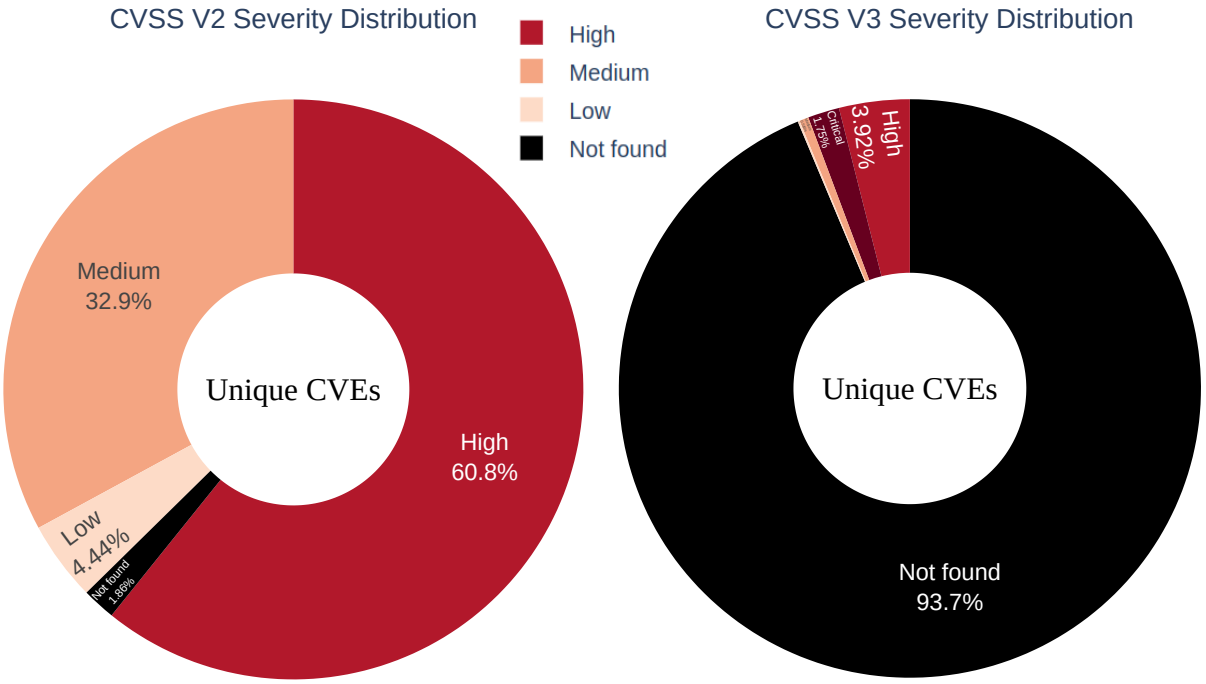


Figure 10 – Common Vulnerability Scoring System (CVSS) severity level, we compare the version 2 and 3.1 of CVSS scores. We note that about 93.7% CVSS v3.1 scores are not available.

considering the three maturity levels in a unified framework.

Across all vulnerabilities, Figure 9 shows the distribution of CVSS and EPSS scores conditioned on the thread class and across all threads. Grey boxplots consider one sample per CVE citation (with repetitions), and red boxplots consider a single sample per distinct CVE (without repetitions). Figure 9 (a) displays the distribution of CVSS scores, for CVSS versions 2.0 and 3.1 (the latter is available for a subset of CVEs). Across all categories, the median CVSS values exceed 7.0. Accounting for CVSS 2.0, the figure shows that exploitation usually corresponds to higher CVSS values when compared against PoC and weaponization.

Additionally, still accounting for CVSS 2.0 the figure reveals that CVEs with higher CVSS values and severity are the most frequently cited (about 60.8% of CVEs are high risk). This is evidenced by the fact that the medians of the red boxplots are all above 9.0, indicating that the risk scores are magnified by repeated citations (see Figure 10). A similar trend is observed in Figure 9 (b) for EPSS. EPSS was released in 2021, and we used its latest version on August 28, 2024, which represents the probability of exploitation in the wild in the next 30 days. Despite the gap between post-citations and the release of EPSS scores, we observe that EPSS scores are able to capture high risks for the vulnerabilities present in our dataset.

(a) All (b) Exploitation

(c) Help (d) Others

(e) Poc (f) Scam

(g) warning (h) Weaponization

Figure 11 – Word clouds showing the most frequent keywords appearing across posts agrupated by expert annotations: (a) all posts; (b) PoC; (c) weaponization and (d) exploitation.

4.3.5 Content wordclouds

Figure 11 shows the word clouds obtained from CrimeBB posts grouped by classes and across all posts. The presence of the word "multisploit" in the PoC category indicates that Trillium Security MultiSploit Tool is a popular tool used by hackers. Furthermore, we note that how the language used by hackers in the threads is instrumental to distinguish

exploitation in the wild from the rest of the posts. The presence of the word “thanks”, in the exploitation cloud, refers to the behavior of users that confirm that a given exploitation worked.

Note that the term “clean” is also prevalent across posts about exploitation in the wild. Indeed, it appears in the context of FUD exploits. Users typically indicate that they were able to run the exploit and that it was not detected by any antivirus. As an example, in a single post, we found 35 occurrences of the word “clean” when referring to a “silent” exploit to CVE-2011-3544. The exploit authors share their MD5 hash and report a detection ratio of 0/35. To provide evidence of the miss-detection, the authors present a sample output: `AVG Free: Clean; ArcaVir: Clean; Avast 5: Clean`. The list continues with a total of 35 antiviruses.

4.4 FINAL CONSIDERATIONS

In this Chapter, we describe the CrimeBB and PostCog framework used to download and obtain our data. We also describe the data set preparation process, starting with text pre-processing, text corpus, and labeling techniques. We ask domain experts to annotate about 1,037 posts from Hackforums underground kindly. In addition, we use ChatGPT model to classify thread based on their content and re-label the annotations from experts and the Postcog labels. We also describe our findings using NVD dataset to obtain CVSS and EPSS. We also analyse and compare prices mentioned in discussion threads, delays in responses or trades, risks, and users activities.

5 DISTINGUISHING POTENTIAL AGAINST IMMINENT THREATS

In this chapter, we present the discussion and results obtained from our experiments.

5.1 ENVIRONMENT PREPARATION

In our study, we compared the three encoding techniques and found that BoW is more interpretable, while TF-IDF and doc2vec yield higher accuracy.

5.1.1 **Feature extraction**

For BoW and TF-IDF, we consider the following four parameters: 1) the top 30,000 most frequently occurring words, 2) that appear at least 5 times, and 3) in at least 90% of the posts in the corpus are considered for analysis. In addition, we consider 4) n-grams, ranging from one word to three words. Overall, these parameters define the criteria for selecting which words or groups of words are included in the analysis and how frequently they must appear in the corpus. For Doc2Vec, we encode posts into 5000-dimensional vectors. We use standard NLP pre-processing techniques, e.g., filtering English language stop words and punctuation from the posts.

5.1.2 **Model configurations and metrics**

We perform classification tasks using linear models such as Ridge Classifier (TIKHONOV, 1943), LASSO (TIBSHIRANI, 1996), logistic regression (HOSMER; LEMESHOW; STURDIVANT, 2005), linear SVC (CORTES; VAPNIK, 1995), and SVM (BOSER; GUYON; VAPNIK, 1992). We also add ensemble models such as Decision Tree (SPEY-BROECK, 2012) and RandomForests (BREIMAN, 2001). We use *Accuracy*, *Precision*, *Recall*, and *F1 score* to compare the performance of the models, we mainly rely on the values reported by *F1 score*. These metrics are calculated as:

Table 4 – Number of posts (threads) citing CVEs in the top 10 Hackforums boards, ranked by number of tagged posts

Board	Number of posts (threads) citing CVEs							
	Posts tagged as							
	PoC		Weapon		Exploit		All posts	
Pentesting and Forensics	271	(55)	210	(57)	11	(3)	557	(166)
Premium Tools and Programs	198	(1)	28	(3)	142	(4)	433	(20)
Website and Forum Hacking	93	(34)	139	(43)	16	(12)	333	(132)
Hacking Tools and Programs	10	(7)	57	(28)	174	(7)	260	(59)
Premium Sellers Section	–	–	81	(28)	89	(26)	210	(66)
Beginner Hacking	86	(43)	58	(47)	6	(6)	219	(143)
Botnets, IRC, and Zombies	24	(4)	85	(34)	22	(5)	160	(62)
Hacking Tutorials	58	(21)	8	(4)	3	(3)	74	(33)
Secondary Sellers Market	8	(4)	33	(21)	–	–	91	(40)
News and Happenings	9	(9)	11	(5)	1	(1)	75	(54)
Total, all boards	757	(244)	710	(397)	464	(102)	3,037	(1,162)

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{5.1}$$

$$Precision = \frac{T_P}{T_P + F_P} \tag{5.2}$$

$$Recall = \frac{T_P}{T_P + F_N} \tag{5.3}$$

$$F1_{score} = 2\frac{Precision * Recall}{Precision + Recall} \tag{5.4}$$

Where T_P means *True Positive*, T_N means *True Negative*, F_P means *False Positive* and F_N means *True Negative*. In addition, we started with an imbalanced dataset in all categories (see Table 2 and Table 3). To address this imbalance, we used the Random Over-Sampling heuristic (LEMAîTRE; NOGUEIRA; ARIDAS, 2017) to produce a balanced dataset. We split the dataset into 75% for training+validation and 25% for testing. For each experiment, we then performed a five StratifiedKFold cross-validation grid search to find the optimal hyperparameters, such as regularization parameter, learning rate, tree depth, the number of features to consider at each tree split, minimum samples required to split an internal node, and maximum node degree. All experiments were trained on GPU NVIDIA GeForce GTX 1070 with 8 Gb VRAM and Intel Core i5 Processor with 6Gb RAM.

Table 5 – Decision Tree (DT) and Random Forest (RF) performance.

	Text encoding	Target classes	Accuracy	Precision	Recall	F1
DT	BoW	PoC, Weaponization, Exploitation	0.71	0.71	0.72	0.70
DT	TF-IDF	PoC, Weaponization, Exploitation	0.73	0.73	0.74	0.72
DT	doc2vec	PoC, Weaponization, Exploitation	0.74	0.74	0.74	0.73
DT	BoW	Exploitation vs Non-exploitation	0.85	0.86	0.85	0.85
DT	TF-IDF	Exploitation vs Non-exploitation	0.91	0.91	0.91	0.91
DT	doc2vec	Exploitation vs Non-exploitation	**0.92**	**0.93**	**0.92**	**0.92**
DT	BoW	PoC vs Non-PoC	0.75	0.75	0.75	0.75
DT	TF-IDF	PoC vs Non-PoC	0.77	0.78	0.77	0.77
DT	doc2vec	PoC vs Non-PoC	0.70	0.71	0.70	0.70
DT	BoW	Weaponization vs Non-weapon.	0.68	0.68	0.68	0.68
DT	TF-IDF	Weaponization vs Non-weapon.	0.63	0.64	0.63	0.62
DT	doc2vec	Weaponization vs Non-weapon.	0.59	0.59	0.59	0.59
RF	BoW	PoC, Weaponization, Exploitation	0.85	0.84	0.85	0.84
RF	TF-IDF	PoC, Weaponization, Exploitation	0.86	0.87	0.86	0.86
RF	doc2vec	PoC, Weaponization, Exploitation	0.86	0.90	0.86	0.86
RF	BoW	Exploitation vs Non-exploitation	0.98	0.98	0.98	0.98
RF	TF-IDF	Exploitation vs Non-exploitation	0.98	0.98	0.98	0.98
RF	doc2vec	Exploitation vs Non-exploitation	**0.99**	**0.99**	**0.99**	**0.99**
RF	BoW	PoC vs Non-PoC	0.84	0.84	0.84	0.84
RF	TF-IDF	PoC vs Non-PoC	0.87	0.87	0.87	0.87
RF	doc2vec	PoC vs Non-PoC	0.88	0.90	0.88	0.87
RF	BoW	Weaponization vs Non-weapon.	0.67	0.67	0.67	0.67
RF	TF-IDF	Weaponization vs Non-weapon.	0.69	0.70	0.69	0.69
RF	doc2vec	Weaponization vs Non-weapon.	0.58	0.58	0.58	0.58

5.2 IDENTIFYING INFORMATION POINTS FROM ONLINE FORUMS

In this experiment, we focus on the 1,037 expert's annotation of HackForums. For experiments, we only use Proof-Of-Concept (PoC), Weaponization, and Exploitation labels.

5.2.1 **Dataset**

Table 4 shows the boards that contain most of the posts citing CVEs. In the top two boards, we find users selling and buying exploits, which indicates that discussions about vulnerabilities are generally about exploits already available on the market. Furthermore, Table 4 also shows the distribution of posts across different classes over the different boards at Hackforums. Note that in the board of pentesting, for instance, we find significant activity related to weaponization and exploitation. In contrast, few posts explicitly cite CVE identifiers in the board of hacking tutorials.

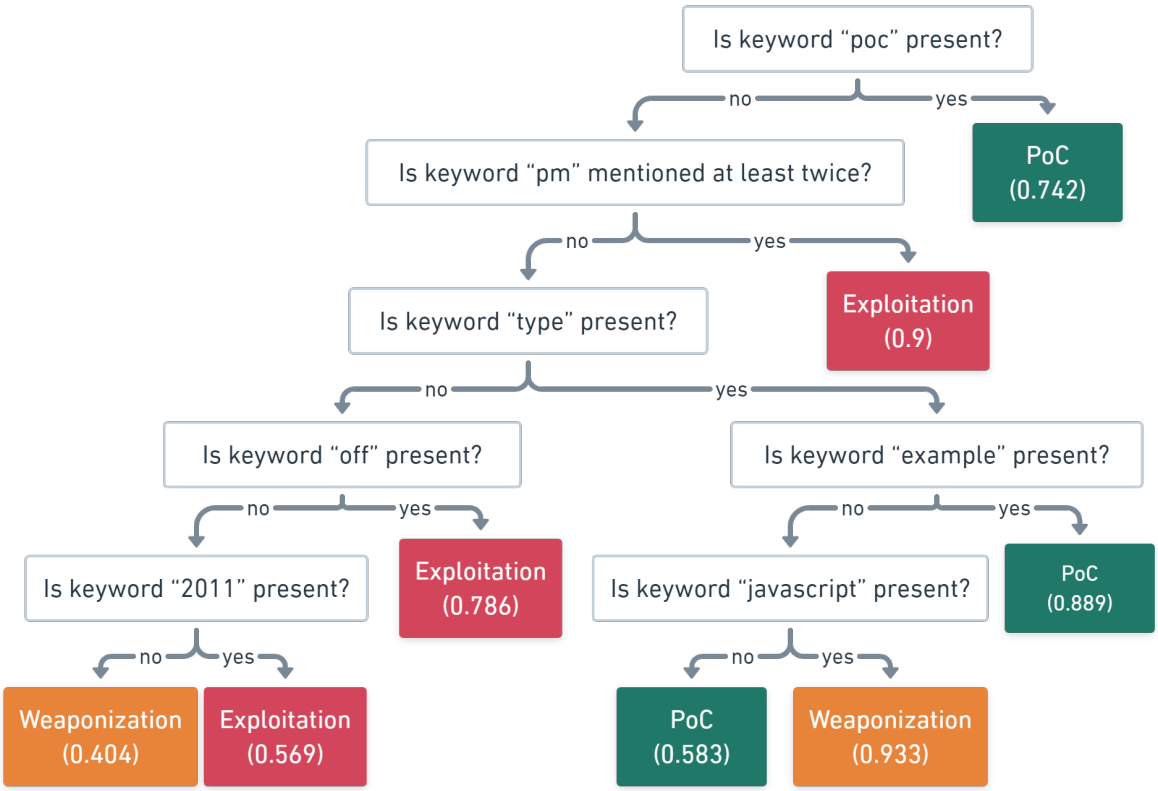


Figure 12 – Decision tree to classify PoC, weaponization, and exploitation.

5.2.2 Model interpretations

We report the performance of the considered classifiers. To that aim, we account for four metrics: accuracy, precision, recall, and F1. Table 5 shows the obtained results, considering the best set of hyperparameters for each configuration, as described above. In particular, our configurations vary as a function of the text encoding strategy and target classes. We observed increasing accuracy when switching from the simpler and more interpretable encoding (BoW) to the most complex but less interpretable one (doc2vec). Indeed, doc2vec outperforms BoW and TF-IDF except for the cases "PoC versus Non-PoC" and "Weaponization versus Non- Weaponization". Nonetheless, BoW is instrumental to produce the interpretable tree presented in Figure 12.

With respect to the target classes, we consider PoC vs Weaponization vs Exploitation and three additional one-against-all classifiers, in which each binary classifier separates members of a class from members of other classes. The best results were obtained when filtering exploitation in the wild from the rest of the threads, which is arguably the first step towards identifying relevant information at underground forums, as exploitation poses the most eminent risk. With respect to the classifier model, decision trees are simpler than random forests, producing less accurate predictions but being amenable to interpretation, as illustrated below.

Figure 12 illustrates the decision tree used to classify between PoC, weaponization, and exploitation (first line in Table 5). Each internal node in the tree contains a rule that splits the dataset, and each leaf indicates the most prominent class at that split and its frequency. Despite the fact that not all splitting rules that appear in Figure 12 are interpretable, we can already extract interesting insights from it. In the root of the tree, we find the rule with the highest splitting power, according to the Gini index criterion. Indeed, the root together with the leaf immediately below it indicate that if the thread contains the keyword "poc", with a 74.2% chance, it is actually a proof-of-concept. The following rule indicates that posts wherein users are concerned about privacy, i.e., containing the keyword "pm", which stands for "private message" in the black forum jargon, correspond to exploitation in the wild. Finally, we also observe that JavaScript is a common language used to produce exploits, e.g., that injects code through unverified input fields.

5.2.3 Conclusions of the experiment

In this experiment, we leveraged CrimeBB and machine learning methods to learn textual content and distinguish between: (1) potential threat (proof of concept), (2) eminent threat (weaponization), and (3) criminals chatting about a threat (exploitation in the wild). We found that the most cited CVEs are typically related to higher risks and that it is feasible to filter exploitation threads, with an accuracy above 99% automatically.

5.3 INFERRING TOPICS FROM HACKING FORUMS

In this experiment, we use topic modeling to analyze vulnerability exploitation in underground hacking forums. We apply Latent Dirichlet Allocation (LDA) (BLEI; NG; JORDAN, 2001) to infer topics, meaning that, based on the discussion thread content, we identify the topic being discussed.

5.3.1 Dataset

We employ the labels as topics: Proof of Concept (PoC), weaponization, exploitation, and "other" (including others, scam, warning, and help), labeled with corresponding IDs 1, 2, 3, and 4, respectively. The goal is to identify key themes related to exploit techniques, vulnerabilities, and potential targets, providing valuable insights into the landscape of vulnerability exploitation.

5.3.2 Topic modeling

We use the Gensim library (REHUREK; SOJKA, 2011) to perform topic modeling using the LDA algorithm. In Figure 13 (a) we show the Topic group projection by principal

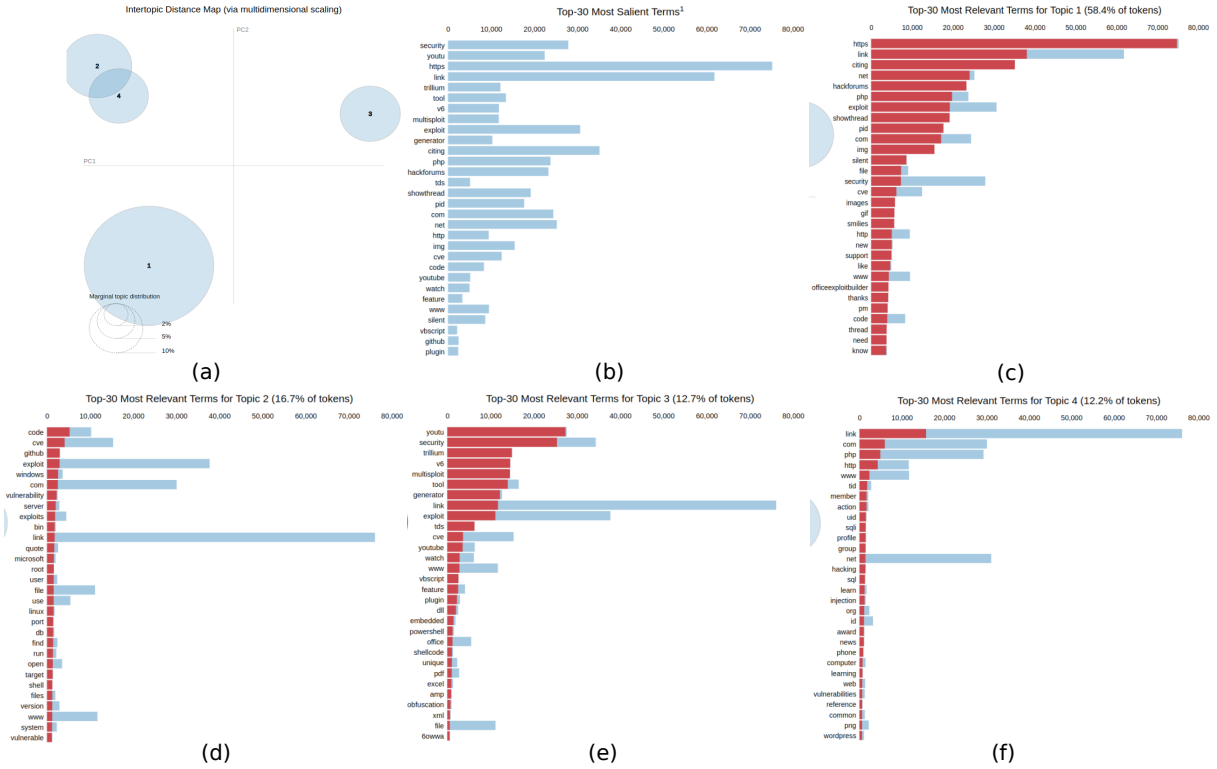


Figure 13 – Topic group projection by principal components. (a) The radius of each group determines the marginal topic distribution, (b) the top 30 most salient terms, (c) the top 30 most relevant terms for topic *PoC*, (d) the top 30 most relevant terms for topic *Weaponization*, (e) the top 30 most relevant terms for topic *Exploitation*, and (f) the top 30 most relevant terms for topic *Others*.

components, we note that topics *Weaponization* and *exploitation* have an intersection. This is an interesting result due to the proximity between a vulnerability from weaponization to exploitation. In Figure 13 (b) we show the 30 top words in all topics, we note that words such as “https”, “link”, “citing”, and “exploit” are the most relevant.

In Figure 13 (c) using only the 58.4 % of tokens the most relevant for the *PoC* topic are “https”, “link”, “php”, etc. We note that in general, those words are relevant for all topics except for “code” and “security”. In Figure 13 (d) we show the relevant words for the topic *weaponization* using only the 16.7 % of tokens are the words “code”, “cve”, “github”, etc. In Figure 13 (e) we show the relevant words for the topic *exploitation* using only the 12.7 % of tokens are the words “security”, “trillium” (the user who cites the highest quantity of CVE codes (MORENO-VERA et al., 2023)), “multisploit”, “tool”, “exploit”, etc.

In Figure 13 (f) we show the relevant words for the topic *others* using only the 12.2 % of tokens are the words “link”, “com”, “php”, “member”, “profile”, “learn”, etc. We note that for each topic we have some relevant words that let us understand the main discussion. In the intersection between the topics *Weaponization* and *exploitation*, we have some relevant words such as “code”, “cve”, “exploit”, “link”, and “vulnerability”.

Besides, in the topic *PoC* the most relevant word is "https", "link", "citing", "net", etc. This happens due to the nature of the discussion, sharing links, tutorials, code, etc. From this, we observe the topic modeling could infer the discussion theme within a thread. Furthermore, we know that in each thread can be discussed several themes but only one topic.

5.3.3 Conclusions of the experiment

In this experiment, we apply an unsupervised learning technique, topic modeling, to analyze the exploitation in the wild. We use experts annotations and re-categorize them into PoC, exploitation, weaponization, and others. We were able to identify separate clusters of words corresponding to each topic, we also identified relevant words such as "code", "cve", "exploit", "link", and "vulnerability" are commonly used in exploitation and weaponization discussions. We found that it is possible to identify emerging threats, helping security professionals, and researchers stay informed and prioritize their defense strategies accordingly.

5.4 UNVEILING INFORMATION POINTS FROM FORUMS

In this experiment, we use PostCog labels, expert annotations, and GPT classification labels.

5.4.1 Dataset

We compare and evaluate the performance of our models described using the following labeling:

- **PostCog Labels**: These labels were provided by PostCog domain experts and served as the baseline for our analysis (see Table 2 (a), (b), and (c)).
- **Expert Labels**: These labels were provided by domain experts using a code book reported in (MORENO-VERA et al., 2023) (see Table 2 (d)). We compare our new results against those ones, focusing on three classes: "poc", "weaponization", and "exploitation".
- **ChatGPT Labels**: We used ChatGPT to generate new labels, to explore the potential of AI-driven annotations and contrast the classification results using the new labels (see Table 3) against the PostCog labels.

HackForum contains 4 million threads, of which only approximately 3,200 are labeled with their crime type by at PostCog. In the case of post type and intent labels,

Table 6 – Random Forest (RF) performance summary for all experiments.

	Target classes	Accuracy	Precision	Recall	F1
Crime type	PostCog labels	**0.97**	0.97	0.99	0.98
	ChatGPT labels	0.95	0.98	0.94	0.96
	Previous work (SIU; COLLIER; HUTCHINGS, 2021)	0.89	0.9	0.89	0.89
Intention	PostCog labels	0.98	0.95	0.97	0.95
	ChatGPT labels	**0.99**	0.97	0.99	0.98
	Previous work (CAINES et al., 2018b)	–	0.78	0.49	0.61
Post type	PostCog labels	**0.81**	0.79	0.89	0.82
	ChatGPT labels	0.74	0.75	0.76	0.75
	Previous work (CAINES et al., 2018b)	–	0.91	0.78	0.84
Expert annotations	Expert labels	**0.96**	0.97	0.98	0.97
	ChatGPT labels	0.91	0.92	0.93	0.92
	Previous work (MORENO-VERA et al., 2023)	0.86	0.87	0.86	0.86

there are even fewer labels at PostCog (2700), while the remainder were cataloged as "other" (corresponding to no label). Therefore, one of our attempts while utilizing ChatGPT was to label more threads, to fill up missing data.

5.4.2 GPT labeler and classifiers

We report the performance of the considered classifiers. We report the four metrics described: accuracy, precision, recall, and F1 score. In general, RandomForest emerged as the model that performed the best, achieving the highest accuracy and robustness in our experiments. For all experiments using RandomForest, we achieved an accuracy higher than 74%-99%. Other methods produced accuracies lower than 60%. In addition, we observed that the results using the ChatGPT labeling do not always improve the baseline results, especially when aiming to predict post type and crime type. Table 6 reports the performance of random forests to infer labels provided by PostCog, ChatGPT and by our own expert annotations. We have a separate RandomForest to infer post type, crime type, intention, and expert annotations. We note that for crime type and intention, models trained using ChatGPT labels and PostCog labels produce similar performance. However, for post type and expert annotations, ChatGPT labels were harder to predict. In addition, we take results from our previous work (CAINES et al., 2018b; SIU; COLLIER; HUTCHINGS, 2021; MORENO-VERA et al., 2023), and compare them against our new results, indicating that our new results, irrespectively of the use of ChatGPT labels, correspond to better performance. Manual inspection of expert labels and ChatGPT labels suggests that ChatGPT is not always better than human expert annotations.

5.4.3 Model explanations

In order to explain the outcomes produced by RandomForests, we use the SHAP (SHapley Additive exPlanations), which enables the interpretation of text classifiers, offering clear and consistent insights into model predictions (LUNDBERG; LEE, 2017;

RIBEIRO; SINGH; GUESTRIN, 2016).

Crime Type

Figure 14 reports the classification results and SHAP-generated explanations for the PostCog crime type labels, which achieved a model accuracy of 97%, and the ChatGPT crime type labels, which achieved a model accuracy of 95%. In Figure 14 (a) we focus on PostCog labels. Note that we have eight classes, but classes with low number of samples have minimal relevance for the explanations generated by SHAP. The words "injection", "fud" (fully undetectable),[1] "hacked", and "open" present a high relevance for the classes "bots/malware", "not criminal", and "wrong access/sql injection".

The word "pm" is relevant when assessing the class "trading credential". "pm" means 'private message', and indicates that a private message will be used to share details about the post, due to privacy issues. Words such as "btc" (for Bitcoin) and "selling" are also relevant for "spam-related/marketing". In Figure 14(b), the explanations using ChatGPT labels focus on two classes: "not criminal" and "cybercrime activities". The words "server" and "prices" shows a relevance to the class "cybercrime support services". We also find that "php", "malicious", "scan", and "free" are words related to "cybercrime support services".

Intention

In Figure 15, we present the classification results and SHAP-generated explanations for the PostCog intention labels, corresponding to a model accuracy of 98%, and the ChatGPT intention labels, with a model accuracy of 99%. In Figure 15 (a), we show the relevance of different words for the PostCog intention labels. The word "thanks" has a high relevance for the classes "neutral" and "gratitude", and a minimal contribution to "positive". The word "vouch" is informative for the class with same name.

Figure 15 (b) shows that the word "thanks" is relevant to determine the intention classes as determined by ChatGPT. Indeed, "thanks" helps to differentiate between the "sentiment" and "expression" classes. We also identify words whose meaning does not represent something relevant, such as "com", "de", "la", "para", etc. This indicates that although the precision of the random forest was high, it is not always straightforward to find SHAP-based explanations for the classification results.

Post Type

Figure 16 presents the results of classification and SHAP-based explanations for the PostCog post type labels, with a model accuracy of 81%, and the ChatGPT post type

[1] A fully undetectable exploit is an exploit that is not detected by any of the known antivirus tools.

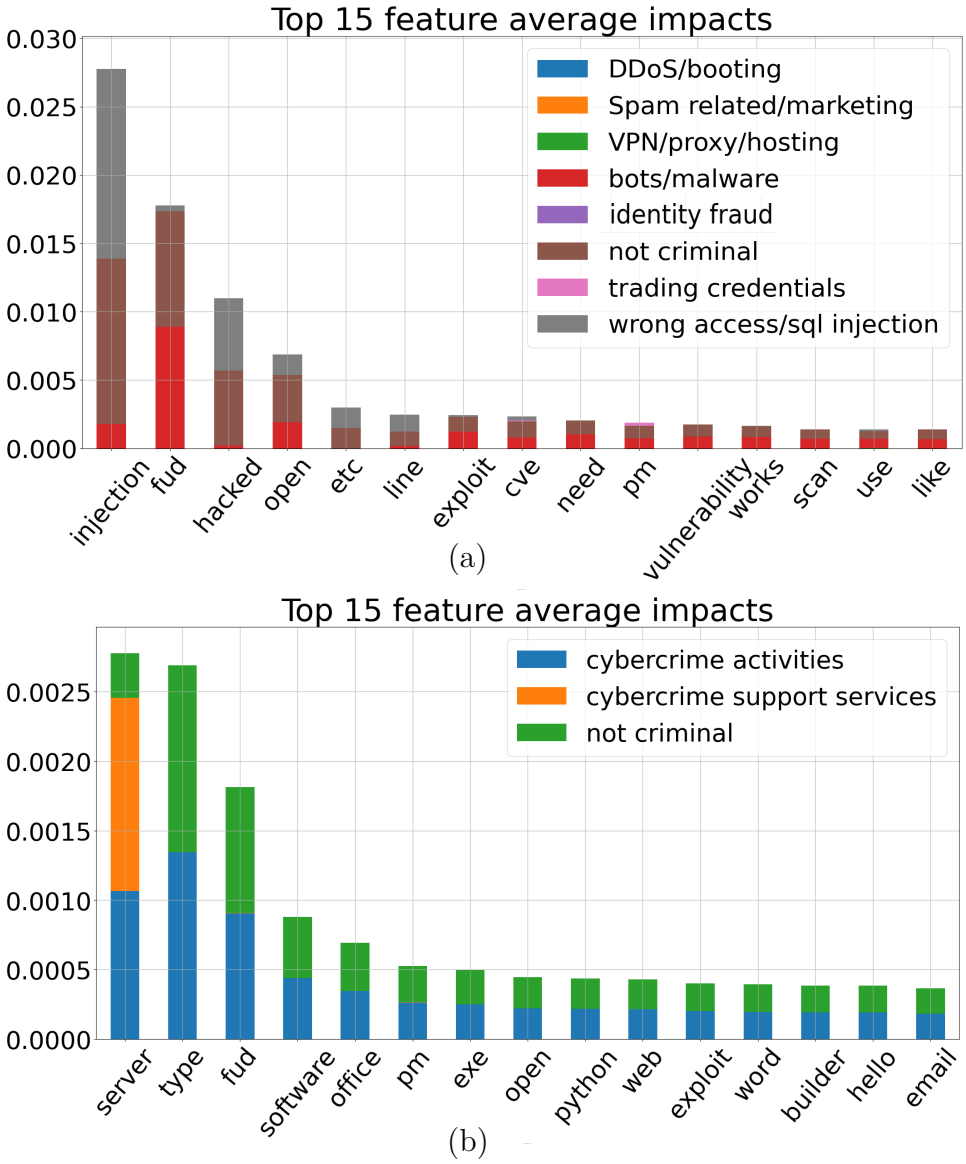
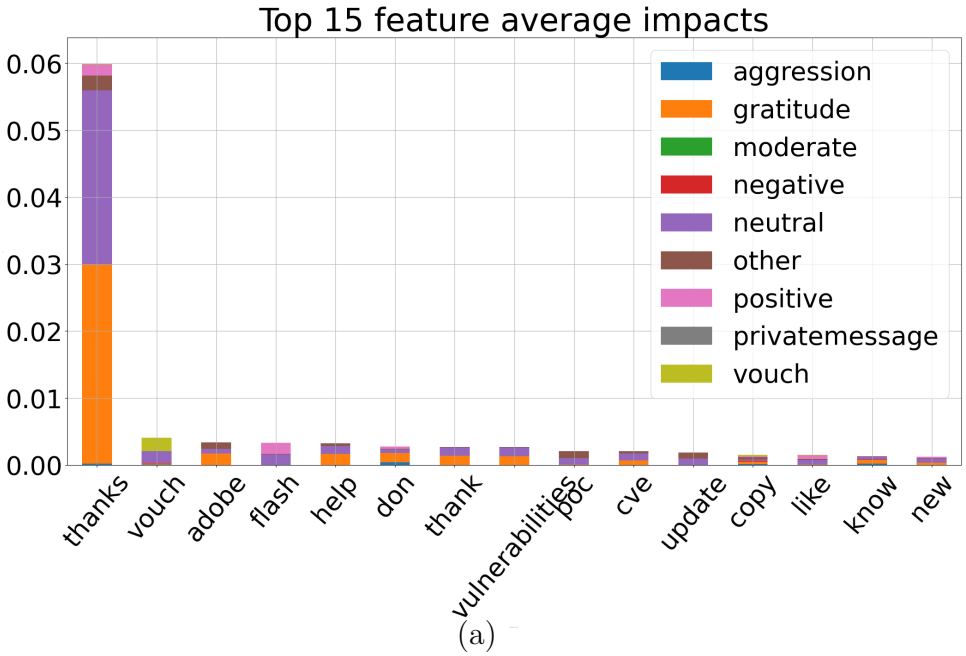


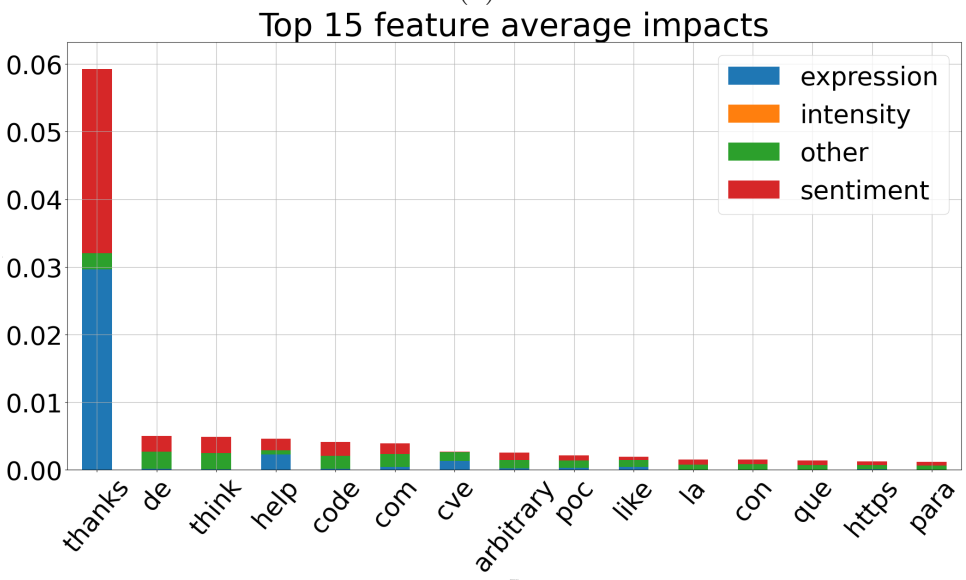
Figure 14 – The 15 most relevant features based on SHAP explanation values were determined from models trained with (a) the **PostCog crime type** labels and (b) the **ChatGPT crime type** labels.

labels, with a model accuracy of 74%. In Figure 16(a) we present the explanations for the model trained using PostCog post type labels, noting that the keyword “fud” is relevant to distinguish between infoRequest, comment, and offerX.

Other keywords, such as “poc”, “https”, and “found” are more relevant to identify classes such as comment and infoRequest. Figure 16(b) illustrates that SHAP-based explanations using the ChatGPT intention labels exhibit a better class balance importance for each word. For instance, words like “found” and “public” are more associated with requests, communication, and offers. Additionally, terms such as “www”, “poc”, and “exploit” can help distinguish whether the corresponding class is communication or offers.



(a)



(b)

Figure 15 – The 15 most relevant features based on SHAP explanation values were determined from models trained with (a) the **PostCog intention** labels and (b) the **ChatGPT intention** labels.

Expert Annotations

In Figure 17, we show the SHAP-based explanation of the random forest models trained using expert annotation labels (accuracy of 96%) and the corresponding ChatGPT-based labels (accuracy of 91%). In Figure 17(a) we note that the word "exploit" is very related to "weaponization", while the word "sorry" is related to the class "exploitation". We also find important words such as "pm" (private message), "fud", "issue", and "information" relevant to "poc" and "weaponization" classes.

In Figure 17 (b), the relevance of the words "fud", "exploit", "companies", "mal-

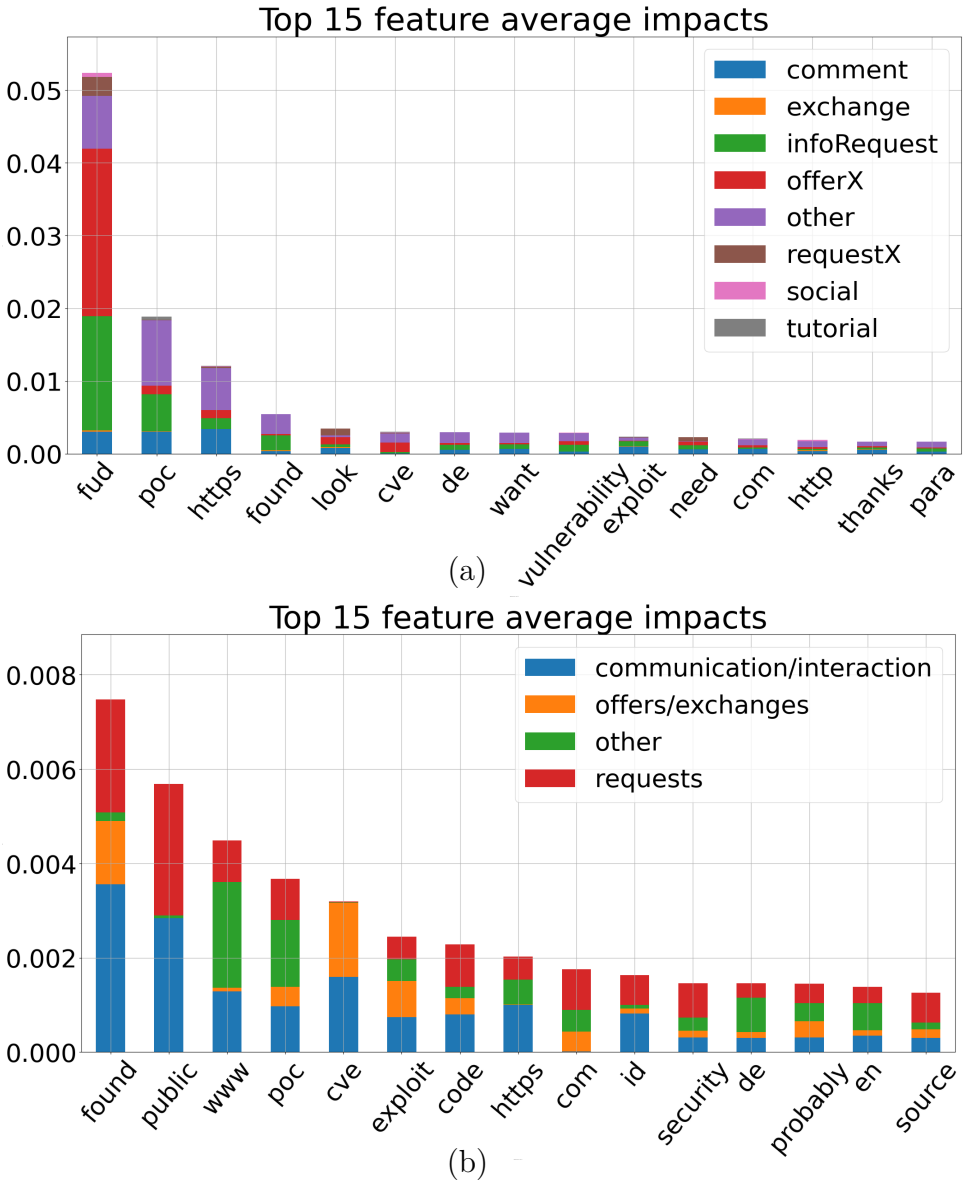
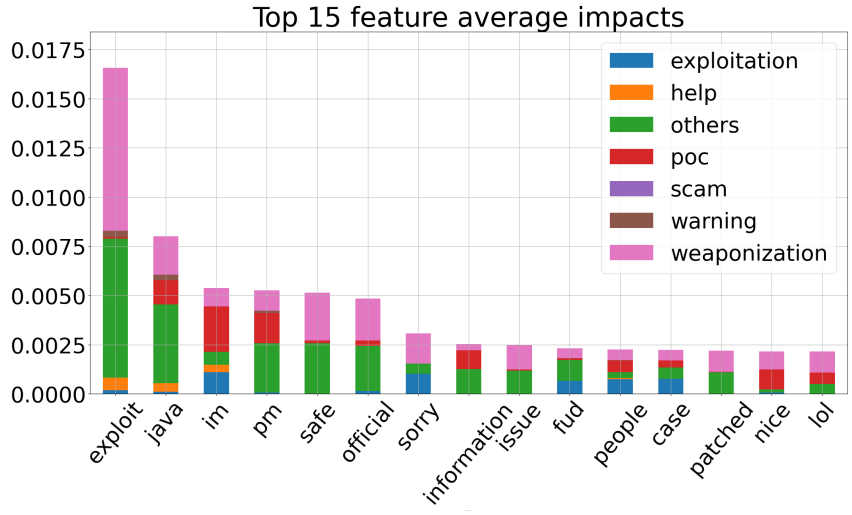
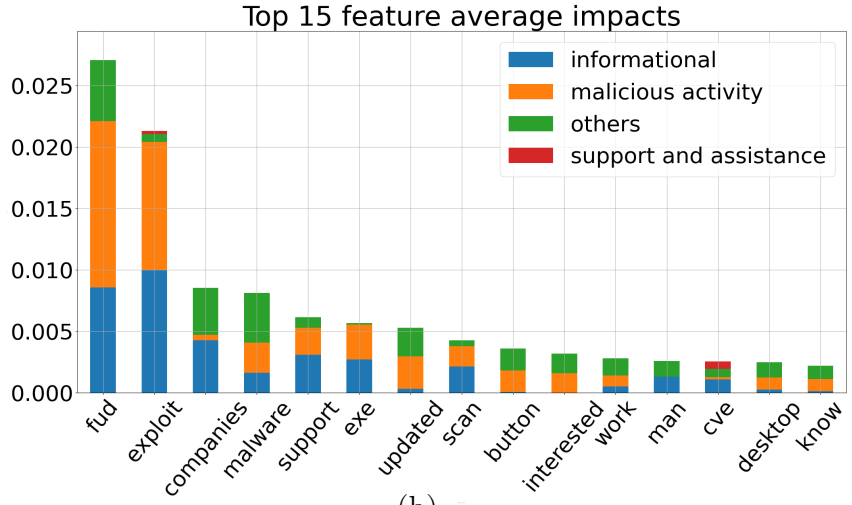


Figure 16 – The 15 most relevant features based on SHAP explanation values were determined from models trained with (a) the **PostCog post type** labels and (b) the **ChatGPT post type** labels.

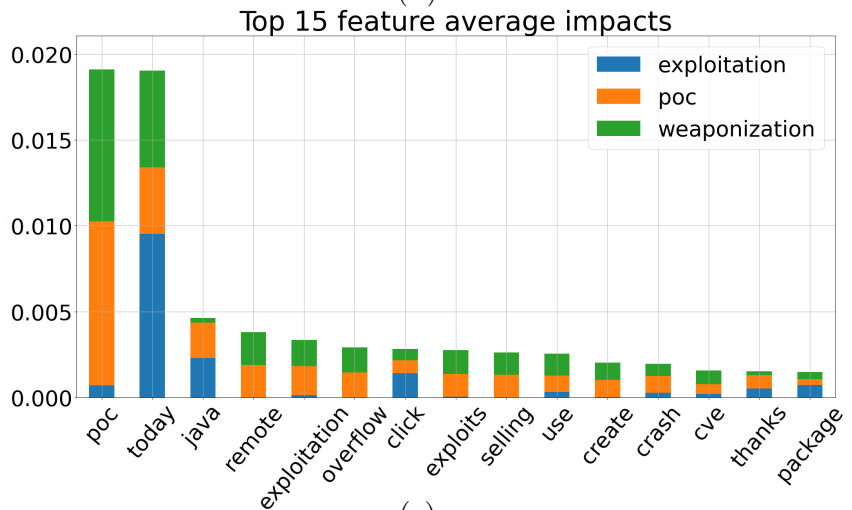
ware", "scan", "interested", and "man" is are informative for "informational" and "malicious activities". These explanations show the significance of feature selection; some words are valuable due to their context, e.g., words such as "nice", "lol", among others. In Figure 17 (c) we report results from our previous model, with 86% of accuracy, as reported in (MORENO-VERA et al., 2023); using SHAP, we identify that the words "remote", "pm", "today", "java", "selling", "crash" and "use" are relevant for classification purposes, in agreement with the findings reported in (MORENO-VERA et al., 2023) using alternative methods.



(a)



(b)



(c)

Figure 17 – The most 15 relevant features SHAP explanation values calculated from models trained using the **expert annotation** labels (a); the **ChatGPT expert annotation** labels (b), and the **previous work** results (c)

5.4.4 Conclusions of the experiment

In this experiment, we classified underground forum posts by cybercriminal activity, textual intentions, and content type using TF-IDF, a RandomForest classifier, and we also used the SHAP method for model explanation analysis. In addition, we observed that ChatGPT introduced noisy labels in several instances. Its tendency to generate plausible but not always accurate or relevant information can lead to misleading or irrelevant features in the training data. This may degrade classifier performance by skewing feature importance or introducing bias. Thus, while ChatGPT can be a valuable tool, it is crucial to use it carefully and to complement it with rigorous data validation and preprocessing steps to mitigate potential drawbacks.

6 CONCLUSIONS

"Data is power as long as you know how to wield it."

In the field of threat intelligence, there has been a growing trend towards leveraging data-driven approaches to inform security decisions. Whether for threat classification, patch management, or threat detection, data is playing an increasingly transformative role in modern cybersecurity practices. Traditionally, security operators have relied on threat intelligence (TI) feeds, including IP blacklists, file signatures, and textual descriptions, to guide their decision-making. In summary, our experiments demonstrate the potential of machine learning and natural language processing techniques to analyze and categorize content in underground forums, aiding in the identification and prioritization of emerging cybersecurity threats.

We utilized CrimeBB, the PostCog framework, and machine learning techniques to analyze textual content and classify it based on the level of exploitation. Our empirical findings, derived from correlating CVSS and EPSS scores with CVEs identified in CrimeBB, revealed that the most frequently cited CVEs are typically associated with higher risks. This suggests that it is feasible to automatically filter exploitation-related threads with high accuracy using our content classifier. Additionally, we performed an unsupervised analysis using topic modeling, which demonstrated that discussions can be grouped based on relevant terms. By applying GPT models to analyze discussion threads, we were able to generate new labels based on the content, providing valuable insights into what GPT models prioritize compared to expert annotations.

Overall, our experiments underscore the effectiveness of combining machine learning, unsupervised learning, and explainable AI techniques in threat detection and cyber intelligence, while also highlighting the importance of rigorous validation in leveraging AI-assisted methods. These approaches can significantly enhance threat analysis capabilities, enabling a proactive and informed cybersecurity posture.

REFERENCES

ABLON, L.; LIBICKI, M. C.; GOLAY, A. A. *Markets for cybercrime tools and stolen data: Hackers' bazaar*. [S.l.]: Rand Corporation, 2014. Citado na página 13.

AHMED, T.; DEVANBU, P. Few-shot training llms for project-specific code-summarization. In: *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*. [S.l.: s.n.], 2022. p. 1–5. Citado na página 37.

ALLODI, L. Economic factors of vulnerability trade and exploitation. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017. Disponível em: https://api.semanticscholar.org/CorpusID:20070349. Citado 6 vezes nas páginas 13, 29, 34, 41, 42, and 43.

ANDERSON, R. et al. Measuring the changing cost of cybercrime. *The 2019 Workshop on the Economics of Information Security*, 2019. Citado na página 13.

BADA, M.; PETE, I. An exploration of the cybercrime ecosystem around shodan. *2020 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, p. 1–8, 2020. Disponível em: https://api.semanticscholar.org/CorpusID:231204238. Citado na página 29.

BASHEER, R.; ALKHATIB, B. Threats from the dark: a review over dark web investigation research for cyber threat intelligence. *Journal of Computer Networks and Communications*, Hindawi Limited, v. 2021, p. 1–21, 2021. Citado na página 13.

BENGIO, Y. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, v. 2, p. 1–127, 2007. Disponível em: https://api.semanticscholar.org/CorpusID:207178999. Citado na página 18.

BLEI, D. M.; NG, A.; JORDAN, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res.*, v. 3, p. 993–1022, 2001. Citado na página 52.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: ACM. *Proceedings of the fifth annual workshop on Computational learning theory*. [S.l.], 1992. p. 144–152. Citado na página 48.

BREIMAN, L. Random forests. *Machine Learning*, v. 45, p. 5–32, 2001. Citado na página 48.

BUTLER, S. Cyber 9/11 will not take place: A user perspective of bitcoin and cryptocurrencies from underground and dark net forums. In: *Workshop on Socio-Technical Aspects in Security and Trust*. [s.n.], 2020. Disponível em: https://api.semanticscholar.org/CorpusID:235600176. Citado na página 29.

CAINES, A. et al. Aggressive language in an online hacking forum. In: *Workshop on Abusive Language Online*. [s.n.], 2018. Disponível em: https://api.semanticscholar.org/CorpusID:53651435. Citado na página 29.

CAINES, A. et al. Automatically identifying the function and intent of posts in underground forums. *Crime Science*, v. 7, 2018. Citado 3 vezes nas páginas 29, 35, and 55.

CAMPOBASSO, M.; ALLODI, L. Threat/crawl: a trainable, highly-reusable, and extensible automated method and tool to crawl criminal underground forums. In: *APWG eCrime 2022*. [S.l.: s.n.], 2022. ArXiv:2212.03641. Citado na página 13.

CHEN, D. D. et al. Towards automated dynamic analysis for linux-based embedded firmware. In: *Network and Distributed System Security Symposium*. [S.l.: s.n.], 2016. Citado na página 29.

CHUA, Y. T.; WILSON, L. Beyond black and white: the intersection of ideologies in online extremist communities. *European Journal on Criminal Policy and Research*, v. 29, p. 337–354, 2023. Disponível em: https://api.semanticscholar.org/CorpusID:261128636. Citado na página 29.

CORTES, C.; VAPNIK, V. N. Support-vector networks. *Machine Learning*, v. 20, p. 273–297, 1995. Disponível em: https://api.semanticscholar.org/CorpusID:52874011. Citado na página 48.

DEGUARA, N. et al. Threat miner: A text analysis engine for threat identification using dark web data. In: *Big Data*. [S.l.: s.n.], 2022. p. 3043–3052. Citado na página 28.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. Disponível em: https://arxiv.org/abs/1810.04805. Citado na página 25.

EDKRANTZ, M.; TRUVÉ, S.; SAID, A. Predicting vulnerability exploits in the wild. *2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing*, p. 513–514, 2015. Citado na página 29.

FANG, Y. et al. Analyzing and identifying data breaches in underground forums. *IEEE Access*, v. 7, p. 48770–48777, 2019. Disponível em: https://api.semanticscholar.org/CorpusID:131776921. Citado na página 29.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016. Citado na página 18.

HANKS, C. et al. Recognizing and extracting cybersecurity entities from text. In: *ICML*. [S.l.: s.n.], 2022. Citado na página 13.

HARRIS, Z. S. Distributional structure. In: . [s.n.], 1954. Disponível em: https://api.semanticscholar.org/CorpusID:86680084. Citado 2 vezes nas páginas 21 and 40.

HOSMER, D. W.; LEMESHOW, S.; STURDIVANT, R. X. Applied logistic regression: Hosmer/applied logistic regression. In: . [S.l.: s.n.], 2005. Citado na página 48.

JACOBS, J.; ROMANOSKY, S. et al. Exploit prediction scoring system (EPSS). *Digital Threats: Research and Practice*, ACM New York, NY, USA, v. 2, n. 3, p. 1–17, 2021. Citado na página 28.

LE, Q. V.; MIKOLOV, T. Distributed representations of sentences and documents. In: *International Conference on Machine Learning*. [s.n.], 2014. Disponível em: https://api.semanticscholar.org/CorpusID:2407601. Citado na página 23.

LEMAîTRE, G.; NOGUEIRA, F.; ARIDAS, C. K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, v. 18, n. 17, p. 1–5, 2017. Citado na página 49.

LIANG, H. et al. Fuzzing: State of the art. *IEEE Transactions on Reliability*, v. 67, p. 1199–1218, 2018. Citado na página 29.

LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. In: *Neural Information Processing Systems*. [S.l.: s.n.], 2017. Citado 2 vezes nas páginas 55 and 56.

MAN, J.; SIU, G. A.; HUTCHINGS, A. Autism disclosures and cybercrime discourse on a large underground forum. *2023 APWG Symposium on Electronic Crime Research (eCrime)*, p. 1–14, 2023. Disponível em: https://api.semanticscholar.org/CorpusID:268499519. Citado na página 29.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. In: *International Conference on Learning Representations*. [s.n.], 2013. Disponível em: https://api.semanticscholar.org/CorpusID:5959482. Citado 2 vezes nas páginas 23 and 40.

MORENO-VERA, F. Inferring discussion topics about exploitation of vulnerabilities from underground hacking forums. In: *ICTC*. [S.l.: s.n.], 2023. p. 816–821. Citado 3 vezes nas páginas 14, 16, and 29.

MORENO-VERA, F.; MENASCHE, D.; LIMA, C. Beneath the cream: Unveiling relevant information points from crimebb underground forums with its ground truth labels. In: *International Symposium on Cyber Security, Cryptology and Machine Learning*. [S.l.]: Springer, 2024. p. 280–290. Citado na página 16.

MORENO-VERA, F. et al. Cream skimming the underground: Identifying relevant information points from online forums. In: *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*. [S.l.: s.n.], 2023. p. 66–71. Citado 10 vezes nas páginas 14, 16, 28, 29, 34, 35, 53, 54, 55, and 59.

MORROW, T. C. S. The future of cybercrime & security. 2019. Citado na página 13.

PARACHA, A.; ARSHAD, J.; KHAN, M. M. SUS You're SUS!-Identifying Influencer Hackers on Dark Web Social Networks. *Computers and Electrical Engineering*, Elsevier, 2023. Citado na página 41.

PASTRANA, S. et al. Characterizing eve: Analysing cybercrime actors in a large underground forum. In: *International Symposium on Recent Advances in Intrusion Detection*. [s.n.], 2018. Disponível em: https://api.semanticscholar.org/CorpusID:51686519. Citado na página 29.

PASTRANA, S.; HUTCHINGS, A. et al. Measuring ewhoring. In: *Proceedings of the Internet Measurement Conference*. [S.l.: s.n.], 2019. p. 463–477. Citado na página 28.

PASTRANA, S.; THOMAS, D. R. et al. CrimeBB: Enabling cybercrime research on underground forums at scale. In: *Proceedings of the 2018 World Wide Web Conference*. [S.l.: s.n.], 2018. p. 1845–1854. Citado 4 vezes nas páginas 13, 14, 28, and 31.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: *Conference on Empirical Methods in Natural Language Processing*. [s.n.], 2014. Disponível em: https://api.semanticscholar.org/CorpusID:1957433. Citado na página 23.

PETE, I. et al. Postcog: A tool for interdisciplinary research into underground forums at scale. *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, p. 93–104, 2022. Citado 2 vezes nas páginas 31 and 32.

RADFORD, A.; NARASIMHAN, K. Improving language understanding by generative pre-training. In: . [s.n.], 2018. Disponível em: https://api.semanticscholar.org/CorpusID:49313245. Citado na página 25.

RAHMAN, M. R. et al. What are the attackers doing now? automating cyberthreat intelligence extraction from text on pace with the changing threat landscape: A survey. *ACM Computing Surveys*, ACM New York, NY, 2021. Citado na página 28.

REHUREK, R.; SOJKA, P. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, v. 3, n. 2, 2011. Citado na página 52.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *SIGKDD*, 2016. Citado 2 vezes nas páginas 55 and 56.

SALTON, G.; BUCKLEY, C. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.*, v. 24, p. 513–523, 1988. Citado 2 vezes nas páginas 22 and 40.

SIU, G. A.; COLLIER, B.; HUTCHINGS, A. Follow the money: The relationship between currency exchange and illicit behaviour in an underground forum. *EuroS&PW*, p. 191–201, 2021. Citado 3 vezes nas páginas 29, 35, and 55.

SPEYBROECK, N. Classification and regression trees. *International Journal of Public Health*, v. 57, p. 243–246, 2012. Citado na página 48.

SUCIU, O. et al. Expected exploitability: Predicting the development of functional vulnerability exploits. In: *31st USENIX Security Symposium*. [S.l.: s.n.], 2022. p. 377–394. Citado na página 28.

SUN, J. et al. Generating informative cve description from exploitdb posts by extractive summarization. *ArXiv*, abs/2101.01431, 2021. Disponível em: https://api.semanticscholar.org/CorpusID:230523926. Citado na página 29.

SUTTON, M. S.; GREENE, A. R.; AMINI, P. F. Fuzzing: Brute force vulnerability discovery. In: . [S.l.: s.n.], 2007. Citado na página 29.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. *Journal of the royal statistical society series b-methodological*, v. 58, p. 267–288, 1996. Citado na página 48.

TIKHONOV, A. N. On the stability of inverse problems. In: *Dokl. Akad. Nauk SSSR*. [S.l.: s.n.], 1943. v. 39, p. 195–198. Citado na página 48.

VALEROS, V.; GARCIA, S. Growth and commoditization of remote access trojans. In: IEEE. *2020 IEEE EuroS&PW*. [S.l.], 2020. p. 454–462. Citado na página 42.

WANG, B. et al. Characterizing and modeling patching practices of industrial control systems. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, ACM New York, NY, USA, v. 1, n. 1, p. 1–23, 2017. Citado na página 43.