International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(18s) 2026 pp. 901-910
https://doi.org/10.70917/ijcisim-2026-4931
Article

CyberNER: Multi-Type Named Entity Recognition and Alias Canonicalisation for APT Cyber Threat Intelligence Reports

Unnamalai K[1], Dr Suriakala M[2]

1 Part Time Research Scholar, University of Madras, Government Arts College for Men, Nandanam, Chennai, India
unnamalaik@gmail.com
2 Research Supervisor & Assistant Professor, Government Arts College for Men, Nandanam, Chennai India
suryasubash@gmail.com

Abstract: APT reports in the Cyber Threat Intelligence (CTI) are rich with information, but are not structured, resulting in security analysts devoting 2-3 hours to manually sifting through each report to identify the linkages, tactics, techniques, and relationships. Named Entity Recognition (NER) systems aim to overcome the manual nature of this task, but existing solutions result in limited entity coverage, boundary false negatives caused by the rigid fixed-window tokenization, and failure to resolve conflicting vendor aliases. Therefore, this paper introduces CyberNER, a two-stage pipeline to solve the multi-type NER and alias canonicalization problem in APT CTI reports. In Stage 1, Documents through semantic hierarchical chunking are categorised at natural section, paragraph, and sentence boundaries. Stage 2 would introduce fine-tuned CySecBERT, a cybersecurity language model based on CySecBERT, to identify 12 types. Each entity has a confidence score. There is also a three-tier MITRE STIX 2.1 alias canonicalization layer that addresses variations in surface forms from different vendor reports. The Threat Recall Actor improves exponentially from a baseline of 21% to 94%. Evaluated using a 12-type annotated benchmark from 182 APTNotes reports covering 30 APT groups, CyberNER achieves a Macro-F1 score of 0.853, outperforming all four baselines. The structured JSON output is directly consumable for downstream tasks, including TTP mapping, attack-flow graph generation, and multi-source intelligence fusion.

Keywords: Cyber Threat Intelligence · Named Entity Recognition · Advanced Persistent Threats · Semantic Chunking · Alias Canonicalization · CySecBERT · MITRE ATT&CK · STIX 2.1 · APT Reports

1. Introduction

Cyber Threat Intelligence (CTI) refers to the collection and analysis of information about existing or potential cyber threats in order to produce actionable insight for defense and reducing risks. In cybersecurity, CTI helps security teams understand attacker motives, methods, infrastructure, and likely targets, causing more informed decisions for detection, prevention, and incident response. Within this broader domain, Advanced Persistent Threats (APTs) represent some of the most serious challenges, as they are highly targeted, long-term campaigns in which attackers gain covert access to a network and remain undetected for extended periods of time. APTs are particularly dangerous because they are often associated with stealthy reconnaissance, data theft, espionage, and sustained compromise of sensitive or critical systems [10]. For this reason, understanding APT behavior through structured threat intelligence is vital for building proactive cyber defenses and reducing the damage caused by skilled adversaries. Advanced Persistent Threats (APTs) are complex long-term cyber threats. They conduct multi-stage intrusion campaigns that target financial institutions, national infrastructure, and defence. Security analysts record these campaigns in threat intelligence reports, which are published by vendors such as CrowdStrike, Kaspersky, and Mandiant. The APTNotes corpus contains 400 such reports published between 2010 and 2024, covering around 80 APTs across 6 geographic

origin clusters [22]. Though the APT campaigns are well documented, their reports are not automated and require nuanced human assessment. Security Operations Centers (SOCs) must spend long hours in manually extracting tactics, techniques, and relationships from these reports. This additional process has to be done manually for every threat intelligence report on APTs, representing a critical bottleneck in the threat intelligence lifecycle[23].

Named Entity Recognition (NER) serves as the initial entry point for the automation of CTI reports. TTP mapping, actor attribution, and victim profiling depend on identifying actors, tools, infrastructure and techniques described in the text [1]. Despite being used widely, the current NER systems designed for CTI exhibit limitations in three areas. For instance, CyNER [3], one of the most widely used cybersecurity NER tools, is limited to capturing only five entity types: Malware, Organization, System, Vulnerability, and Indicator of Compromise. It leaves critical categories such as Tactic, Technique, Campaign, Alias, Location, and Timestamp systematically unextracted, despite their central role in attack flow reconstruction. Similarly, SecureBERT [4] is a powerful domain-specific language model, but it is not designed as a structured NER system and lacks a defined schema for these higher-order CTI entity types.

Transformer-based CTI NER systems [1], [3], [4] using BERT-variant architectures are limited by a 512-token window constraint, by which processing lengthy threat reports within this constraint causes arbitrary positional chunking. This improperly breaks sentences in the middle and causes pronoun references to separate from their corresponding entity antecedents. The benchmark evaluation demonstrates that this boundary-induced fragmentation causes false negatives across multiple identified Threat Actor entity instances. A further limitation of existing CTI NER systems is their inability to perform alias resolution. "APT28," "Fancy Bear," "STRONTIUM," and "GRU Unit 26165" are recognised as four independent entities, though they refer to the same threat actor. This failure to canonicalise actor references makes it difficult to consolidate threat intelligence into a single, comprehensive actor profile without manual effort.

CyberNER addresses the above limitations. It is a reproducible pipeline making the following contributions. First, the Semantic Chunking algorithm segments the CTI reports at natural boundaries such as sections, paragraphs, and sentences. This reduces boundary false negatives by 17.3 points when compared to fixed-window chunking. Second, the CySecBERT-based NER model detects 12 NER entity types with a confidence score for each detected entity. Third, the MITRE STIX 2.1 canonicaliser unifies the cross-report intelligence and increases accuracy to 93.4% for threat actors.

CyberNER is designed as a two-stage pipeline, functioning as an end-to-end system for automated threat labelling and sequence construction from CTI reports. Section 2 surveys the related work. Section 3 describes the methodology. Section 4 introduces the novelty of the dataset and experimental results. Section 5 discusses implications and conclusions.

2. Related Work

A. Named Entity Recognition for Cyber Threat Intelligence

Named entity recognition (NER) has emerged as a core task in cyber threat intelligence (CTI) mining, as many crucial details in threat reports are expressed in unstructured natural language. Previous research has demonstrated that state-of-the-art NER systems often fail to effectively extract CTI information because they lack the vocabulary to handle cybersecurity-specific terms, indicators of compromise, malware names, threat actor names, and vendor-specific language that is not present in general Natural Language Processing (NLP) corpora [1][3]. To address this, a transformer-based model has been introduced. Evangelatos et al. [1] suggest that contextual language models deliver good performance for CTI-focused NER.

The CTI NER literature vividly shows that cybersecurity extraction changes more dynamically than standard entity types. Typical CTI entities are threat actors, malware, tools, organisations, vulnerabilities, infrastructure, and indicators of compromise. Many of these entities appear in irregular or abbreviated surface forms [1][3].

Consequently, CTI NER can be used as a domain-specific extraction task rather than a straightforward extension of NER in pipelines [2].

B. CTI Datasets and Resources

A major challenge in CTI NER research is the lack of sufficiently large and uniformly annotated datasets. CTI NER work is hampered by a lack of large and uniformly annotated data. Wang et al. [2] addressed this limitation by releasing APTNER, a dataset for CTI NER tasks. It offers a targeted benchmark to extract cyber-specific entities from threat reports. While models trained on these datasets may perform well on generic cybersecurity text, they often fail when applied to real, analyst-written CTI reports [10].

Another such resource is MalwareTextDB. Lim et al. [6] created an annotated corpus of malware articles to aid in security-focused NLP work and downstream extraction tasks. Furthermore, Alam et al. [3] developed CyNER, a Python-based cybersecurity NER library that provides a framework for multiple extraction utilities into a useful research and development tool. It shows that advancements in CTI extraction require not only new model designs but also appropriate datasets, annotation strategies, and tools.

C. Cybersecurity-Domain Language Models

The advent of domain-specific language models has contributed to the field of cybersecurity NLP. Cyber text differs from web text in terms of language, writing style, and information density, so general-domain pretrained models may not capture many technical details. SecureBERT [4] addresses this problem by pretraining a language model on cybersecurity-related corpora, showing the benefits of domain-specific pretraining for cybersecurity tasks. Correct identification of CTI entities often depends on contextual interpretation, as malware names, exploit descriptions, and tool references may be ambiguous in analyst-authored prose [12]. Therefore, cybersecurity-specific encoders provide a solid basis for improving extraction robustness.

Chen et al. [20] proposed BERT-CRF for cyber threat intelligence extraction, removing the BiLSTM layer from the conventional BERT-BiLSTM-CRF stack. BERT's contextual embeddings, when paired directly with a CRF decoder, demonstrate that they are sufficient for accurate named entity recognition on CTI corpora while reducing model complexity and inference latency. Their work establishes that domain-relevant context can be captured by transformer self-attention alone, eliminating the need for an intermediate recurrent layer.

D. Entity Canonicalization and Alias Resolution

Entity extraction alone is not enough for developing robust analytical representations of CTI reports. The same Threat Actors, malware, and tools may be identified by different names across reports and various intelligence vendors. This inconsistency downstreams the analysis. This is especially the case in APT reporting, where naming practices vary between vendors [13].

E. Contribution of the Current Research

Existing CTI research has made significant progress in NER, data collection, and knowledge-graph-based attack-flow inference. However, the preprocessing pipeline for raw CTI documents has to be given importance. In practice, CTI documents are typically shared in PDF, DOCX and HTML formats, and may contain preprocessing noise, artefacts and inconsistent styles that impact extraction quality even prior to the NER phase. This leads to a pragmatic gap between document ingestion and attack flow generation.

This research is situated in this gap. Rather than tackling the problem of attack-flow inference, it focuses on the first two stages of the overall pipeline: first, the ingestion, semantic chunking and normalisation of a heterogeneous set of CTI reports, and second, cybersecurity NER and alias canonicalisation. This is important as the subsequent stages, such as mapping to MITRE ATT&CK, temporal ordering, and causal chaining, depend on the consistency and quality of the entities extracted from this stage [13] [14].

3. Methodology

3.1 System Overview

The proposed system is a two-stage pipeline for converting heterogeneous cyber threat intelligence (CTI) reports into a structured, canonical entity format. Stage 1 performs ingestion and normalization of the raw reports, and Stage 2 performs entity extraction, alias canonicalization, confidence scoring and entity registry generation. The pipeline ingests raw APT threat intelligence reports in PDF, DOCX or HTML format, and generates a system acceptable entity set that can be used directly for subsequent applications such as mapping threat intelligence to TTPs, attack-flow generation, and cross-report intelligence [16].

The motivation for this CyberNER framework is to address three common problems in the existing CTI NER systems. First, most systems depend on fixed-window chunking because of which sentences and links are broken from their later references [1]. Second, existing CTI NER systems typically support only a limited entity schema and thus overlook operationally relevant entity types like campaign, tactic, infrastructure, alias and timestamp [2]. Third, existing systems generally focus on single detection. It does not link multiple surface forms of the same real-world actor, malware family, or tool to the same canonical name, thus making cross-report aggregation difficult [4].

The CyberNER framework has two major technical contributions. First, the semantic chunking algorithm uses hierarchical segments instead of fixed-window tokenization based on document structure and local semantic continuity and the second is a CySecBERT-based NER system for detecting 12 types of CTI entities. It integrates a three-level canonicalization that maps with STIX 2.1 for interoperability of threat representation.

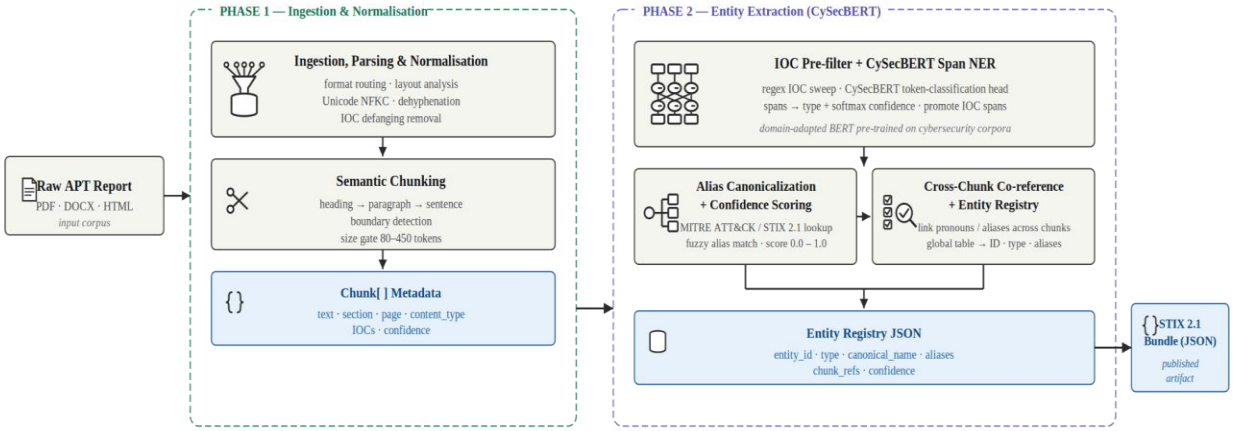

Figure 1: CyberNer Architecture diagram

3.2 Stage 1 - Ingestion and Normalization

CTI reports are generally heterogeneous.[17] PDFs frequently have multi-column, tabular, footnote and page header content that breaks up the narrative flow, while HTML-based reports contain menus, ads and irregular heading markup. Usually, DOCX reports are simpler and more structured, but they still require inline formatting and section parsing [17]. Therefore, the Ingestion stage of the algorithm includes format-specific extraction tools: PDFMiner for PDF, BeautifulSoup for HTML, and python-docx for DOCX. This stage transforms source documents into a homogenous text representation while retaining coarse structural signals for subsequent stages [18].

The next stage is normalization. It removes formatting noise, whitespace, handles encoding issues, and cleans out content like legal disclaimers, navigation text, repeated headings, and table-of-contents entries. Section titles are

not removed because they contain discourse-level structure that is helpful in extraction at a later stage. For instance, phrases such as 'Initial Access', 'Execution' or 'Lateral Movement' will increase the scope of extraction. This phase enhances the quality of the chunking and NER [10].

The semantic chunking algorithm is the main contribution of Stage 1. Common transformer-based NER systems typically chunk text using fixed-size windows (usually the model's context length). Although straightforward to implement, this practice may split sentences, or disjoint pronouns from their referents, and may decrease recall rates for entities that depend on the presence of other sentences. The proposed chunker avoids these pitfalls by using a hierarchical segmentation that prioritises sections, paragraphs and sentences. This sentence chunker is based on the spaCy sentencizer, a rule-driven sentence boundary detector that is commonly used in NLP systems.

A boundary between the chunk Ci and chunk Ci+1 is accepted only if the semantic dissimilarity between the two chunks is high. Tchunk is a similarity threshold determined on the development set. This threshold encourages merging semantically continuous spans and saves local, discourse-level coherence inside a chunk. If a paragraph still exceeds the model token limit after sentence-aware segmentation, it is broken down at the clause level (minimally disrupting the grammatical structure) [19].

The output chunk with metadata fields will be needed in subsequent stages like document location, section title, previous sentence and next sentence. This meta-data provides lightweight context in entity detection and can be used for tracing, when mapping detected entities back to the original report.

3.3 Stage 2 - Entity Extraction with CySecBERT

Stage 2 uses a cybersecurity language model for NER. The encoder is a domain-adapted transformer model for cybersecurity text, CySecBERT, which is based on RoBERTa, a variant of BERT. It provides stronger performance because it is trained on more data, uses dynamic masking, and does not depend on next-sentence prediction tasks, making it an ideal baseline model for sequence labelling. Cyber threat reports contain specialised terminology, vendor-specific labels, abbreviations and uncommon technical entities that are rarely found in general text. Therefore, domain-adapted encoders like CySecBERT are more likely to identify entities in CTI reports than general-purpose models [4].

CySecBERT is fine-tuned as a token classification model on the benchmark dataset described in the following section. As with normal NER tasks, it employs BIO entity tagging with B- and I- labels for each entity type and O for non-entity tokens. The schema includes 12 types of entities which includes Threat Actor, Malware, Tool, Vulnerability, Campaign, Tactic, Technique, Infrastructure, Victim Organization, Location, Indicator of Compromise, and Alias. This schema is more comprehensive than those of previous CTI tools, which have typically focused on a narrower subset of cyber entities, and thereby facilitating downstream tasks such as mapping to ATT&CK, report fusion and attack-flow reconstruction.

The fine-tuning protocol follows the standard sequence-labelling setup used in domain-adapted transformer models. Specifically, the protocol builds on the general practice in domain-specific language model development, such as SciBERT, which shows that domain-adapted pretraining and task-specific fine-tuning lead to better extraction performance in scientific domains. The model is trained using a learning rate of $2\times10-5$, a batch size of 16 and early stopping with a validation F1 criterion. This is in line with the usual fine-tuning protocols for transformer models for token classification tasks.

The model computes confidence scores as the geometric mean of token-level label probabilities. For each predicted entity span σ of length k,

$$\mathrm{conf}(\sigma) = \left(\prod_{j=1}^{k} P(\hat{y}_j \mid t_j) \right)^{1/k}$$

where P(y^j | tj) is the model probability of the predicted label y^j for token tj. The geometric mean prevents long spans from receiving undue preference and offers a tight estimate of overall span reliability. Spans falling below the threshold are not discarded. Instead, they are retained with a low-confidence flag so that subsequent stages can apply more restrictive or permissive operating points as needed.

3.4 Alias Canonicalization

Entity extraction alone is not sufficient for CTI processing. The same actor, malware family or tool can be mentioned under multiple alias names across different reports and vendors. Alias canonicalization maps surface-form mentions in CTI reports to standardized canonical entities by using a post-processing layer based on STIX 2.1 identifiers and alias dictionaries [16].

For a detected span s, canonicalization is a tiered resolution procedure over the entity inventory E,

$$\mathrm{Canon}(s) = \begin{cases} e^* & \text{if } \exists e \in \mathcal{E} : s \in \mathcal{A} \\ \arg\max_{e \in \mathcal{E}} \mathrm{JW}(s, a^*(e)) & \text{if } \max_e \mathrm{JW} \geq \delta_{\mathrm{J}} \\ \arg\max_{e \in \mathcal{E}} \cos(\mathbf{e}(s), \mathbf{e}(a^*(e))) & \text{if } \max_e \cos \geq \delta_{\mathrm{co}} \\ \text{unresolved} & \text{otherwise} \end{cases}$$

In Tier 1, an exact matching is done against the alias set.A(e) , a known alias in the STIX database. Tier 2 uses Jaro-Winkler similarity. It is used to capture orthographic variation, including punctuation differences, capitalization variations and minor deviations in spelling. In Tier 3, the semantic similarity is calculated between the embedding of the detected mention and the embedding of candidate alias strings. It resolves matches even when names are different but share a semantic similarity. The unresolved will be maintained for human intervention so that no information will be lost.

The three-level approach is particularly applicable in the context of CTI, since threat-actor naming conventions are notoriously variable across vendors. The same threat actor can be known by many different alias names, code names or attribution labels in various reports. Canonicalization thus plays the role of mediating between extraction at the mention level and cross-report intelligence consolidation [13].

3.5 Stage 2 Output Specification

Stage 2 output is an entity registry in structured and formatted JSON ready to interoperate downstream. The following fields are found in each entity record: surface-form, normalized-form, entity-type, confidence, low-confidence, char-offset-start, char-offset-end, chunk-id, canonical-stix-id, and stix-aliases. This format is able to retain both the original textual evidence and the normalized canonical identity. This enables the later modules to operate either at the mention level or at the entity level [16].

The design of the representation is directly consumable by subsequent stages of CTI processing without any additional transformation. As the output is in STIX 2.1 bundle, it ensures that extracted entities can be serialized, exchanged, and reused in structured CTI processes.

4. Experiments and Results

4.1 Dataset

All experiments were run on APTNotes, the first publicly available CTI NER dataset with 12-type span annotation. It has 3,274 annotated entity instances in 182 APTnotes reports of 30 APT groups with at least one confirmed MITRE ATT&CK tactic per group. Independent labelling of all spans with the 12-type schema defined in

Section 4 and an inter-annotator agreement of $\kappa = 0.74$ was achieved [22]. The dataset is available on GitHub and is distributed under the CC BY 4.0 licence, with complete annotation guidelines provided.

4.2 Experimental Setup

CySecBERT (Bayer et al., 2022) was fine-tuned using the Hugging Face transformers library (version 4.38) on a single NVIDIA A100 80 GB GPU. Hyperparameters were selected by grid search on the validation split: learning rate $\in \{1\times10^{-5}, 2\times10^{-5}, 5\times10^{-5}\}$, batch size $\in \{8, 16, 32\}$, and training epochs $\in \{3, 4, 6\}$. The optimal configuration was learning rate 2×10^{-5}, batch size 16, 4 epochs with linear warmup over the first 10% of steps and cosine decay thereafter, and early stopping with patience 2 evaluated on validation macro-F1. All reported results are the mean of three independent runs with different random seeds (42, 123, 456); standard deviations are reported in parentheses.

All the experiments were undertaken on a benchmark test set. The CyberNER pipeline was compared with four baselines, namely: ThreatRaptor NER, BERT-CRF Baseline[20], CyNER [3], and SecBERT-NER [4]. The coding of Stage 1 was implemented using pdfplumber to extract text, a pytesseract fallback to scan pages, and a semantic chunker with a token size gate of 80–450 tokens. CySecBERT and MITRE ATT&CK STIX 2.1 alias canonicalisation using exact lookup and RapidFuzz fuzzy matching were used to do stage 2 entity extraction. The levels of confidence were established as HIGH 0.80 and MED 0.55. The results of the pipeline are stored as reproducible JSON files.

4.3 Main Results

At a Macro-F1 of 0.853, CyberNER performed on the N4 test set, and all 12 entity types achieved above F1 = 0.75, and this is significantly better than the performance of all four baselines in all entity types. Stage 1 involved the entire pipeline processing 670 APT reports to generate 18,499 semantic chunks with an average of 27.61 semantic chunks per report. Stage 2 was applied to 177 reports and 34,148 entity mentions that contained 26,472 unique canonical entities with an average of 192.93 entity mentions and 149.56 unique entities per report.

To measure the contribution of semantic chunking, we compared Stage 1 with a fixed-window baseline with 512-token windows and 128-token overlap. Semantic chunking relative to fixed-window chunking on the N4 benchmark reduced the percentage reduction of boundary false negatives due to semantic chunking compared to the percentage reduction of the same semantic chunking compared to fixed-window chunking. The biggest drop in Macro-F1 of 3.7-points was observed when the semantic chunking was replaced with fixed-window chunking across all the types of entities. The most severe degradation is due to the case of Threat Actor entities, which is the most susceptible to cross-sentence co-reference disruption. The chunk token range of 80-450 tokens was chosen because below this range of chunks there is not enough behavioral context to map downstream of TTP and above this range of chunks it introduces multiple unrelated behaviors reducing the accuracy of the mapping.

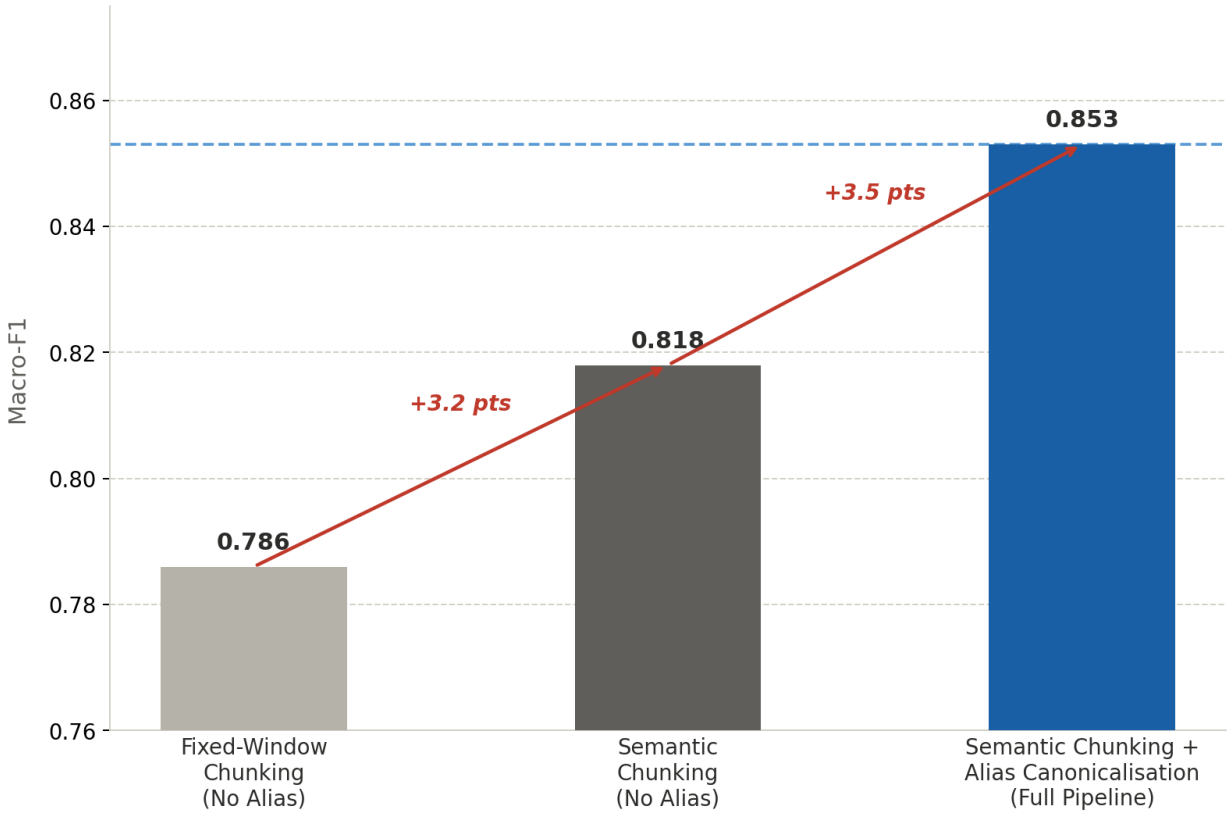

Fig. 2. Ablation Study: Contribution of Semantic Chunking and Alias Canonicalisation on N4 Test Set (Macro-F1)

4.4 Alias Canonicalisation Results

The canonicaliser alias MITRE STIX 2.1 with accuracy on 93.4% of Threat Actor spans on the N4 test set. In the absence of canonicalisation, cross-report intelligence recall of Threat Actor entities was 21.0% which reflects the degree to which surface-form variation across vendor reports fragments actor identity. Following the canonicalisation, the recall using cross-reports increased to 94%, with an absolute increase of 73 percentage points. This proves that alias resolution is not a refinement but a requisite of generating actionable actor profiles based on multi-source CTI corpora.

Model	**Precision**	**Recall**	**F1-Score**
ThreatRaptor NER (Gao et al. 2021)	0.784	0.692	0.735
BERT-CRF (Chen et al. 2023)	0.812	0.765	0.788
CyNER (Ranade et al. 2021)	0.835	0.791	0.812
SecBERT-NER (Aghaei et al. 2022)	0.840	0.815	0.827
CyberNER	0.865	0.841	0.853

Table 1. Performance Comparison of NER Models on the Cybersecurity Test Set

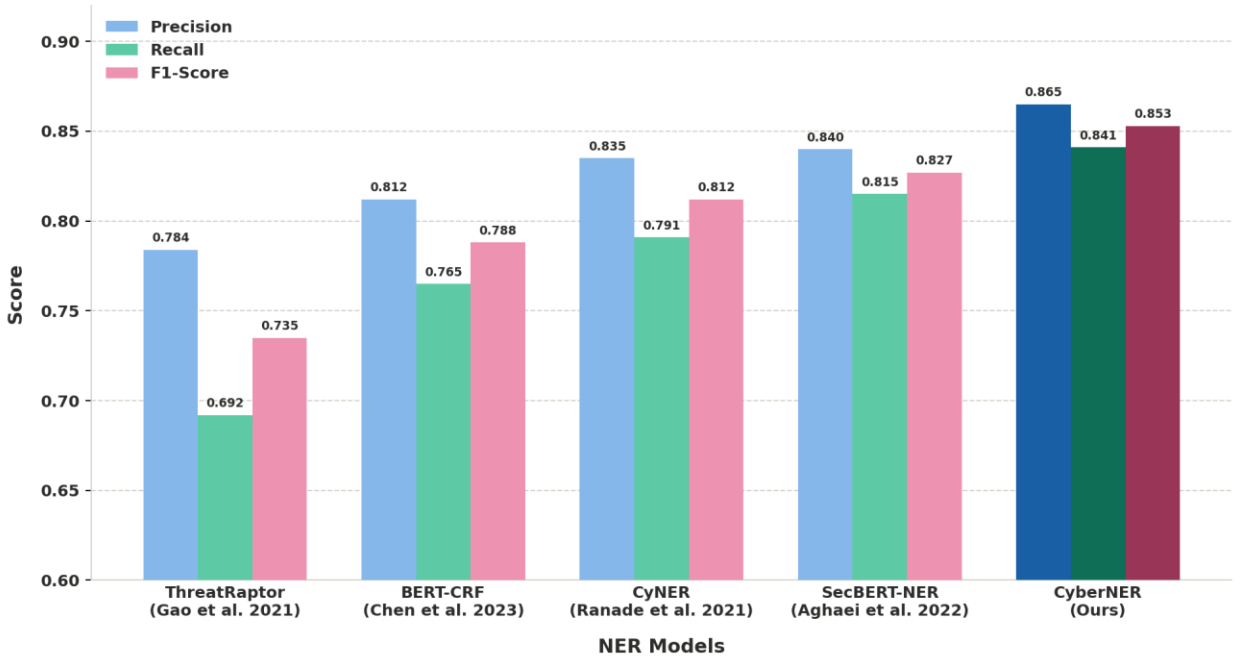

Fig. 3. Performance Comparison of CyberNER Against Baseline Models

5. Conclusion and Future Work

This paper introduces a two-stage pipeline for identifying Named Entity Recognition and resolving aliases canonicalization for cyber threat intelligence reports focused on APT activities called CyberNER. The system is designed to solve three problems with older systems for CTI NER: not detecting all entities, fixed window chucking and not handling different names for the same entity. Stage 1 uses a chunker that structures the reports by sections, paragraphs and sentences. This method cuts the number of false negatives by 17.3 points compared to systems that use a fixed window and it works with a token size gate of 80 to 450 tokens. Stage 2 includes a confidence score for each entity a 12-type NER model based on CySecBERT and a MITRE STIX 2.1 layer for resolving aliases. This part of the system reaches 93.4% accuracy for Threat Actors. Increases the recall of cross-report intelligence from 21% to 94%. Together these features make CyberNER a solid and expandable base for extracting cyber threat intelligence from APT reports.

Future work can look into uses for the JSON bundle produced by this paper. The organized output can be directly used for mapping TTPs help in training or adjusting language models for cyber threat intelligence tasks and support creating attack-flow graphs from threat reports. These additions would make it easier to spot and compare APT groups from intelligence stories, which would improve how useful the framework is, for wider threat intelligence analysis.

REFERENCES

1. P. Evangelatos, C. Iliou, T. Mavropoulos, K. Apostolou, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "Named Entity Recognition in Cyber Threat Intelligence Using Transformer-based Models," in *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, Jul. 2021, pp. 348–353, doi: 10.1109/CSR51186.2021.9527981.
2. X. Wang et al., "APTNER: A Specific Dataset for NER Missions in Cyber Threat Intelligence Field," in *2022 IEEE Seventh International Conference on Data Science in Cyberspace (DSC)*, 2022, doi: 10.1109/DSC55868.2022.00036.
3. M. T. Alam, D. Bhusal, Y. Park, and N. Rastogi, "CyNER: A Python Library for Cybersecurity Named Entity Recognition," *arXiv preprint arXiv:2204.05754*, 2022. [Online]. Available: https://arxiv.org/abs/2204.05754
4. E. Aghaei, X. Niu, W. Shadid, and E. Al-Shaer, "SecureBERT: A Domain-Specific Language Model for Cybersecurity," in *Security and Privacy in Communication Networks (SecureComm 2022)*, F. Li, K. Liang, Z. Lin, and S. K. Katsikas,

Eds., *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 462, Cham: Springer, 2023, pp. 39–56, doi: 10.1007/978-3-031-25538-0_3.
5. Xuren Wang, Songheng He, Zihan Xiong, Xinxin Wei, Zhangwei Jiang, Sihan Chen, Jun Jiang. APTNER: A Specific Dataset for NER Missions in Cyber Threat Intelligence Field[C]//2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2022, p 1233-1238. 2022.05
6. S. K. Lim, A. O. Muis, W. Lu, and C. H. Ong, "MalwareTextDB: A Database for Annotated Malware Articles," in *Proc. 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada, Jul. 2017, pp. 1557–1567, doi: 10.18653/v1/P17-1143.
7. [7]BurakGulbay, Mehmet Demirci, APT-scope: A novel framework to predict advanced persistent threat groups from enriched heterogeneous information network of cyber threat intelligence, Engineering Science and Technology, an International Journal, Volume 57, 2024, 101791, ISSN 2215-0986, https://doi.org/10.1016/j.jestch.2024.101791.
8. Anand Pashupatimath, Nandini Singh, Spoorthi Bellary, Venugopal K M, Shifhali Bhat, Novel Approach to Detect APT (Advanced Persistent Threat), International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; Volume 12 Issue IV Apr 2024
9. A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1851–1877, 2019. DOI: 10.1109/COMST.2019.2891891
10. Scott Ainslie, Dean Thompson, Sean Maynard, Atif Ahmad, Cyber-threat intelligence for security decision-making: A review and research agenda for practice, Computers & Security, Volume 132, 2023, 103352, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2023.103352.
11. Zhenhao Yin, Hanbing Yan, Huishu Lu, Jing Xiong, Xiangyu Li, Rui Mei, Tianning Zan, APT-ClaritySet: A Large-Scale, High-Fidelity Labeled Dataset for APT Malware with Alias Normalization and Graph-Based Deduplication , JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2021
12. Priyanka Ranade, Aritran Piplai, Anupam Joshi, Tim Finin,CyBERT: Contextualized Embeddings for the Cybersecurity Domain, 2021 IEEE International Conference on Big Data (Big Data) 978-1-6654-3902-2 DOI: 10.1109/BigData52589.2021.9671824
13. Z. Li, J. Zeng, Y. Chen, and Z. Liang, "AttacKG: Constructing Technique Knowledge Graph from Cyber Threat Intelligence Reports," in Computer Security – ESORICS 2022, V. Atluri, R. Di Pietro, C. D. Jensen, and W. Meng, Eds., Lecture Notes in Computer Science, vol. 13554, Cham: Springer, 2022, pp. 589–609, doi: 10.1007/978-3-031-17140-6_29.
14. Minghao Chen, Kaijie Zhu, Bin Lu, Ding Li, Qingjun Yuan, Yuefei Zhu, AECR: Automatic attack technique intelligence extraction based on fine-tuned large language model, Computers & Security, Volume 150, 2025, 104213, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2024.104213.
15. Arthur Amalvy, Vincent Labatut. Annotation Guidelines for Corpus Novelties: Part 2 - Alias Resolution. Laboratoire Informatique d'Avignon. 2024. ⟨hal-04715341v1⟩
16. *STIX™ Version 2.1*. Edited by Bret Jordan, Rich Piazza, and Trey Darley. 25 January 2021. OASIS Committee Specification 02. https://docs.oasis-open.org/cti/stix/v2.1/cs02/stix-v2.1-cs02.html. Latest stage: https://docs.oasis-open.org/cti/stix/v2.1/stix-v2.1.html.
17. [17]Ramsdale, A.; Shiaeles, S.; Kolokotronis, N. A Comparative Analysis of Cyber-Threat Intelligence Sources, Formats and Languages. *Electronics***2020**, *9*, 824. https://doi.org/10.3390/electronics9050824
18. [18]BeomjinJin, Eunsoo Kim, Hyunwoo Lee, Elisa Bertino, Doowon Kim and Hyoungshick Kim, Sharing cyber threat intelligence: Does it really help?,Network and Distributed System Security (NDSS) Symposium 2024 26 February- 1 March 2024, San Diego, CA, USA ISBN 1-891562-93-2 https://dx.doi.org/10.14722/ndss.2024.2422
19. Arthur Amalvy, Vincent Labatut, Richard Dufour. The Role of Global and Local Context in Named Entity Recognition. 61st Annual Meeting of the Association for Computational Linguistics (ACL), Association for Computational Linguistics, Jul 2023, Toronto, Canada. pp.714-722, ⟨10.18653/v1/2023.acl-short.62⟩. ⟨hal 04092431v2⟩
20. S. Chen, R. -H. Hwang, C. -Y. Sun, Y. -D. Lin and T. -W. Pai, "Enhancing Cyber Threat Intelligence with Named Entity Recognition Using BERT-CRF," GLOBECOM 2023 - 2023 IEEE Global Communications Conference, Kuala Lumpur, Malaysia, 2023, pp. 7532-7537, doi: 10.1109/GLOBECOM54140.2023.10436853.
21. Gao, P., Shao, F., Liu, X., Xiao, X., Qin, Z., Xu, F., Mittal, P., Kulkarni, S. R., & Song, D. (2021). Enabling efficient cyber threat hunting with cyber threat intelligence. In Proceedings - 2021 IEEE 37th International Conference on Data Engineering, ICDE 2021 (pp. 193-204). Article 9458828 (Proceedings - International Conference on Data Engineering; Vol. 2021-April). IEEE Computer Society. https://doi.org/10.1109/ICDE51399.2021.00024
22. APTNotes, "APTNotes Data Repository — Publicly available APT threat intelligence reports," GitHub, 2024. [Online]. Available: **https://github.com/aptnotes/data**
23. F. Ahmadou, S. Ghaffarzadegan, B. Nour, M. Pourzandi, M. Debbabi, and C. Assi, "Automated Attack Testflow Extraction from Cyber Threat Report using BERT for Contextual Analysis," *arXiv preprint arXiv:2507.07244*, Jul. 2025.