YACSDB-NER: Yet Another Cybersecurity Database for Named Entity Recognition Task

Yuri do Amaral Nobre Maia[1], Robson de Oliveira Albuquerque[1 2], and Demétrio Antônio da Silva Filho[1]

[1]University of Brasília (UnB), Brasília DF 70910-900, Brazil
[2]Catholic University of Brasília (UCB), Brasília DF 71966-700, Brazil
yuri.maia@aluno.unb.br, robson@redes.unb.br, dasf@unb.br

Abstract. The increasing number of cybersecurity reports poses a challenge to efficiently retrieving and sharing Cyber Threat Intelligence. However, publicly available cybersecurity datasets for Natural Language Processing (NLP) remain scarce, hindering progress in automated intelligence production. To tackle this challenge, this article presents Yet Another Cybersecurity Database (YACSDB), a dataset designed to enhance Named Entity Recognition (NER) using Structured Threat Information Expression (STIX) entities for interoperability. Our pipeline extracts STIX Domain Objects from unstructured reports, leveraging Google's Gemini and Bidirectional Encoder Representations from Transformers (BERT) model to assist in labeling and reduce resource needs. The dataset uses Inside–Outside–Beginning (IOB) notation to facilitate fine-tuning in sequence tagging tasks. Reports were selected for representativeness across different years. To the best of our knowledge, it is among the largest cybersecurity NER dataset with temporal information annotated by a single machine-assisted annotator. To evaluate the dataset, we fine-tuned seven BERT models to demonstrate its effectiveness for NER. The results emphasize the importance of domain-specific datasets in cybersecurity NLP and highlight key challenges. YACSDB serves as a benchmark for model comparison, solution development and knowledge graph generation. It is publicly available to foster future research in cybersecurity NLP.

Keywords: BERT · Gemini · Named Entity Recognition · STIX.

1 Introduction

Ransomware attacks on critical infrastructure and large-scale data breaches affecting countless individuals highlight a growing threat landscape [23]. In response, cybersecurity reports are extensively published to analyze and document emerging threats.

Advances in Natural Language Processing (NLP), particularly with the advent of Large Language Models, improved the automation of information extraction from cybersecurity reports [22]. Furthermore, Transformer models reshaped

numerous NLP tasks by enabling machines to retrieve deep, context-aware information with high accuracy [28]. Bidirectional Encoder Representations from Transformers (BERT) [8] stands as a foundational Transformer model, showing remarkable performance through domain adaptation for specialized tasks [20]. Consequently, researchers also leveraged this approach within the cybersecurity domain [1,5,22].

In cybersecurity domain, Named Entity Recognition (NER) poses a challenge since most general-purpose corpora used for pre-training models contain text whose meaning can diverge significantly when applied to the cybersecurity domain [5]. Important to note that NER is a preceding problem to more complex classification problems, such as relation extraction and knowledge graph generation. Despite the extensive amount of cybersecurity reports, publicly available datasets for NER remain scarce, as it is noted on studies [3,19,22,23,24,32]. Moreover, labeling a dataset require great specialized effort. For example, the development of APTNER [30] engaged 36 participants. However, this process can be facilitated with generative models, such as Gemini [11], that achieves impressive results in many text benchmarks [14].

To bridge this data gap and foster further developments of robust NER models for cyber security, we propose an iterative method to extract entities from cybersecurity reports and generate a dataset for fine-tuning in NER task. The main contributions of this studies are the following:

- A pipeline using BERT and Gemini models to preprocess, analyze, and compile cybersecurity reports into a dataset;
- Our public dataset, named Yet Another Cybersecurity Database for Named Entity Recognition (YACSDB$_{NER}$), consisting of annotated text spans for NER of STIX entities;
- An evaluation of YACSDB$_{NER}$ on the main domain-adapted BERT-based models in cybersecurity.

The remainder of the paper is organized as follows. First, Section 2 sets the theoretical framework and explores related work on NER task in cybersecurity and available datasets. Building on this foundation, Section 3 describes the pipeline to achieve the final dataset. Section 4 details the setup for evaluating the dataset. Following this, Section 5 presents the findings and discussion over the results. Finally, the conclusion of this study is included in Section 6 along with future works.

YACSDB$_{NER}$ is available on GitHub `https://github.com/boutdatansec/YACSDB`.

2 Main Concepts and Related Work

This section introduces the concepts, models and challenges related to the scarcity of NER datasets in cybersecurity. First, we will outline the key methods used for NLP and applied to cyber domain. Next, we will describe the core taxonomy

of STIX. Later, a literature review on studies regarding NER task in cybersecurity and related researches in Machine Learning (ML). Lastly, most relevant studies on the development of datasets for Cyber Threat Intelligence (CTI) are explored, highlighting their characteristics and contributions.

2.1 NLP in Cybersecurity

The application of deep neural networks, including Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Long Short-Term Memory (LSTM) networks, has started a new phase in NLP by improving the capacity to process unstructured textual data. Transformer architectures [28] upgraded this progress, leading to the Bidirectional Encoder Representations from Transformers (BERT) model, which, through its deep bi-directionality and large-scale pre-training, reshaped NLP . Collectively, these advancements offer powerful tools for extracting critical information from cybersecurity reports and other domain-specific texts.

BERT [8] is a model built on the Transformer encoder architecture pre-trained in large text corpora on two tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). BERT's groundbreaking performance in NLP initiated a series of subsequent studies. Among these, Robustly Optimized BERT Pretraining Approach (RoBERTa) [18] introduced key modifications, for instance, employing an adapted Byte-Pair Encoding for tokenization and altering pre-training objectives. These changes, alongside other optimizations, lead to enhanced performance.

In these Large Language Models (LLMs), pre-training is a self-supervised step for transfer learning, enabling the model to acquire broad general linguistic knowledge from large unlabeled text data. It often reduces the necessity for extensive domain-adaptive pre-training (DAPT) [25] in specialized fields. This is fundamental for the application on low resource domain-specific tasks such as cybersecurity NER.

In addition to discriminative models like BERT, which primarily focus on understanding and classifying input, autoregressive generative models have gained significant attention, exemplified by Gemini [11]. This generative, multimodal LLM is based on Transformer decoders to produce coherent text and images. For the purpose of this study, we will consider solely its text-to-text capability. Its following version, Gemini 1.5 [10] expanded long-context capabilities from the previous version, while the Flash variant achieves comparable performance, despite its smaller scale.

2.2 STIX

A standardized, structured language is needed for efficient sharing of CTI. Structured Threat Information Expression (STIX) [15] project provides a common vocabulary for communication among different organizations. The project states that three main objects compose the STIX Core Objects, which may be used to share broad and comprehensive CTI: STIX Domain Objects (SDO), STIX

Cyber-observables Objects (SCO) and STIX Relationship Objects (SRO). SDO are Higher Level Intelligence Objects, which are common concepts an analyst would need to describe a CTI. There are 19 classes of SDO, as defined in Table 1. SCO are actual host-based and network-based information, which is close to the concept of Indicator of Compromise to Threat Intelligence. SRO is, straightforward, the relation between objects.

When analyzing cybersecurity reports, they frequently do not encompass all objects. This study focuses on retrieving information related to the threat. SDO of interest are highlighted on Table 1, along comments on the remaining objects.

Table 1. STIX Domain Objects and their Descriptions [15]

STIX Domain Object	**Description**
Attack Pattern	A type of TTP (Tactics, Techniques, and Procedures) describing ways adversaries attempt to compromise targets.
Campaign	A grouping of adversary behavior describing its malicious activities over a period of time against a specific set of targets.
Course of Action	A recommendation or action to mitigate, respond to, or prevent threats. *As it focus on defense, it is out-of-scope.*
Grouping	A collection of STIX Objects with a shared context, but without requiring a relationship between them. *It works as a meta-object before the finished intelligence object, therefore, out-of-scope.*
Identity	A characterizing object that defines individuals, organizations, systems or groups.
Incident	*Yet to be defined by OASIS.*
Indicator	A pattern used to detect suspicious or malicious activity. It may be defined in terms of SCO.
Infrastructure	It describes systems, services, or any physical or virtual resources used by threat actors. *It is listed as Indicator.*
Intrusion Set	A grouped set of adversary behavior and resources used repeatedly across multiple attacks believed to belong to a single organization. It is a superset of campaigns and activities that indicate a Threat Actor. *There is not an objective criterion to differentiate from campaigns, which will be considered in the study.*
Location	A specific physical or geopolitical location.
Malware	A type of TTP describing malicious code, should it be a program, payload or another.
Malware Analysis	A specific assessment of malware samples, providing details on its behavior, capabilities, and indicators. It is described in the intelligence product and is excluded to avoid redundancy.
Note	A textual annotation to provide context that can't be represented by STIX Objects. *It is a subjective annotation, therefore, it is out-of-scope.*
Observed Data	Represents artifacts related to cybersecurity entities using SCO. *It is closer to raw data, on reports it is expected to be Indicators.*
Opinion	An evaluation of the information that was produced by another entity. *It falls in the same category as Note.*
Report	A set of CTI describing the details and context over one or more topics. *All analyzed inputs are Reports, reason it is not necessary.*
Threat Actor	An individual, group, or organization conducting malicious activities.
Tool	A legitimate software, script, or utility that can be used on cyberattacks.
Vulnerability	A weakness in software or hardware can be exploited for malicious means.

2.3 NER in Cybersecurity

Numerous studies have investigated text mining techniques to address the challenge of automated CTI extraction from textual sources. The literature review by Rahman *et al.* [21] observed that Threat Reports were the primary source of CTI. Many of the analyzed studies applied NLP for the purpose of extrac-

tion of Indicators of Compromise (IoC), an intrinsically NER task, and Tactics, Techniques and Procedures (TTPs).

Accordingly, cybersecurity researchers applied diverse NER strategies tailored on specific goals. Table 2 summarizes six studies in NER, illustrating two key aspects: the variety of research goals and the requirement for manually annotate datasets.

Table 2. NER studies summary

Study	Year	Goal	Data Source	NER Strategy	Dataset Availability
Yi et al. [32]	2020	NER from reports	14,000† security texts from CERT-CN	RDF-CRF + regex + known-entity dictionaries	No
TIMiner [35]	2020	IoC extraction and CTI categorization from social media	15,000† annotated	Regex + BiLSTM-CRF, word similarity	No
TCENet (TIM) [33]	2022	TTP classification for STIX/Sigma export	10,761† reports of 5 security vendor blogs	SentenceBERT embedding, one CNN, one BiLSTM with attention mechanism	No
Alves et al. [2]	2022	BERT variants evaluation for MITRE TTP classification	10,360 sentences from MITRE curated examples + 80† sentences	11 BERT variants	Yes
STIXnet [19]	2023	STIX NER and RE	MITRE threat actor descriptions†	Regex, Knowledge Base, rcATT [16], rule-based + SentenceBERT	Yes
KnowCTI [29]	2024	NER and RE with Graph Neural Networks	4,896† texts for classification + 8,872† texts for CTI extraction	BERT embedding, Graph Attention Network	No

†manual annotation

Modern NER studies in cybersecurity leverages BERT model, as adapting it to specific domains is often beneficial, as observed by Peng *et al.* [20]. The researchers found that using the general BERT vocabulary outperformed domain-specific variants on most datasets. Training on small corpora also risks performance instability. A set of studies on DAPT in cybersecurity is presented in Table 3. The models are based on BERT or RoBERTa with different strategies. SecBERT[1] and SecRoBERTa[2] do not have any publication regarding the training method and evaluations, but they were further evaluated in this study due their popularity on Hugging Face, an open-source platform for Machine Learning.

NER with BERT is a sequence tagging task, and the choice of the annotation schema can affect model performance [27]. This study does not intend to explore these differences; thus, the Inside-Outside-Beginning (IOB) notation was chosen as the output labeling format. An identified entity will have its first token classified as "B-" (Beginning) of the specific label type, while all subsequent tokens within that entity will be classified as "I-" (Inside). Tokens that are not named entities are labeled as "O" (Outside). Figure 1 demonstrates the use of this notation with word tokenization.

[1] `https://huggingface.co/jackaduma/SecBERT`
[2] `https://huggingface.co/jackaduma/SecRoBERTa`

A	HIDDEN	COBRA	server	delivers	DeltaCharlie	malware	.
O	B-threat-actor	I-threat-actor	O	O	B-malware	O	O

Fig. 1. Example of IOB notation

Table 3. Studies adapting BERT model

Model	**Year**	**Training**	**Data Source**	**Training size**	**NER Downstream**
RuCyBERT [26]	2020	DAPT BERT on a Russian cybersecurity corpus with modified vocabulary	Sec_col corpus (augmented), security reports	500,000 texts	Yes
CyBERT (Ameri et al.) [3]	2021	Fine-tuning BERT on a cybersecurity feature claims	ICS device information documents	41,073,376 words	No
CyBERT (Ranade et al.) [22]	2021	DAPT BERT on a cybersecurity corpus with vocabulary extension	Security news, *CVE* vulnerability reports and *APTNotes* reports	17,000 texts	Yes
SecBERT and SecRoBERTa	2022	DAPT BERT and RoBERTa on cybersecurity corpora with vocabulary extension	APTnotes, Stucco, CASIE, SemEval 2018 Task 8	Not disclosed	No
SecureBERT [1]	2023	DAPT RoBERTa on a cybersecurity corpus with vocabulary extension	Varied cybersecurity-related text	1,072,798,637 words	Yes
CySecBERT [5]	2024	DAPT BERT on a cybersecurity corpus	Blogs, arXiv, National Vulnerability Data, Twitter	4.3 million entries	Yes

2.4 Datasets

General-purpose NER algorithms have been extensively studied. However, their performance may not be sustained when applied to the cybersecurity domain [6]. Although deep learning models can leverage transfer learning, their ability to maintain performance across different domains cannot be guaranteed. This challenge motivates the creation of domain-specific corpora. A comprehensive dataset overview is presented on Table 4.

MalwareTextDB [17] is a dataset constructed by annotating 39 malware reports of 2014 from APTnotes using the MAEC vocabulary. Conversely, DNRTI [31] involved annotating open-source threat intelligence reports into custom classes, though details regarding the source reports are no provided. Hanks *et al.* [13] retrieved open-source reports and analysis to annotate into cybersecurity-relevant entities. APTNER [30] also utilized open-source CTI reports, which were manually annotated by a 36 people team, but it lacks specific details about the source of these reports. Notably, it has the largest number of entities. Siracusano *et al.* [24] presents a dataset of open-source reports from 62 sources, yet the dataset is not public as the link is hidden for anonymity. More recently, AttackER [7] introduced a dataset for cyber-attack attribution from various security blogs reports.

While the FEW-NERD dataset [9] is unrelated to cybersecurity, it offers key takeaways for building multiclasses datasets. This large-scale dataset was

manually annotated and specifically designed for few-shot NER, comprising of 188,238 sentences labeled in a hierarchy of 8 coarse-grained classes and 66 fine-grained classes. Despite BERT's strong result on other datasets, it struggles on FEW-NERD, which is suggested to be due to the larger number of types present in the dataset.

Table 4. Dataset Overview

Dataset	**Data Sources**	**Classes**	**Entities**	**Relations**	**Publicly Available**
MalwareTextDB [17] 2017	39 reports 2,080 sentences	4 + 444	10,983 + 7,102	8,705	Yes
DNRTI [31] 2020	300+ reports 6,570 sentences	27	36,412	–	Yes
Hanks et al. [13] 2022	380 reports 1,339 sentences	29	801	–	Partial
APTNER [30] 2022	10,984 sentences	7 SDO + 12 SCO + 2	39,565	–	Yes
Siracusano et al. [24] 2023	204 reports	9 SDO	36,100	13,600	No
AttackER [7] 2024	217 reports 2,640 sentences	14 SDO + 4	7,026	–	Yes
YACSDB$_{NER}$ 2025	422 reports 29,878 sentences	8 SDO	15,140	–	Yes

3 Dataset Generation

To propose YACSDB, we aim to address the downstream training for NER in cybersecurity. As seen in Subsection 2.3, recent strategies use LLM approach. Downstreaming can be done in considerably smaller datasets when compared to DAPT, provided that they are labeled. Therefore will be explored the annotation method.

Studies have observed the lack of publicly available dataset [3,19,22,23,24,32] and generated their own dataset, which hinders the comparison among them. This has been stated on many classification problems such as sequence tagging, NER, RE and knowledge graph generation. NER is a preceding problem for the others, hence it will be addressed.

Existing datasets may use different formats. To address it, STIX has been chosen as common representation language. The first scope will be SDO as the coarse-grained classes. Token tagging format will be used because converting to word or span tagging from it is straightforward. Analyzing the classes, Attack Pattern is multi-word and may overlap with other classes. The single-class problem will be addressed initially, thus Attack Pattern is out of scope. Final format used is IOB notation.

We selected VX-Underground, a website about malware and cybersecurity containing a series of reports over APTs split by year, as main source of texts.

Original sources may be security vendors' blogs, CERT incident report, X (former Twitter) posts, private researchers reports, and others.

The pipeline for building YACSDB has four steps: preprocessing, processing and analysis, classification process and evaluation.

3.1 Preprocessing

Preprocessing consisted in scrapping the reports, extracting the text from files, cleaning data and selecting sentences.

There were 2,164 reports listed from 2010 to 2023 which were scrapped using Python, Selenium and BeautifulSoup. Due to duplication, broken links, or empty files, the total amount of files is 2,068 in PDF format. Text retrieval was accomplished with PyMuPDF[3] using fitz natural sort.

Paragraphs in non-english language or containing symbols were stripped. Each paragraph was broken into sentences using spaCy's[4] English general-purpose large model. Only sentences with verbs were considered for this step. We retrieved 314,323 sentences unprocessed of which 263,066 remained valid for analysis.

3.2 Processing and analysis

A primary concern is ensuring the dataset's representativeness of real-world reports and the sufficiency of instances for each label, as data scarcity negatively impacts the model training [4]. Therefore, an evaluation is proposed to address these aspects.

Inspired by AttackER's results with generative models in NER tasks [7], we employ a zero-shot learning approach using the Gemini Flash 1.5 model to assess the quality of the reports in terms of label coverage as a preliminary metric. The model was prompted to extract SDO entities. To address hallucination, all extracted entities were cross-checked against the source text, with only verified instances being retained. In total, 1,924 reports yielded successful entity extraction through this method.

These reports were compared in terms of extracted entities per 1,000 characters (EPTL). Excluding documents primarily composed of IoC listings, an EPTL ratio between 5 and 20 was consistently observed, regardless of report length, as illustrated in Figure 2. To optimize processing efficiency and maximize report inclusions, we excluded instances with larger lengths, given their relatively uniform entity density. Additionally, reports from 2010 to 2014 were excluded due to their sparse entity instances. The initial version of the dataset comprises 1,127 candidate reports, as detailed in Table 5

[3] `https://pymupdftest.readthedocs.io/en/stable/intro.html`
[4] `https://spacy.io/`

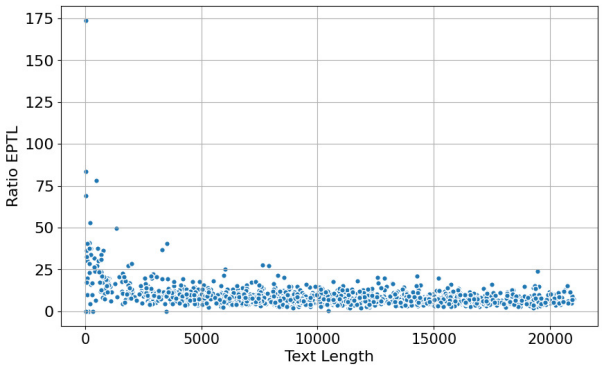

Fig. 2. Ratio of Entities per Text Length (EPTL) versus text length.

Table 5. Number of reports per year

	2015	2016	2017	2018	2019	2020	2021	2022	2023
Reports	73	76	77	106	120	95	85	300	195

3.3 Classification process

We propose an LLM-assisted methodology for building the dataset for NER. The primary objective of this approach is assist cybersecurity experts in the labeling task by leveraging Gemini and BERT models.

Gemini model supports fine-tuning on input-output pair tasks. For a zero-shot prompt approach, the input examples should mirror the instructions provided in inference, enabling the model to internalize patterns. Google recommends providing a minimum of 100 examples for fine-tuning in classification task and up to 500 for summarization task[5]. While it is not stated in its documentation, this fine-tuning process is likely a form of Parameter-Efficient Fine-Tuning (PEFT) [12].

We executed three successive rounds of fine-tuning on the Gemini model. Each model M_n initiates its training from the final last state of preceding model M_{n-1}. We adopted pseudo-labeling strategy, where model M_n is fine-tuned on dataset D_n. Dataset D_n comprises pseudo-lables generated by M_{n-1} on an unlabeled sample set $\mathcal{X}_n$. Let $M_{n,\theta}$ be the Gemini model of round n with parameters θ, and $\mathcal{L}$ represents the loss function applied to M_n on D_n. The fine-tuning process can be denoted by *Eq. 1.*

$$M_n = \arg\min_{\theta_n} \sum_{i=1}^{n} \mathcal{L}(M_{i-1,\theta_{i-1}}, D_{i-1}) \tag{1}$$

To initiate training, we use the Gemini 1.5 Flash base model `gemini-1.5-flash-001-tuning` as M_0. Our initial dataset D_0 is derived from STIXnet's eval-

[5] `https://ai.google.dev/gemini-api/docs/model-tuning#size-recommendation`

uation dataset [19], comprising 52 APT reports with 1,407 entities. We made minor adaptations to this dataset to align it strictly with SDO. For training, the texts are segmented into spans of at most 128 tokens using a BERT tokenizer, a strategy that will be later leveraged for BERT's training. This process yields an initial dataset of 152 labeled spans.

For each round, a small unlabeled sample is selected for inference. This strategy is employed to gradually enhance the quality of model M and to generate an initial ground truth with the sufficient number of instances recommended by Google. The resulting dataset contains 416 spans. Hyperparameters used are listed in Table 6, and all models have the temperature set to 0.3.

Table 6. Training Parameters for Gemini Fine-tune

Parameter	M_1	M_2	M_3
Epoch Count	5	10	10
Batch Size	4	4	4
Learning Rate	0.001	0.0005	0.0005

Alongside Gemini, a BERT model is also trained on the same dataset. Since Gemini does not output token probabilities, we cannot directly measure its labeling confidence. Thereby, we use the BERT score as a proxy for labeling confidence. This data is then ingested in LabelStudio[6], an open-source data labeling tool, for review review.

The final Gemini model M_3 and the fine-tuned BERT model are applied to a subset of 380 candidate reports. The resulting dataset is then evaluated.

3.4 Evaluation

To ensure correctness of the final dataset, a manual review process was implemented, comparing the inferences from both models. Label consensus between the models' prediction supported by high confidence scores from BERT are accepted. All remaining entities are reviewed by an expert with 10 years of experience in cybersecurity. Spans without entities are removed to favor higher label density. The F1-score for Gemini M_3 model alone yielded F1-score of 0.904.

The YACSDB$_{NER}$ dataset comprises the aggregation of the newly annotated dataset and the incorporated content of the reviewed STIXnet dataset. Its main characteristics are described in Table 7. Sentence counts are obtained using the spaCy model from Subsection 3.1. The number of classes considers 'B-' and 'I-' tags for each entity, alongside the 'O' class, totaling 17 classes, whereas total entities consider the 8 entities types, taking into account subsection 2.2 and section 3 regarding Attack Pattern.

[6] `https://labelstud.io/`

Table 7. YACSDB$_{NER}$ description

Sentences	24,878
Spans	8,169
Total Entities	15,140
Number of Classes	17
Source Reports	422

4 Experiment

To assess the utility of the dataset for NER, we fine-tuned seven BERT-like model on this task. The fine-tuning was conducted within a Google Colab Python 3 environment with NVIDIA T4 GPU.

We split the dataset into a 70:15:15 ratio for training, validation and testing sets. As a baseline, we used BERT$_{frozen}$, a model with frozen base where only the classifier head was trained. Additionally, we evaluated a standard BERT base cased model without any self-supervised DAPT. These models were compared against three DAPT BERT versions and two DAPT RoBERTa versions, all pre-trained on cybersecurity datasets. A simple classifier head $Linear(768 \rightarrow 17)$ was applied on top of each model following the original BERT [8] approach. All the models utilized identical training, validation and testing sets, and were fine-tuned using the Hugging Face library with equal hyperparameters: 5 epochs, a training batch size of 8, a validation batch size of 2, and a learning rate of $1e-05$. All other parameters were set to their default values.

5 Results and Discussion

Table 8 presents the training outcomes with the YACSDB$_{NER}$ dataset. The results indicate that this dataset is suitable for NER and can serve as a benchmark for evaluating DAPT models. The general-purpose BERT outperformed all DAPT models in all metrics, followed by SecureBERT. Benchmarking with this dataset allows direct comparison of models, suggesting that SecureBERT's adaptation strategy is better suited for this specific task. Nevertheless, this comparison also underlines a significant opportunity for improvement in the model adaptation process within the cybersecurity domain.

5.1 Discussion

The strategy adopted of using Gemini 1.5 as a naïve pre-evaluation tool for the report selection is promising; however, the impact of this data preprocessing steps warrants further investigation. Through iterative fine-tuning, we developed the Gemini M_3 model to assist the annotation task, yielding an F1-score of 0.904 and enabling the annotation of close to 25,000 sentences by a single annotator. This is notable when compared to APTNER [30], which required 36 annotator to label nearly 11,000 sentences. Nevertheless, the M_3's F1-score should be interpreted

Table 8. BERT models results

	Precision	Recall	F1-score
BERT_{frozen}	0.136	0.000*	0.001
BERT	**0.767**	**0.806**	**0.786**
CyBERT [22]	0.578	0.629	0.603
CySecBERT [5]	0.567	0.630	0.597
SecBERT	0.552	0.519	0.535
SecRoBERTa	0.572	0.528	0.549
SecureBERT [1]	0.750	0.791	0.770

*value is small, but not zero

with caution, as the training set for each iteration was optimized to favor the annotation process, potentially limiting the model's generalization capabilities. Given the selected emphasis on higher entity density in the dataset, training strategies should consider employing measures to mitigate potential overfitting.

YACSDB_{NER} presents advantages over existing public datasets for cybersecurity NER tasks. MalwareTextDB [17] focus specifically on malware analysis, it is outdated and it does not encompass key STIX entities. DNRTI [31] offers improvements over MalwareTextDB but it still omits SDOs. Hanks el al [13] provides a public corpus without annotations, requiring additional effort for NER. APTNER [30] is a valuable resource with labels for indicators; however, sources and dates of reports are neglected and a workforce of 36 annotators poses practical limitations, reason an LLM-assisted annotation is more accessible. Finally, AttackER [7] includes fewer sentences and annotated entities compared to YACSDB_{NER}.

The performance of fine-tuned models highlights the challenges of supervised NER in the cybersecurity domain. The number of classes may a limitation for BERT models, as it was observed in general-purpose settings [9]. Conversely, the results achieved by BERT reinforce the advantages of downstreaming in domain. However, the observed suboptimal performance of DAPT models warrants further investigation. This finding draws parallels with results in Zanella and Toussaint [34] in the biomedical domain, where certain models did not outperform BERT when employing linear classifiers. Notably, SecureBERT, a RoBERTa-based model, incorporates larger vocabulary modifications – representing 0.35% of total tokens, compared to approximately 0.03% for CyBERT – and benefits from pre-training on a larger corpus than both CyBERT and CySecBERT. The DAPT models may have underperformed due to a mismatch between the linguistic style of their pretraining corpus and that of the YACSDB_{NER} reports. Nonetheless, the overall performance of these DAPT models should also be assessed in light of total dataset size and the distribution of entities in the task.

6 Conclusion

We introduce YACSDB_{NER}, a publicly available dataset for Named Entity Recognition in the cybersecurity domain, structured around the STIX language.

The dataset comprises 24,878 sentences from 422 cybersecurity reports in IOB format. It is designed to extract STIX Domain Objects for easy CTI sharing and supporting the construction of knowledge graphs. YACSDB$_{NER}$ addresses key limitations observed in existing datasets. In addition, we describe a semi-automated annotation pipeline with Gemini, which could reduce annotation costs and promote community-driven dataset enhancement. Overall YACSDB$_{NER}$ addresses the lack of benchmarks for NER models, supporting the development and evaluation of new techniques and models.

We compare the performance of seven BERT-based models fine-tuned on the dataset to assess its effectiveness for cybersecurity NER. A deeper analysis of DAPT BERT models in the cybersecurity domain is deemed. YACSDB$_{NER}$ advances the landscape of NER datasets in the cybersecurity domain, enabling tasks such as the extraction of strucutred knowledge graphs from unstructured text.

For future work, we propose to extend the labeling process to additional reports to expand the dataset and investigate the impact of dataset size on model performance. Furthermore, the contribution of new data to model performance will be assessed to provide insights into concept drift. Building on these efforts, a diagnostic study will be undertaken to investigate the suboptimal performance observed in DAPT models, encompassing a comparative analysis with other Transformer-based architectures beyond BERT.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Acknowledgments. The authors acknowledge the brazilian funding agencies CNPq (Grants 307108/2023-6 and 306080/2022-2), CAPES, FAP-DF (00193-00001831/2023-47), Edital DPI/DPG N. 04/2024, Edital Interno IF/UnB 0004/2024, and INCT-CiMol (CNPq grant 406804/2022).

References

1. Aghaei, E., Niu, X., Shadid, W., Al-Shaer, E.: Securebert: A domain-specific language model for cybersecurity. In: Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST. vol. 462 LNICST (2023). https://doi.org/10.1007/978-3-031-25538-0_3
2. Alves, P., Filho, G.P.R., Gonçalves, V.P.: Modelo de classificação de TTP baseado em transformadas BERT. Proceedings of the Ibero American Conferences on Applied Computing 2022 and WWW/Internet 2022 (2022), https://www.iadisportal.org/digital-library/modelo-de-classifica%C3%A7%C3%A3o-de-ttp-baseado-em-transformadas-bert
3. Ameri, K., Hempel, M., Sharif, H., Lopez, J., Perumalla, K.: Cybert: Cybersecurity claim classification by fine-tuning the bert language model. Journal of Cybersecurity and Privacy **1** (2021). https://doi.org/10.3390/jcp1040031
4. Bayer, M., Kaufhold, M.A., Reuter, C.: A survey on data augmentation for text classification. ACM Computing Surveys **55** (2022). https://doi.org/10.1145/3544558

5. Bayer, M., Kuehn, P., Shanehsaz, R., Reuter, C.: Cysecbert: A domain-adapted language model for the cybersecurity domain. ACM Transactions on Privacy and Security **27** (2024). https://doi.org/10.1145/3652594
6. Dasgupta, S., Piplai, A., Kotal, A., Joshi, A.: A comparative study of deep learning based named entity recognition algorithms for cybersecurity. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 2596–2604. IEEE (2020)
7. Deka, P., Rajapaksha, S., Rani, R., Almutairi, A., Karafili, E.: Attacker: towards enhancing cyber-attack attribution with a named entity recognition dataset. In: International Conference on Web Information Systems Engineering. pp. 255–270. Springer (2024)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). pp. 4171–4186 (2019)
9. Ding, N., Xu, G., Chen, Y., Wang, X., Han, X., Xie, P., Zheng, H.T., Liu, Z.: Few-nerd: A few-shot named entity recognition dataset. In: ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference (2021). https://doi.org/10.18653/v1/2021.acl-long.248
10. Gemini Team et al: Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context (2024), `https://arxiv.org/abs/2403.05530`
11. Gemini Team et al: Gemini: A family of highly capable multimodal models (2024), `https://arxiv.org/abs/2312.11805`
12. Han, Z., Gao, C., Liu, J., Zhang, J., Zhang, S.Q.: Parameter-efficient fine-tuning for large models: A comprehensive survey. arXiv preprint arXiv:2403.14608 (2024)
13. Hanks, C., Maiden, M., Ranade, P., Finin, T., Joshi, A., et al.: Recognizing and extracting cybersecurity entities from text. In: Workshop on Machine Learning for Cybersecurity, International Conference on Machine Learning (2022)
14. Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J.: Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300 (2020)
15. Jordan, B., Piazza, R., Darley, T.: Stix version 2.1. `https://docs.oasis-open.org/cti/stix/v2.1/os/stix-v2.1-os.html` (2021), [Accessed 28-02-2025]
16. Legoy, V., Caselli, M., Seifert, C., Peter, A.: Automated retrieval of att&ck tactics and techniques for cyber threat reports (2020)
17. Lim, S.K., Muis, A.O., Lu, W., Ong, C.H.: Malwaretextdb: A database for annotated malware articles. In: ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers). vol. 1 (2017). https://doi.org/10.18653/v1/P17-1143
18. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
19. Marchiori, F., Conti, M., Verde, N.V.: Stixnet: A novel and modular solution for extracting all stix objects in cti reports. In: ACM International Conference Proceeding Series (2023). https://doi.org/10.1145/3600160.3600182
20. Peng, B., Chersoni, E., Hsu, Y.Y., Huang, C.R.: Is domain adaptation worth your investment? comparing bert and finbert on financial tasks. In: Proceedings of the 3rd Workshop on Economics and Natural Language Processing, ECONLP 2021 (2021). https://doi.org/10.18653/v1/2021.econlp-1.5, domain adaptation shows to be good on biomedical domain

21. Rahman, M.R., Mahdavi-Hezaveh, R., Williams, L.: A literature review on mining cyberthreat intelligence from unstructured texts. In: IEEE International Conference on Data Mining Workshops, ICDMW. vol. 2020-November (2020). https://doi.org/10.1109/ICDMW51313.2020.00075
22. Ranade, P., Piplai, A., Joshi, A., Finin, T.: Cybert: Contextualized embeddings for the cybersecurity domain. In: Proceedings - 2021 IEEE International Conference on Big Data, Big Data 2021 (2021). https://doi.org/10.1109/BigData52589.2021.9671824
23. Sauerwein, C., Pfohl, A.: Towards automated classification of attackers' ttps by combining nlp with ml techniques. arXiv preprint arXiv:2207.08478 (2022)
24. Siracusano, G., Sanvito, D., Gonzalez, R., Srinivasan, M., Kamatchi, S., Takahashi, W., Kawakita, M., Kakumaru, T., Bifulco, R.: Time for action: Automated analysis of cyber threat intelligence in the wild. arXiv preprint arXiv:2307.10214 (2023)
25. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune bert for text classification? In: China national conference on Chinese computational linguistics. pp. 194–206. Springer (2019)
26. Tikhomirov, M., Loukachevitch, N., Sirotina, A., Dobrov, B.: Using bert and augmentation in named entity recognition for cybersecurity domain. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 12089 LNCS (2020). https://doi.org/10.1007/978-3-030-51310-8_2
27. Tual, S., Abadie, N., Chazalon, J., Duménieu, B., Carlinet, E.: A benchmark of nested named entity recognition approaches in historical structured documents. In: International Conference on Document Analysis and Recognition. pp. 115–131. Springer (2023)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Łukasz Kaiser, Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. vol. 2017-December (2017)
29. Wang, G., Liu, P., Huang, J., Bin, H., Wang, X., Zhu, H.: Knowcti: Knowledge-based cyber threat intelligence entity and relation extraction. Computers & Security **141**, 103824 (2024)
30. Wang, X., He, S., Xiong, Z., Wei, X., Jiang, Z., Chen, S., Jiang, J.: Aptner: A specific dataset for ner missions in cyber threat intelligence field. In: 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2022 (2022). https://doi.org/10.1109/CSCWD54268.2022.9776031
31. Wang, X., Liu, X., Ao, S., Li, N., Jiang, Z., Xu, Z., Xiong, Z., Xiong, M., Zhang, X.: Dnrti: A large-scale dataset for named entity recognition in threat intelligence. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). pp. 1842–1848. IEEE (2020)
32. Yi, F., Jiang, B., Wang, L., Wu, J.: Cybersecurity named entity recognition using multi-modal ensemble learning. IEEE Access **8** (2020). https://doi.org/10.1109/ACCESS.2020.2984582
33. You, Y., Jiang, J., Jiang, Z., Yang, P., Liu, B., Feng, H., Wang, X., Li, N.: Tim: threat context-enhanced ttp intelligence mining on unstructured threat data. Cybersecurity **5** (2022). https://doi.org/10.1186/s42400-021-00106-5
34. Zanella, L., Toussaint, Y.: Adding linguistic information to transformer models improves biomedical event detection? In: 2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS). pp. 1211–1216. IEEE (2023)
35. Zhao, J., Yan, Q., Li, J., Shao, M., He, Z., Li, B.: TIMiner: Automatically extracting and analyzing categorized cyber threat intelligence from social data. Computers and Security **95** (2020). https://doi.org/10.1016/j.cose.2020.101867