AC_MAPPER

a robust approach to ATT&CK technique classification using input augmentation and class rebalancing

Albarrak, Majed; Alqudhaibi, Adel; Jagtap, Sandeep

Published in:
International Journal of Information Security

DOI:
10.1007/s10207-025-01146-5

2025

Document Version:
Publisher's PDF, also known as Version of record

International Journal of Information Security (2025) 24:232
https://doi.org/10.1007/s10207-025-01146-5

REGULAR CONTRIBUTION

AC_MAPPER: a robust approach to ATT&CK technique classification using input augmentation and class rebalancing

Majed Albarrak[1] · **Adel Alqudhaibi**[1,2] · **Sandeep Jagtap**[1,3]

Received: 8 August 2025 / Accepted: 20 October 2025

Abstract

The detection and classification of adversarial techniques from cyber threat intelligence (CTI) text is a critical task in threat analysis and mitigation. While recent transformer-based models have shown promise, their general-purpose nature often limits effectiveness on complex, domain-specific datasets. In this paper, we present a novel model designed to address the challenges of technique classification across heterogeneous CTI datasets. The proposed method is evaluated against several baselines, including CTI-specific models as well as general-purpose transformers like SciBERT and DistilBERT. The proposed approach "AC_MAPPER" consistently outperforms all baselines in both Accuracy and F1 scores across five benchmark datasets, achieving up to **93.59%** accuracy and **93.78%** macro F1 on the TRAM Bootstrap dataset. It also demonstrates superior robustness on highly imbalanced and sparse datasets such as HALdata and CAPEC, where baseline models struggle. Comprehensive performance comparisons, highlights the effectiveness of proposed approach. These results underscore the potential of integrating domain-specific design with transformer architectures to advance automated CTI analysis. Our findings contribute toward more accurate and reliable threat detection systems in real-world security applications.

Keywords Cyber threat intelligence (CTI) · MITRE ATT&CK framework · Natural language processing (NLP) · Large language models (LLMs) · Data augmentation

1 Introduction

Cyberattacks continue to escalate in both frequency and severity; the Ponemon Institute and IBM Security report that the average cost of a data breach reached a record USD 4.45 million in 2023—an increase of 2.3% over the prior year [1]. In response to this expanding threat landscape, researchers are advancing technical countermeasures, including deep learning [2] and collaborative approaches. For example, [3]introduce a blockchain-enabled federated-learning framework to enhance IoT threat intelligence, while [4] describe a complementary design that couples machine learning with blockchain. Building on these developments, effective defence increasingly depends on Cyber Threat Intelligence (CTI)—a proactive discipline that collects, analyzes, and enriches threat information to harden organizational security controls [5, 6]. While CTI is an essential component of modern cybersecurity, a significant challenge lies in the extraction of actionable intelligence from the vast amount of unstructured data available in sources such as technical reports, security blogs, and social media. Analysts often manually process this information, a task that is time-consuming, expensive, and often fails to keep pace with the real-time nature of emerging threats [7]. This limitation highlights the urgent need for automated methods to effectively analyse and interpret textual cyber threat intelligence (CTI) data. Fortunately, the use of Natural Language Processing (NLP) has shown promise in automating CTI tasks, particularly in extracting valuable information from unstructured text [7, 8]. Moreover, MITRE ATT&CK framework is a globally recognised knowledge base of adversary tactics and techniques, which provides a standardised language for threat intelligence, and that makes it an ideal target for automation [9]. While more recent neural network models and transformer architectures have significantly improved NLP capabilities,

✉ Sandeep Jagtap
sandeep.jagtap@tlog.lth.se

[1] Sustainable Manufacturing Systems Centre, Cranfield University, Cranfield, UK

[2] Ministry of Defence, Riyadh, Saudi Arabia

[3] Division of Engineering Logistics, Lund University, Lund, Sweden

Published online: 07 November 2025

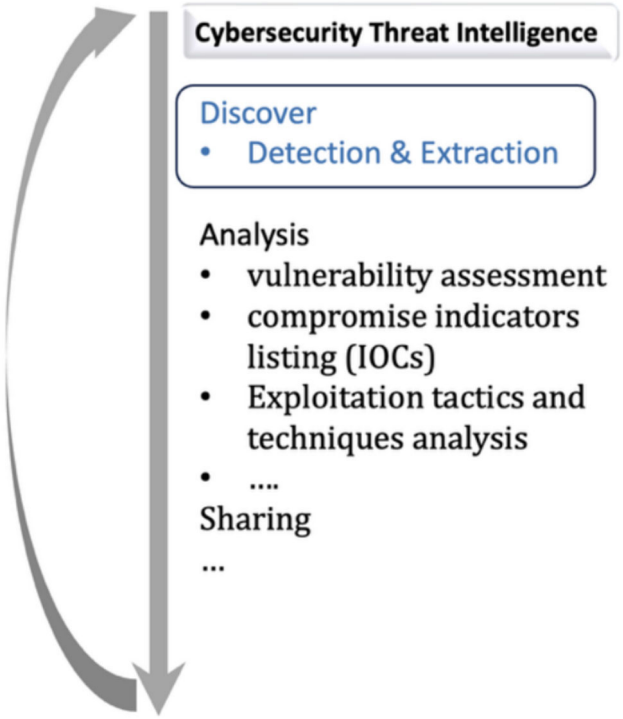

Fig. 1 Cyber threat intelligence process

their application to the specific task of automatically mapping threat intelligence to standardised frameworks remains an active area of research [7].

The remainder of this introduction provides an overview of Cyber Threat Intelligence (CTI), the MITRE ATT&CK Framework, and relevant Natural Language Processing (NLP) solutions.

1.1 Cyber threat intelligence (CTI)

CTI is the process of collecting, analysing, and enriching cybersecurity knowledge about threats to strengthen an organisation's defence [7]. This process involves many tasks such as extraction, sharing, vulnerability assessment, compromise indicators listing (IOCs), threat actor profiling, malware categorisation, and analysis of exploitation tactics and techniques [6, 8]. The priority and importance of CTI tasks differ by context, but the extraction task is fundamental for all subsequent processes. It is the first step of CTI and involves gathering the required information via sharing platforms, experts, or automated solutions such as NLP-based methods [7], Fig. 1.

There are several sharing platforms used to exchange information among cybersecurity professionals and organizations, such as Malware Information Sharing Platform (MISP) and Anomali. Additionally, standardized protocols like Structured Threat Information eXpression (STIX) and Trusted Automated eXchange of Indicator Information (TAXII) facilitate accurate and efficient information exchange [10]. However, much crucial threat intelligence remains in unstructured formats such as technical reports and social media posts [7]. Analysts often manually extract and correlate information from these sources, a process that is costly and time-consuming. Although social networks provide real-time insights on emerging threats, they require advanced techniques to detect relevant discussions. Fortunately, Natural Language Processing (NLP) has proven effective in enhancing textual data interpretation [8]. NLP techniques have shown efficiency in areas such as malware analysis, anomaly detection, and CTI [7, 8].

1.2 MITRE ATT&CK framework

The MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) framework is a globally accessible knowledge base that captures adversary behaviour across various stages of an attack lifecycle [9]. It was introduced by MITRE Corporation in 2013 as part of an internal research project called Fort Meade Experiment (FMX), which aimed to document and categorise the behaviour of Advanced Persistent Threats (APTs) based on real-world observations [11].

ATT&CK is structured around tactics (the adversary's technical goals), techniques (how they achieve those goals), and sub-techniques (refinements of techniques), mapped across different platforms such as Windows, Linux, macOS, cloud, and mobile environments [12]. The framework has become a foundational resource for red and blue teams, threat intelligence analysts, and cybersecurity researchers alike due to its standardised language and comprehensive documentation of adversarial techniques [13]. Figure 2 shows how Bad Rabbit ransomware is analysed in the MITRE ATT&CK® for Industrial Control Systems framework [14].

Security Operations Centres (SOCs) and threat hunters leverage ATT&CK to enhance detection, incident response, and threat modelling efforts. By aligning their detection mechanisms and alerts with ATT&CK techniques, SOCs can standardise how they identify and track threat actor behaviour over time [15]. Moreover, ATT&CK is integrated into many commercial and open-source tools for threat detection, simulation, and analytics, helping organisations evaluate coverage, identify detection gaps, and prioritise defences based on real-world threats [16]. The framework also plays a key role in cyber threat intelligence (CTI) enrichment by allowing analysts to map observed indicators, Tactics, Techniques and Procedures (TTPs), and campaign reports to a known set of adversarial behaviours, improving threat correlation and situational awareness across environments [17].

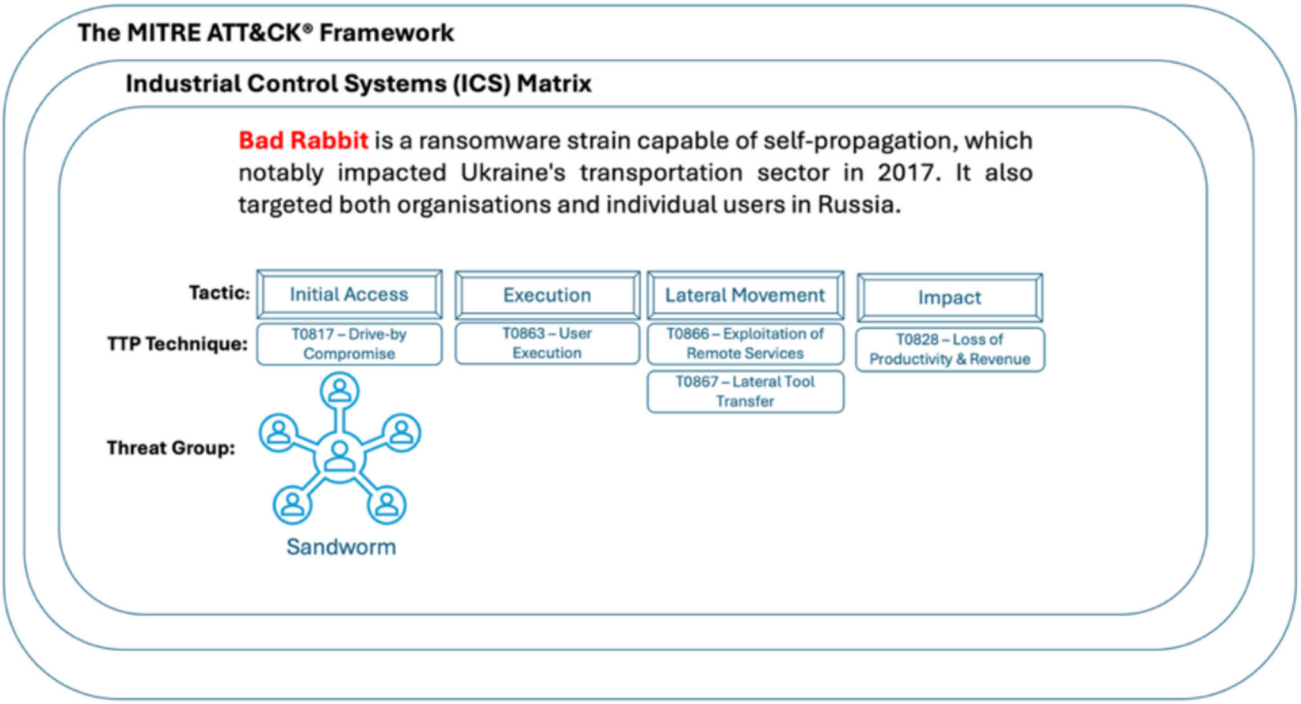

Fig. 2 Tactics, techniques and threat groups associated with the bad rabbit ransomware in the MITRE ATT&CK® for ICS framework [14]

1.3 Natural language processing (NLP) solutions

NLP refers to methods for improving machines' understanding of natural language [18]. It began with rule-based models in the 1980 s [19], followed by Statistical Language Models (SLMs) in the 1990 s [20]. These models used n-grams and Maximum Likelihood Estimation (MLE) to predict the next word in a sequence [21]. Later, neural networks and word embeddings moved NLP into the Neural Language Model (NLM) phase, delivering better performance [18].

1.4 Large language models (LLMs)

The idea of training models on large corpora has existed since the 1990 s, as in the ELIZA model [22, 23]. However, breakthroughs in computational power and neural network design have led to today's transformer-based LLMs [24]. These models are capable of performing human-like tasks, including text completion and translation. LLM training involves a two-phase approach: large-scale pre-training followed by fine-tuning on domain-specific tasks. Examples of pre-trained models include GPT-3, GPT-4 [25], BERT [26], and RoBERTa [27], and they can be customized for specific fields like medicine, law, or cybersecurity. In cybersecurity, some notable examples include:

- FalconLLM, designed for vulnerability detection in C code [28].
- SecureBERT, a language model tailored for cybersecurity threat intelligence [29].
- CySecBERT, also developed for threat intelligence tasks [30].

Therefore, transformer-based models are poised to significantly improve CTI capabilities for detecting ongoing cyberattacks, particularly in unstructured sources such as social media.

In this paper, we address the aforementioned gap by proposing a novel transformer-based approach for the automated extraction and categorization of cyber threat intelligence from unstructured text. The paper's primary objective is to develop a robust methodology that can accurately map descriptions of adversary behaviours directly to specific techniques within the MITRE ATT&CK framework.

The main contributions are as follows:

- A domain-specific Large Language Model (LLM) improved with a class-balanced augmentation strategy is presented to interpret the contextual nuances of cybersecurity reports and align them with the MITRE ATT&CK framework.
- A comprehensive evaluation of the proposed model is conducted to demonstrate its superior performance compared to existing baselines in accurately identifying and classifying MITRE ATT&CK techniques.

The rest of the paper is organised as follows. Section 2 provides a comprehensive literature review. Section 3 details the research methodology, followed by the implementation and the results in Sect. 4, which includes a discussion of the findings. Section 5 concludes the paper with a summary of the contributions and directions for future work.

2 Literature review

2.1 Rule-based and machine learning approaches

Early contributions primarily relied on rule-based and heuristic methods to extract Indicators Of Compromise (IOCs) and map threat intelligence, such as [31–34]. Machine learning and knowledge graph approaches have been widely adopted to automate the extraction and mapping of TTPs from CTI data, with various studies demonstrating success in areas such as multimodal learning for threat action extraction [35], building structured behavior graphs [36], leveraging BERT for TTP classification [37, 38], constructing knowledge graphs from CTI reports [39, 40], conducting large-scale experimental mapping studies [41], and addressing class imbalance in datasets [42]. Similarly, MITRE's TRAM project [43] initially comprised a hybrid rule-based system and an accompanying dataset, combining keyword heuristics with user feedback to map CTI text to ATT&CK tactics. In 2023, TRAM was extended with a pretrained domain-specific encoder (SciBERT [44]) and an expanded annotated dataset, improving contextual understanding and classification performance [43, 44]. In parallel with TTP extraction from CTI, Numerous studies have focused on automatically mapping Common Vulnerabilities and Exposures (CVEs) to MITRE ATT&CK techniques to enhance threat modeling and defense, employing various approaches such as heuristic rules and keyword sets [45], combined rule-based and NLP models [46], self-distillation deep learning, a game-theoretic framework [47], BERT-based semantic similarity [38, 48, 49], and semantic mapping frameworks [50, 51].

2.2 Large language models (LLMs) and transformer-based approaches

More recent research leverages large-scale transformer models and LLMs to address semantic ambiguity in CTI and improve mapping accuracy. Liu et al. [52] evaluated ChatGPT's effectiveness in mapping vulnerability descriptions to ATT&CK, revealing both promise and limitations in using general-purpose LLMs for domain-specific tasks. You and Park [53] proposed a two-stage LLMbased architecture specifically trained to classify cyberattack techniques. Li et al. [54] proposed a comprehensive framework for fully automated ATT&CK mapping using end-to-end deep learning applied to unstructured cyber threat reports, showing high generalizability across CTI domains. Several existing works have explored the use of general-purpose transformer-based language models, particularly BERT and RoBERTa, for cybersecurity-specific tasks before finetuning them into more domain-specific or task-oriented variants. In TTPHunter [55], the authors leveraged both BERT and RoBERTa to extract Tactics, Techniques, and Procedures (TTPs) from finished threat reports, showing improved accuracy through contextual embeddings. Similarly, TIM [56] applies BERT to enhance TTP mining by incorporating threat context from unstructured cyber threat data.

The Extractor framework [36] also relies on BERT for extracting attacker behavior from structured and unstructured threat reports, illustrating the model's strength in modelling semantic patterns across varying report formats. Kumarasinghe et al. [57] apply RoBERTa in a semantic ranking system for automated annotation of adversarial techniques, showing its effectiveness in identifying fine-grained TTP relationships. Furthermore, Alves et al. [37] fine-tune BERT to classify TTPs from unstructured CTI, demonstrating substantial gains over baseline classifiers. Lastly, Zhou et al. [58] incorporate BERT into the CTI View system to analyse APT-related threat intelligence, reinforcing BERT's utility in large-scale CTI applications. While many studies rely on general-purpose language models like BERT and RoBERTa for cybersecurity tasks, several domain-specific language models have been developed to overcome their limitations in handling cybersecurity-specific terminology and context. These specialised models have demonstrated improved performance in cyber threat intelligence (CTI) applications by leveraging pretraining on security-relevant corpora. Notable examples include SecureBERT [29], SecBERT [59], CyberBERT [60], and CTI-LM [61], all of which aim to generate more accurate contextual embeddings tailored to tasks such as threat detection, vulnerability classification, and TTP extraction.

SecureBERT is pretrained on structured cybersecurity datasets, including vulnerability reports and threat intelligence feeds, and has shown strong results in entity recognition and incident classification. This domain-specific model has been effectively adopted in several recent works. In TTPXHunter [55], SecureBERT is used to extract actionable TTPs from finalized cyber threat reports, demonstrating improved performance in mapping text to ATT&CK techniques. Orbinato et al. [41] incorporated SecureBERT in a large-scale experimental study on CTI mapping, using it as a core component in evaluating automated NLP pipelines. Similarly, UniTTP [62] leverages SecureBERT within a unified deep learning framework to enhance the accuracy of TTP identification and mapping across heterogeneous

Table 1 Prior approaches to ATT&CK technique classification versus AC_MAPPER, with emphasis on augmentation type and class-aware balancing

Paper	Backbone/encoder	Data source	Class-aware	Label smoothing
Rani et al. [55]	SecureBERT (domain-specific encoder)	MITRE ATT&CK	Partially	No
Grigorescu et al. [38]	SecBERT (domain-specific encoder)	CVE	Partially	No
El Jaouhari et al. [66]	Universal Sentence Encoder (USE) + MLP (LabelPowerset)	CVE	No	No
Kim & Kim [42]	Classical ML	CTI reports	No	No
AC_MAPPER (Ours)	CTI-BERT (domain-specific encoder)	MITRE ATT&CK	Fully (upsamples every minority class to the target threshold)	Yes (label-smoothing objective)

CTI sources. SecBERT [59] Cyber Threat Intelligence Language Model (CTI-LM) [61], is a BERT-based architecture specifically fine-tuned for the CTI domain. This specialised pre-training and fine-tuning enable the CTI-LM to better understand and extract relevant information from unstructured cyber threat intelligence reports, demonstrating superior performance in tasks such as entity recognition and relation extraction within the cybersecurity context compared to general-purpose LLMs. However, the literature shows that this specialised LLM has not been explored in the CTI mapping tasks.

2.3 Data augmentation in CTI and ATT&CK mapping

Data quality challenges in cyber threat intelligence (CTI) are modality-dependent, so remediation must match the data. In computer vision, geometric transforms and denoising (e.g., median filtering, multi-wavelet transforms) are typical [63, 64]. In text-based CTI and ATT&CK mapping, semantics-preserving augmentation (e.g., constrained synonym/paraphrase edits and template-based entity slotting) together with basic noise reduction is used to mitigate data scarcity and label noise [40]. These issues are compounded by the lack of high-quality, ATT&CK-aligned annotations: many CTI sources are unstructured or inconsistently labeled, which harms generalisation and increases overfitting risk [65].

As summarised in Table 1, recent work explores two design categories for augmentation in this domain: (i) the type of augmentation, contextual (LM-driven) versus generic (rule-based/EDA/oversampling) and (ii) whether the procedure is class-aware, i.e., explicitly rebalances minority techniques. Rani et al. [55] (TTPXHunter) expand minority techniques using a SecureBERT-based masked-language-model augmentation pipeline. The authors grow the sentence corpus from ~ 10,906 to 39,296 instances covering 193 techniques and then evaluate on both the augmented sentences and 149 labelled threat reports. This preserves semantics but does not enforce per-class parity. Grigorescu et al. [38] (CVE2ATT&CK) employ TextAttack EDA (synonym replace/insert/swap/delete) to mitigate skew in a Table 2 multi-label CVE to techniques task. The augmentation improves coverage but still stops short of enforcing a target count per technique. Likewise, El Jaouhari et al., [66] pair Universal Sentence Encoder embeddings with an MLP (Label-Powerset) and apply EDA to enlarge training data, but the augmentation is dataset-wide rather than quota-driven per class. Kim and Kim [42] study classical machine learning with EDA and SMOTE oversampling, and these strategies boost size but remain agnostic to per-class targets. Most reported gains come from increasing the training data, either with contextual, language-model augmentation or with generic methods such as EDA. However, explicit class balancing is uncommon, especially for low-frequency techniques, and label smoothing—a simple way to reduce over-confidence under oversampling—is rarely used.

Table 2 Notation summary

Symbol	Meaning
x	A single data sample (e.g., text input)
c	A class label
D	Original training dataset
D'	Balanced and optionally augmented dataset
T	Target number of samples per class (e.g., median of class counts)
p_{aug}	Probability of applying input text augmentation
s	A sampled instance from underrepresented classes
TxtCol	The label column in the dataset
TxtCol	The text column in the dataset
t'	The paraphrased text
t	The original text
seed	The chosen random seed
doc	The text after parsed

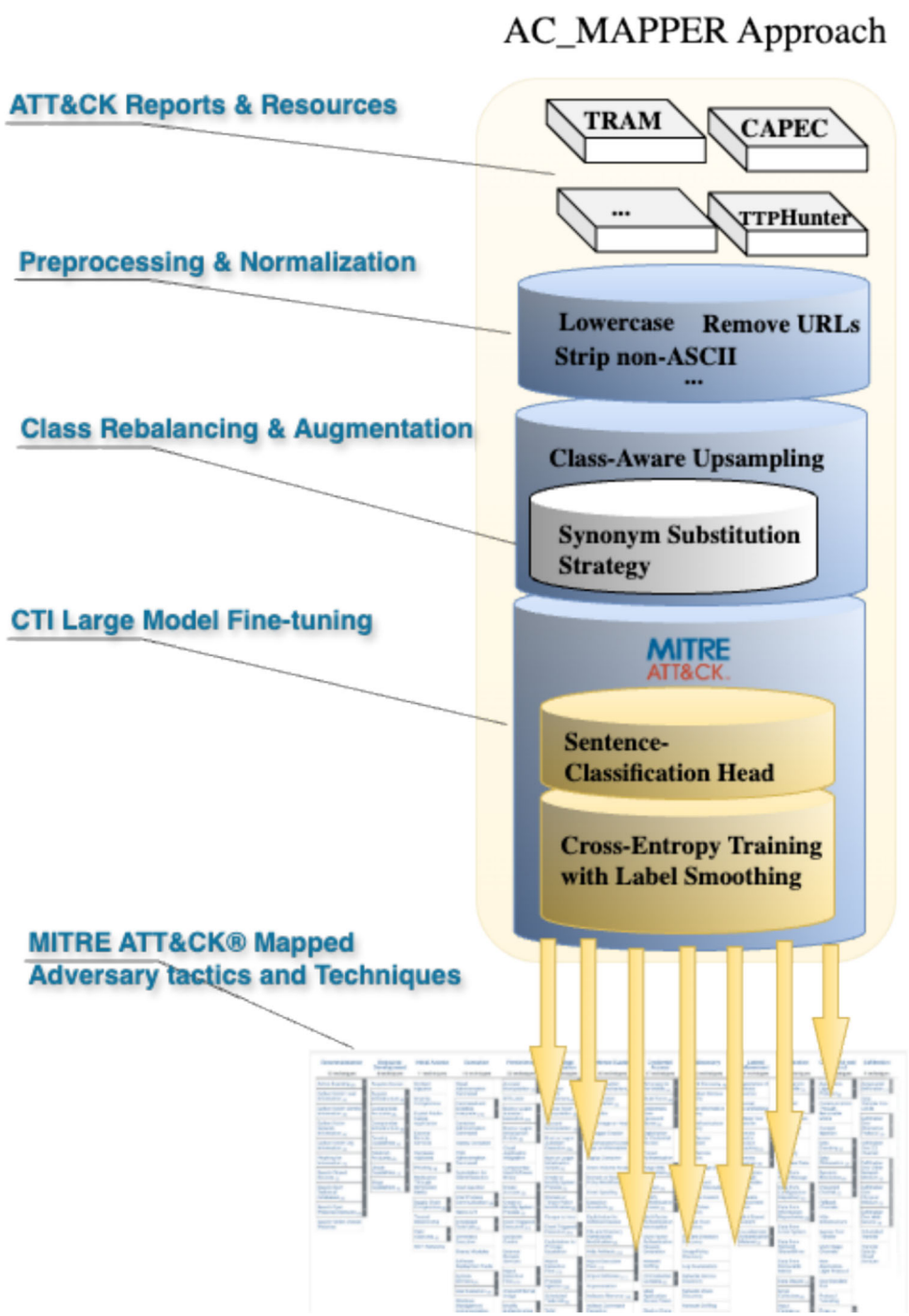

Fig. 3 AC_MAPPER Approach architecture

To address this gap, we propose a novel approach that combines class-aware upsampling with input augmentation, specifically tailored to imbalanced TTP distributions. Additionally, we utilise a domain-specific language model, CTI-BERT [61]. We treat augmentation as a balancing mechanism rather than solely a data-expansion tool. Concretely, we perform class-aware upsampling to a per-class target (median count) and apply restrained synonym substitution only to the oversampled instances, thereby correcting skew while preserving semantics. Also, in the training, we use label smoothing to reduce over-confidence on minority classes. Empirically, the approach consistently improves accuracy and macro-F1—most notably on heavily imbalanced datasets such as HALdata and CAPEC, and supports the premise that balance-then-augment is crucial for robust ATT&CK technique classification.

3 Methodology

This section explains the AC_MAPPER approach for balancing imbalanced labels with quality-preserving augmentation and the training objective used to fine-tune the model. Figure 3 shows the overall workflow, which starts with data acquisition from five public ATT&CK report sources, followed by preprocessing and normalising the text (including removal of noisy data and characters), applying class-aware upsampling and the synonym-substitution methods, and finally domain-specific LLM fine-tuning. The two methods and the training objective are detailed in the following subsections.

3.1 AC_MAPPER: class-aware rebalancing with quality-preserving augmentation

3.1.1 Motivation

Sentence-level ATT&CK labels are notoriously imbalanced in many proposed datasets [55] because a handful of common techniques dominate, while many are sparsely represented. Naïvely training on this distribution leads to high micro-scores but poor macro-F1 and weak performance on rare techniques. The AC_MAPPER approach addresses this with two components:

- Class-aware upsampling method that brings minority classes up to a modest, robust target size; and
- Synonym-guided paraphrasing method that injects lexical diversity without altering the underlying threat semantics.

3.1.2 Class-aware upsampling method

Due to the imbalanced distribution of MITRE technique annotations in threat datasets, we introduce a class-aware upsampling procedure combined with textual augmentation. As illustrated in Algorithm 1, Class-aware upsampling method ensures that under-represented classes are brought up to a target frequency. Specifically, the method computes the median class frequency across the training set and uses it as the target count T.

Let $D = \{(x_i, \ell_i)\}$ denote the training set and D_c the subset for class c. We compute the per-class counts $|D_c|$ and define the target count T as the median of these counts. For each minority class ($|D_c| < T$), we sample with replacement $n = T - |D_c|$ additional instances from D_c. For each sampled instance, with probability p_{aug} we apply the synonym substitution strategy (Sect. 3.1.3) before adding it back to D_c. Concatenating all D_c and shuffling yields the balanced training set D'. Choosing the median rather than the maximum avoids aggressive oversynthesis while still mitigating imbalance. Moreover, three design choices keep the upsampling from degrading data quality:

- We upsample only to the median class size, limiting duplication.
- Augmentation is probabilistic; many minority examples remain unmodified, preserving authentic language.
- The augmentation itself is guarded by a semantic similarity threshold (Sect. 3.1.3).

Algorithm 1 Class-Aware Upsampling Method

Require: Dataset D with text column *TxtCol* and label column LblCol
Require: Augmentation probability p_{aug}
Require: Synonym substitution function: *Synonym_Substitution_Strategy*()
Require: Target sampling size strategy (e.g., median of class sizes)
Ensure: Balanced and augmented dataset D'

```
Compute class counts ← count of each unique label in LblCol
target_count ← median(class counts)
Initialise empty dataset D' ← ∅
for each class c in class counts do
    D_c ← {x ∈ D | x.LblCol = c}
    if |D_c| < target_count then
        n ← target_count − |D_c|
        D_sampled ← randomly sample n instances with replacement from D_c
        for each sample s in D_sampled do
            if random() < p_aug then
            s.TxtCol ← Synonym_Substitution_Strategy(s.TxtCol)
            end if
            Add s to D_c
        end for
    end if
    Add D_c to D'
end for
Shuffle D' randomly
return D'
```

3.1.3 Synonym substitution method

Naïve thesaurus-based replacement can distort the semantics of cybersecurity text (e.g., inappropriate substitutions for beacon or persistence). To mitigate this risk, the Synonym Substitution method paraphrases only eligible content words and protects security-critical spans as shown in Fig. 4, and As illustrated in Algorithm 2. The procedure comprises three stages: (i) parsing and selecting eligible tokens, (ii) generating domain-aware candidate substitutes, and (iii) validating candidates for semantic fidelity.

(i) Parse and select eligible tokens. Using spaCy, we identify tokens that.

- are not stopwords, punctuation, or members of the ATTACKER_ACTIONS list (the latter excludes verbs denoting core attacker actions—e.g., exfiltrate, pivot, lateral move—to avoid weakening technique cues);
- have a part-of-speech tag in {VERB, NOUN, ADJ}; and
- appear, after lemmatisation, in a curated domain synonym lexicon (DOMAIN_SYNS). (Lemmatisation consolidates inflectional variants to a single base form.)

(ii) Generate domain-aware candidates. For each eligible token, we propose one or more replacements drawn from DOMAIN_SYNS lexicon.

(iii) Semantic validation. To prevent semantic drift, we compute Sentence-BERT (SBERT) cosine similarity between the original and paraphrased sentences and accept a substitution only if the similarity exceeds the threshold $\tau = 0.85$. Up to three substitution plans are attempted; if none satisfy the threshold, the original text is retained. When SBERT is unavailable, we default to the first candidate (as noted in the experiments).

Although sampling is performed with replacement, overfitting is mitigated through probabilistic synonym-based augmentation (p_{aug}), label smoothing ($\varepsilon = 0.1$), and random shuffling of the augmented dataset before training.

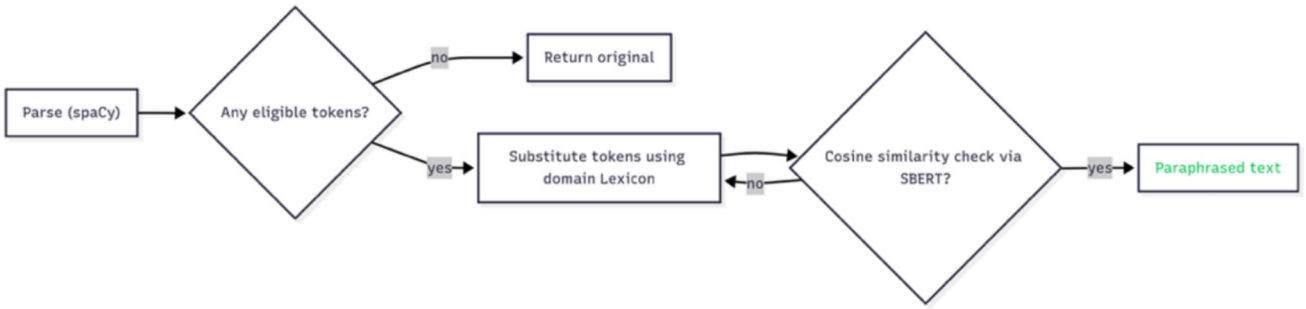

Fig. 4 synonym substitution strategy method pipeline

Algorithm 2 Synonym Substitution Strategy Method (Text)

Require: input text **t**
Require: spaCy model**; DOMAIN_SYNS** map**; ATTACKER_ACTIONS** set
Require: Parameters**: target_pos** = {VERB, NOUN, ADJ},
sbert_thresh, **retries**, **seed**
Require: SBERT model (for cosine similarity)
Ensure: paraphrased text t′ or original **t**

SetRandomSeed(seed)
doc ← SPACY_PARSE(Text)
eligible ← { i | token doc[i] is not stop/punct/protected
∧ doc[i].POS ∈ target_pos
∧ lower(doc[i].lemma) ∉ ATTACKER_ACTIONS
∧ DOMAIN_SYNS has lower(doc[i].lemma) }
if |eligible| = 0 **then return** t
for attempt = 1 .. retries **do**
candidate ← SUBSTITUTE_USING_DOMAIN_LEXICON(doc, eligible)
if SBERT unavailable ∨ COS_SIM(SBERT(t), SBERT(candidate)) ≥ sbert_thresh **then**
return cand ▷ paraphrased text
end if
end for
return t

3.2 Notation summary

3.2.1 Sentence-level classification head and label-smoothing loss

AC_MAPPER turns the transformer-pretrained large model CTI-BERT[58] (trained for masked-language modelling and representation learning) into a sentence-level classifier of MITRE ATT&CK techniques by replacing the pretraining head with a sentence-classification head consisting of dropout followed by a linear layer whose size equals the number the labels (ATT&CK techniques) in the datasets. Furthermore, to improve generalisation and mitigate model overconfidence during training, we employ cross-entropy loss with label smoothing, inspired by [67]. Concretely, the one-hot targets are smoothed with $\varepsilon = 0.1$: the target class probability is reduced from 1.0 to 0.9, and the remaining 0.1 is distributed uniformly across the non-target classes.

Table 3 Dataset statistics and technique distribution

Dataset	Sentences	Techniques	Most frequent	Rare < 10	Mean freq
TRAM	5089	50	T1027 (685)	1	101.78
TRAM bootstrap	11,130	50	T1027 (1043)	1	222
CAPEC	12,945	188	T1059 (698)	23	68.86
TTPHunter	8887	50	T1059 (671)	0	167.74
HALdata	803	46	T1105 (106)	23	17.5

Let $z \in \mathbb{R}^K$ be the logits for a sentence and $p = \text{softmax}(z)$ the predicted distribution over K techniques. Instead of a one-hot target, we use label smoothing with parameter ε:

$$\tilde{y}_k = \begin{cases} 1-\varepsilon, & k = y, \\ \varepsilon/(K-1), & k \neq y, \end{cases} \quad \mathcal{L}_{\text{LS}}(x, y) = -\sum_{k=1}^{K} \tilde{y}_k \log p_k. \tag{1}$$

We set $\varepsilon = 0.1$, which strikes a pragmatic balance by reducing over-confidence on frequent classes, providing regularisation against mild label noise (common in weakly labelled CTI sentences), and consistently improving macro-F1 in our experiments without harming optimisation stability.

4 Experiments and results

4.1 Datasets

The proposed model is evaluated across five publicly available datasets: CAPEC [41], HALdata [68], TTPHunter [69], TRAM[70], and TRAM Bootstrap[43]. As shown in Table 3, the datasets vary significantly in size and technique distribution. CAPEC is the largest and most diverse, containing 12,945 sentences, and covering a wide range of techniques (188). In contrast, HALdata is the smallest dataset with only 805 sentences and a high number of rare techniques, while TRAM, TTPHunter, and TRAM bootstrap offer larger sentence counts and a more focused set of 50 techniques each. The datasets also exhibit varying levels of class imbalance, with T1027 and T1059 being the most frequent techniques in several datasets, and a notable presence of rare techniques in HALdata and CAPEC, indicating a challenging distribution for model training and evaluation.

The Threat Report ATT&CK Mapper (TRAM) is an open-source tool developed by the MITRE Engenuity Center for Threat-Informed Defense to streamline the process of aligning cyber threat intelligence (CTI) reports with the MITRE ATT&CK framework [43]. In 2023, the project evolved into TRAM II, introducing an updated dataset along with a new "bootstrap" dataset aimed at enhancing machine learning training. TRAM supports researchers and analysts in developing and evaluating machine learning models that extract ATT&CK techniques from narrative CTI reports, promoting broader integration of ATT&CK across the threat intelligence community [70].

CTI-to-MITRE (CAPEC) The dataset comprises textual descriptions of cyber threats, primarily sourced from MITRE ATT&CK and CAPEC Knowledge bases [70]. They built the dataset by extracting both descriptions and their mappings to MITRE ATT&CK by analysing the knowledge base as released by MITRE using the Structured Threat Information eXpression (STIX) language [41].

HALdata The CTI-HAL ("Cyber Threat Intelligence—Human-Annotated Labels") dataset is a recent contribution to the field of cybersecurity, aiming to enhance the analysis of cyber threat intelligence (CTI) through structured data which focuses on statement-Level Annotations [68]. Unlike many existing datasets that provide document-level annotations, CTI-HAL offers fine-grained, sentence-level annotations, so each sentence in the CTI reports is examined and labelled with corresponding MITRE ATT&CK technique identifiers, enabling precise mapping between textual descriptions and specific adversarial behaviours. However, the number of rows in this dataset after deleting the duplicates is fewer than others, which affects the learning in the proposed model and can be seen clearly in the results in Sect. 4.

TTPHunter The TTPHunter dataset is introduced in the paper titled"TTPHunter: Automated Extraction of Actionable Intelligence as TTPs from Narrative Threat Reports" by Nanda Rani et al. [69], and it serves the same purpose as automating the extraction of Tactics, Techniques, and Procedures (TTPs) from unstructured Advanced Persistent Threat (APT) reports. It comprises 50 full-length APT reports and contains labelled sentences extracted from those reports.

4.2 Datasets strengths, limitations, and labelling challenges

Most ATT&CK-aligned datasets used in this study vary in domain coverage, sentence diversity, and class balance, reflecting both the breadth and limitations of available cyber threat intelligence sources. Larger resources like [70] and [69] cover hundreds of techniques and thousands of sentences, offering extensive context and adversarial variety,

Table 4 Performance comparison (CAPEC Dataset)

Dataset	Model	Accuracy	Precision	Recall	F1 (Macro)
CAPEC	SCIBERT[9]	0.7686	0.5835	0.5674	0.5585
	TTPHunter[51]	0.7655	0.5284	0.5247	0.5059
	TTPXHunter[52]	0.8208	0.6593	0.6598	0.6439
	DistilBERT[54]	0.7462	0.5584	0.5265	0.5208
	SecBERT[2]	0.7466	0.5696	0.5543	0.5406
	SecureBERT[3]	0.7655	0.5467	0.5514	0.5279
	PRIMUS [72]	0.8443	0.7473	0.7303	0.7223
	CTI-BERT[46]	0.8169	0.6704	0.6587	0.6484
	AC_MAPPER (Ours)	**0.8480**	0.8457	0.8439	**0.8322**

Bold indicates the best performance for each metric (e.g., Accuracy, Precision, etc.), while underlined denotes the second-best performance if applicable

Table 5 Performance comparison (HALdata Dataset)

Dataset	Model	Accuracy	Precision	Recall	F1 (Macro)
HALdata	SCIBERT[9]	0.5901	0.3378	0.3802	0.3469
	TTPHunter[51]	0.5963	0.3428	0.4300	0.3646
	TTPXHunter[52]	0.6211	0.4275	0.4686	0.4217
	DistilBERT[54]	0.5093	0.2540	0.2522	0.2181
	SecBERT[2]	0.6398	0.4548	0.4643	0.4355
	SecureBERT[3]	0.6025	0.2875	0.3798	0.3110
	PRIMUS [72]	0.5776	0.4422	0.4275	0.3955
	CTI-BERT[46]	0.6646	0.4151	0.4806	0.4174
	AC_MAPPER (Ours)	**0.7151**	0.6435	0.6660	**0.6452**

Bold indicates the best performance for each metric (e.g., Accuracy, Precision, etc.), while underlined denotes the second-best performance if applicable

while others remain highly imbalanced or focus only on the most common threats. These inherent disparities challenge generalizability, as models may underperform when exposed to rare techniques or language patterns absent from benchmark corpora.

Critically, constructing and labelling such datasets requires deep expertise and careful interpretation of technical language. Annotation is labour-intensive and subject to ambiguity, especially as the boundaries between ATT&CK techniques can be subtle within complex security reports [68]. Notably, the TRAM and TRAM bootstrap dataset are developed by MITRE itself, ensuring high fidelity with the official ATT&CK framework and providing benchmarks that best represent operational cyber threat intelligence [43].

4.3 Environment setup

All experiments are conducted using a Google Colab GPU backend (Python 3 environment) with access to approximately 83 GB of RAM and GPU acceleration, ensuring practical reproducibility. This setup ensured sufficient memory and computational resources for fine-tuning large transformer-based language models across multiple datasets. The code for implementation, testing, and dataset acquisition will be made available upon publication.

4.4 Training setup

In each experiment, the dataset is split into training and test sets using an 80/20 ratio. The selected transformer-based model is fine-tuned over six epochs, with all layers unfrozen. Training is conducted using a batch size of 16 and optimised with the AdamW optimiser. For Bert and RoBERTa-based models, all transformer layers were unfrozen and fine-tuned jointly using a uniform learning rate of 5e-5 under the AdamW optimiser with a linear warm-up schedule. For the LLaMA-based baseline, fine-tuning was performed using PEFT with LoRA (targeting q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, and down_proj layers), where only adapter parameters were trainable and all base model weights remained frozen.

Table 6 Performance comparison (TTPHunter Dataset)

Dataset	Model	Accuracy	Precision	Recall	F1 (Macro)
TTPHunter	SCIBERT[9]	0.9046	0.8943	0.8902	0.8906
	TTPHunter[51]	**0.9476**	0.9417	0.9413	**0.9405**
	TTPXHunter[52]	0.9428	0.9283	0.9349	0.9304
	DistilBERT[54]	0.9005	0.8808	0.8808	0.8785
	SecBERT[2]	0.8951	0.8783	0.8757	0.8756
	SecureBERT[3]	0.9112	0.8962	0.8945	0.8938
	PRIMUS [72]	0.9207	0.9011	0.9009	0.8989
	CTI-BERT[46]	0.9130	0.8979	0.8987	0.8969
	AC_MAPPER (Ours)	0.9337	0.9288	0.9299	0.9282

Bold indicates the best performance for each metric (e.g., Accuracy, Precision, etc.), while underlined denotes the second-best performance if applicable

Table 7 Performance comparison (TRAM Dataset)

Dataset	Model	Accuracy	Precision	Recall	F1 (Macro)
TRAM	SCIBERT[9]	0.8959	0.8723	0.8264	0.8364
	TTPHunter[51]	0.8929	0.8624	0.8353	0.8368
	TTPXHunter[52]	0.9037	0.8457	0.8357	0.8341
	DistilBERT[54]	0.8792	0.7830	0.7635	0.7618
	SecBERT[2]	0.8644	0.8402	0.7848	0.7909
	SecureBERT[3]	0.8851	0.7901	0.7715	0.7666
	PRIMUS [72]	0.8860	0.8586	0.8364	0.8391
	CTI-BERT[46]	0.9037	0.8461	0.8311	0.8306
	AC_MAPPER (Ours)	**0.9060**	0.8922	0.9090	**0.8975**

Bold indicates the best performance for each metric (e.g., Accuracy, Precision, etc.), while underlined denotes the second-best performance if applicable

4.5 Evaluation metrics and baseline

After training, the model is evaluated using standard classification metrics, including macro, micro F1 scores. The average training loss per epoch is recorded to monitor convergence trends.

5 Results

We evaluated the proposed approach (AC_MAPPER) against seven competitive baselines across five publicly available datasets: CAPEC, HALdata, TTPHunter, TRAM, and TRAM Bootstrap. Each model's performance was measured using **accuracy**, **macro F1-score** and **micro F1-score** to assess both correctness and robustness across imbalanced class distributions. The proposed model is compared against a range of established baseline models, including domain-specific language models, which are: SecBERT [59], SecureBERT [29], TTPHunter [69], and TTPXHunter [55], as well as general-purpose models like DistilBERT [71] and SciBERT [44]. We also evaluate PRIMUS, a recent Llama-based model tailored to cybersecurity [72]. Furthermore, the model is compared with CTI-BERT[61]to show the effects of the upsampling and augmentation strategy. These baselines represent state-of-the-art approaches in cyber threat intelligence extraction and classification, providing a strong foundation for evaluating the effectiveness of our approach.

Overall Performance As shown in Tables 4,5,6,7, and 8, the proposed approach (AC_MAPPER) consistently achieved the highest performance in most datasets across both accuracy and macro F1 metrics. Specifically:

- **CAPEC**: The proposed model's superior performance on the CAPEC dataset is clearly demonstrated as presented in Table 4 and shown in Fig. 5. It achieved the highest accuracy (84.8%) and macro F1 (83.22%), outperforming the CTI-BERT baseline by a significant margin. This strong result is visually supported by the training loss curve for the CAPEC Dataset, which shows the proposed model's loss consistently decreasing and converging to one of the lowest final values among all evaluated models. This indicates that our model is not only achieving better classification

Table 8 Performance comparison (TRAM Bootstrap Dataset)

Dataset	Model	Accuracy	Precision	Recall	F1 (Macro)
TRAM Bootstrap	SCIBERT[9]	0.8944	0.8489	0.8477	0.8429
	TTPHunter[51]	0.9097	0.8694	0.8699	0.8652
	TTPXHunter[52]	0.9133	0.8787	0.8921	0.8825
	DistilBERT[54]	0.9030	0.8600	0.8617	0.8557
	SecBERT[2]	0.8904	0.8548	0.8479	0.8478
	SecureBERT[3]	0.9034	0.8593	0.8672	0.8588
	PRIMUS [72]	0.9164	0.9015	0.8987	0.8963
	CTI-BERT[46]	0.9151	0.8832	0.8929	0.8858
	AC_MAPPER (Ours)	**0.9359**	0.9357	0.9414	**0.9378**

Bold indicates the best performance for each metric (e.g., Accuracy, Precision, etc.), while underlined denotes the second-best performance if applicable

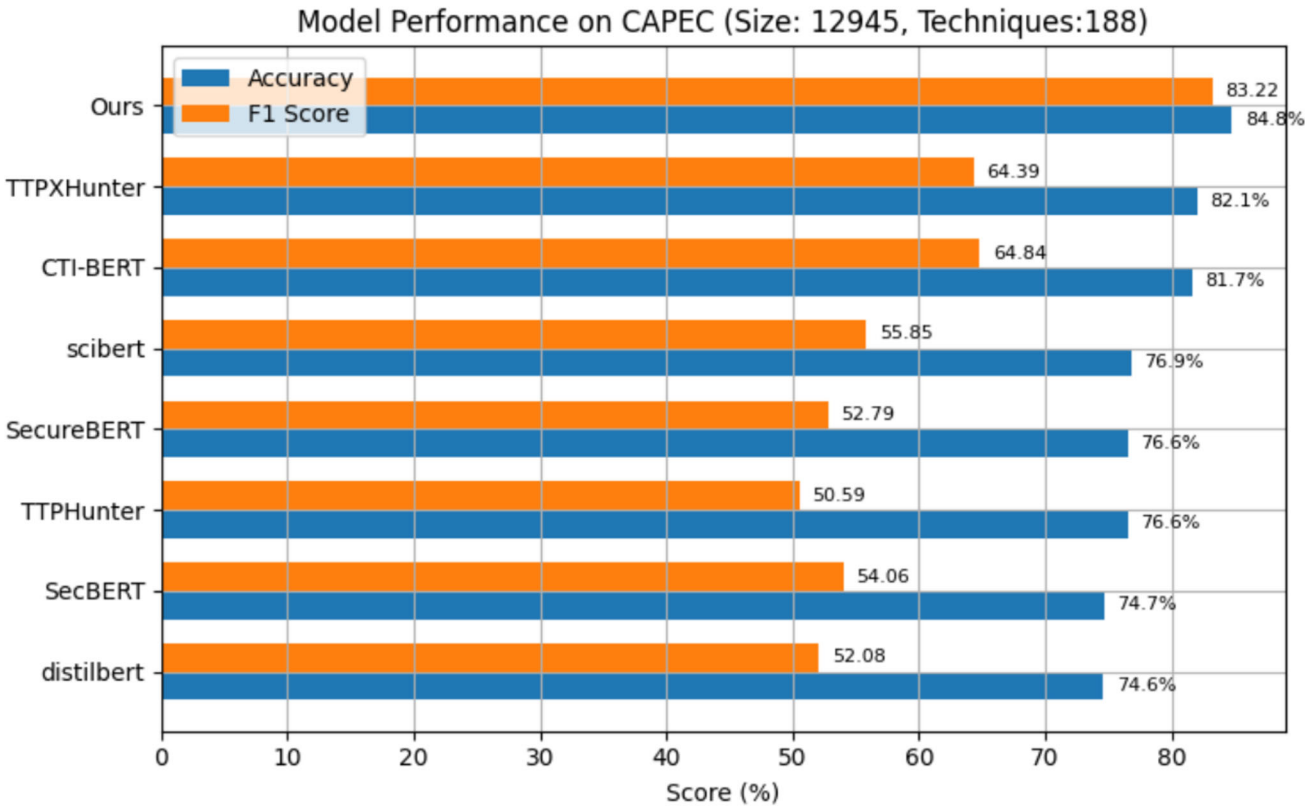

Fig. 5 The best-performing model per dataset—The proposed model's performance (Accuracy & F1) on CAPEC Dataset

metrics but is also learning the underlying patterns of the dataset with greater efficiency and stability over the six training epochs.

- **HALdata**: The HALdata dataset presents a unique challenge due to its small size (805 sentences) and a high number of rare techniques. This is reflected in the loss, which shows that all models, including the proposed one, have a higher starting and ending loss compared to the more balanced and larger datasets like TRAM or CAPEC (Fig. 6). Specifically, AC_MAPPER (Ours)'s loss curve is positioned in the middle of the pack. While it successfully reduces loss throughout the training process, it does not achieve the lowest final loss on this particular dataset. The SecBERT and TTPHunter models appear to converge to slightly lower loss values. This result indicates that while the proposed model is competitive, its performance in minimising loss may be less dominant when faced with the combined challenge of a small training set and significant class imbalance, which is a key characteristic of the HALdata dataset. Despite being the smallest and most

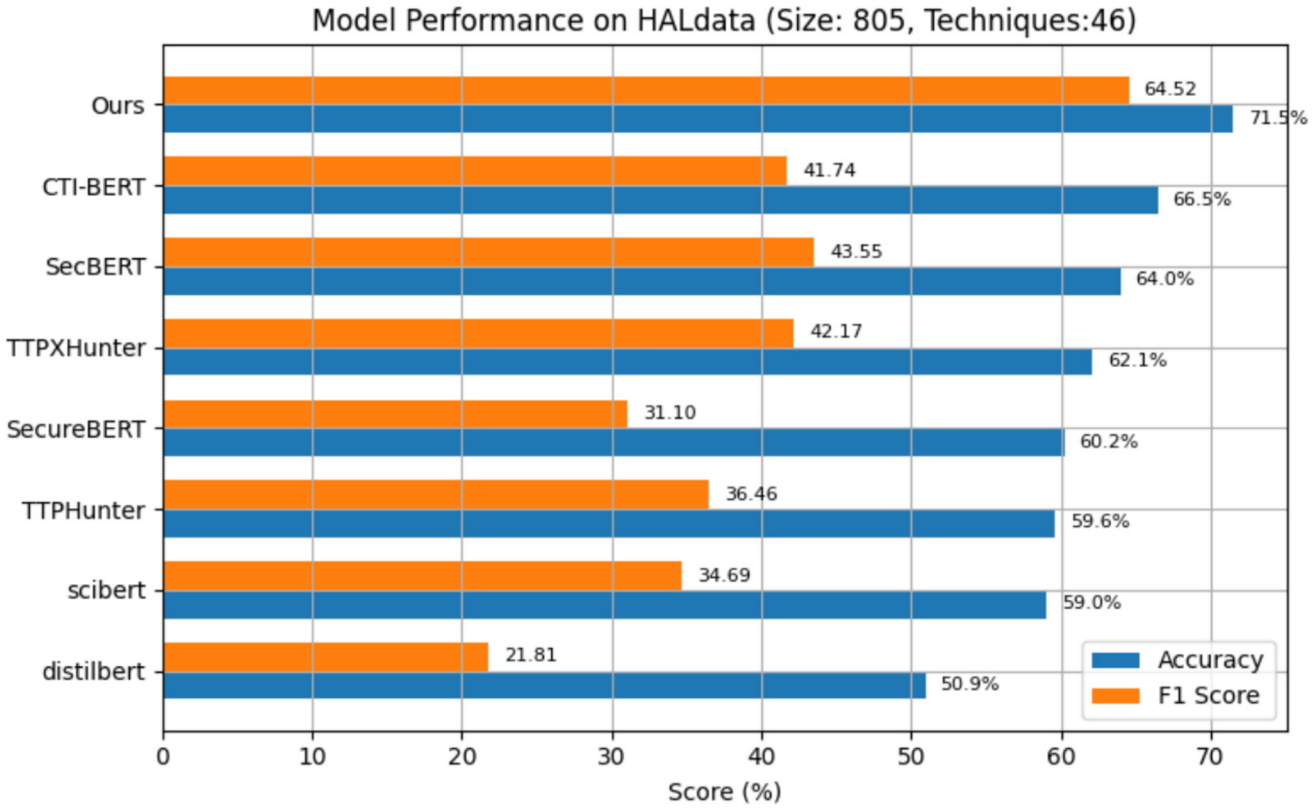

Fig. 6 The Best-Performing Model per Dataset—The proposed Model's Performance (Accuracy & F1) on HALdata Dataset

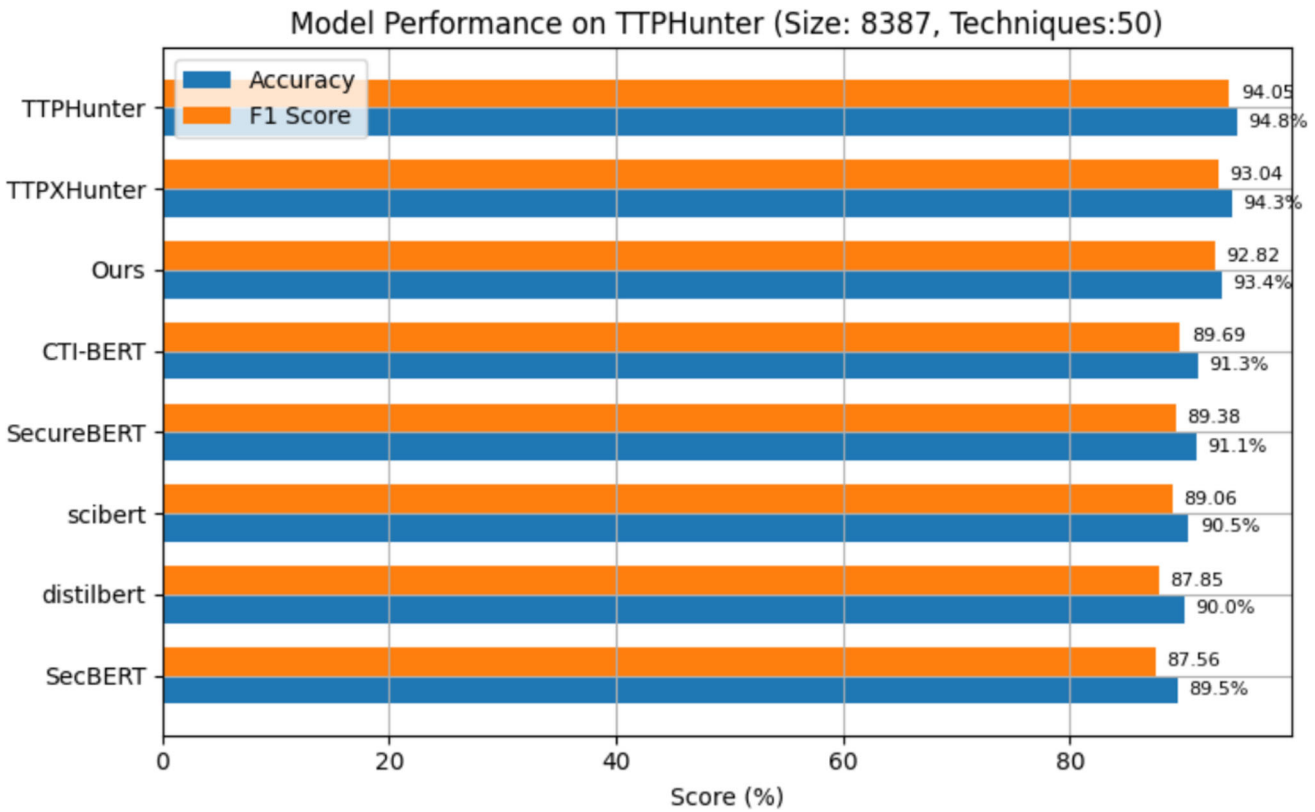

Fig. 7 The Best-Performing Model per Dataset—The proposed Model's Performance (Accuracy & F1) on TTPHunter Dataset

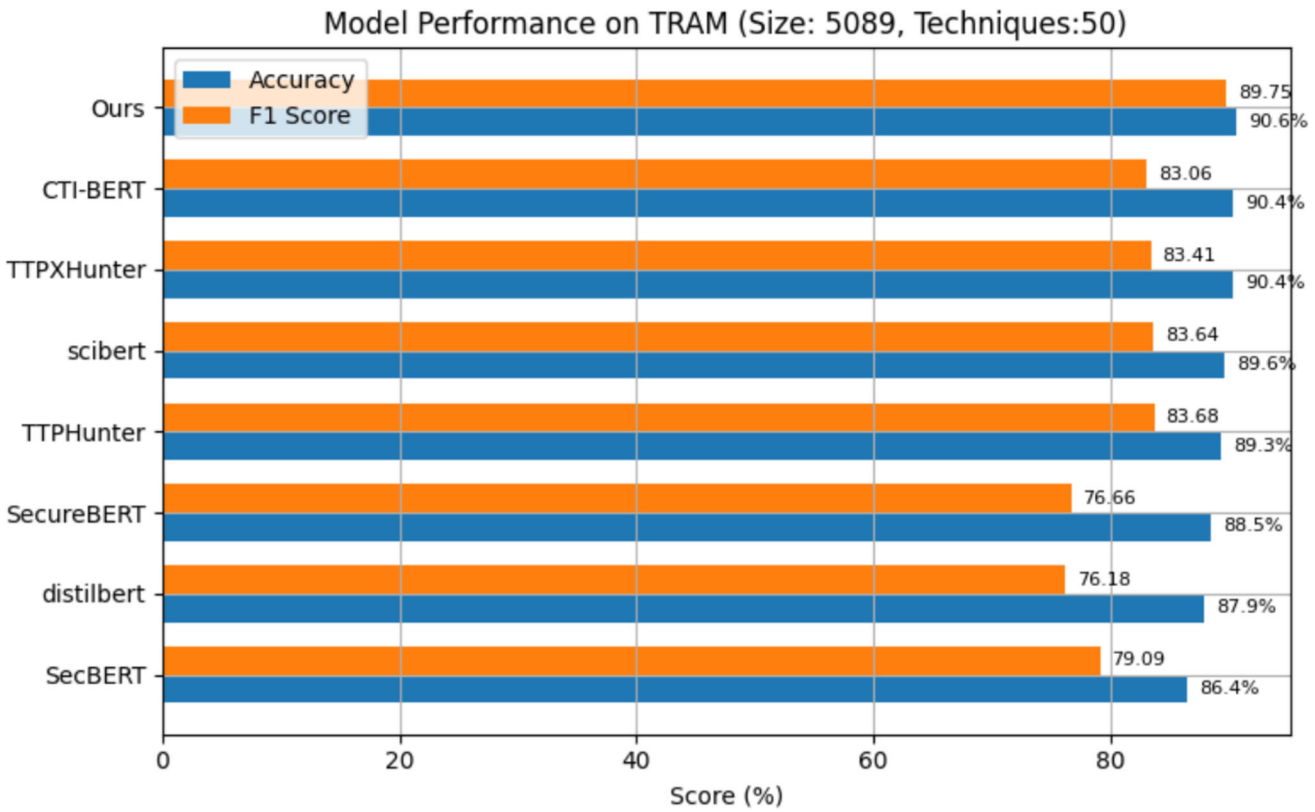

Fig. 8 The Best-Performing Model per Dataset—The proposed Model's Performance (Accuracy & F1) on TRAM Dataset

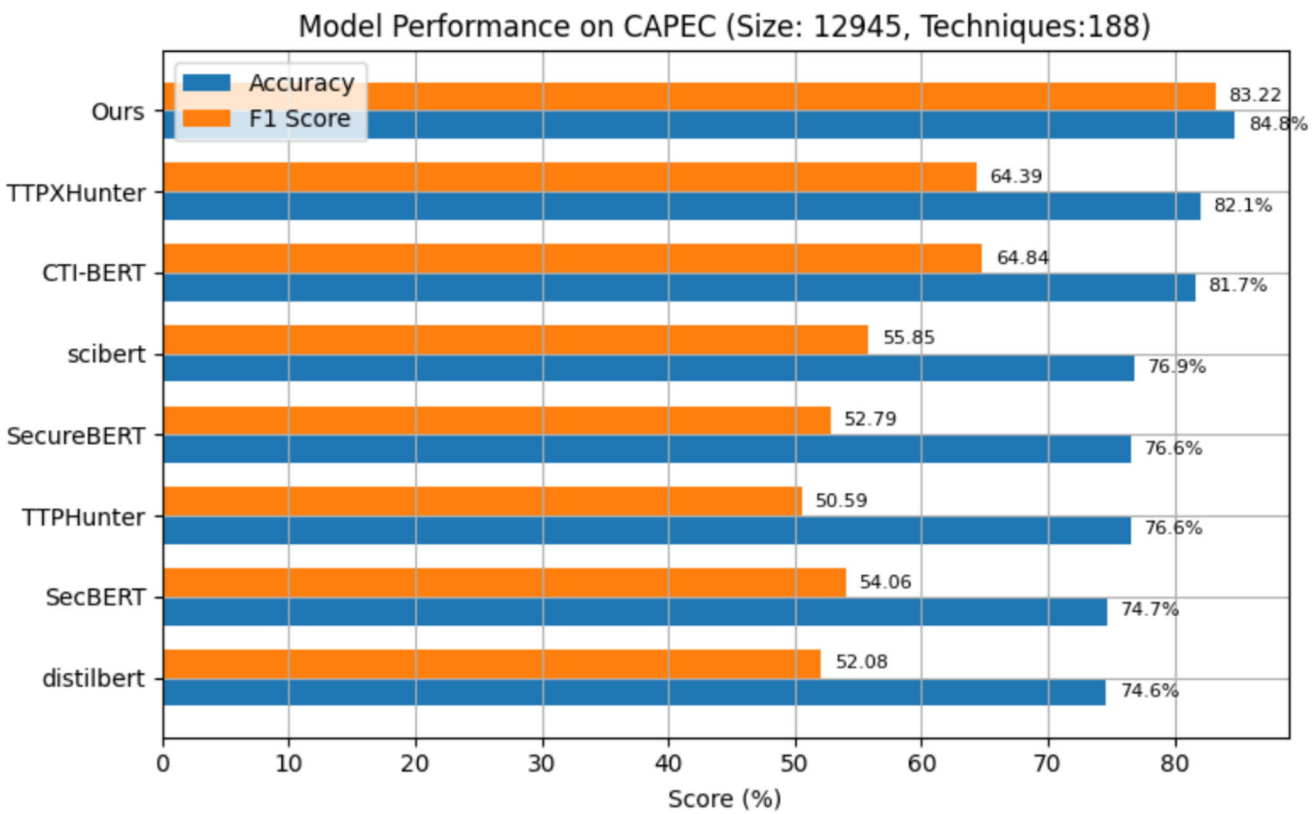

Fig. 9 The Best-Performing Model per Dataset—The proposed Model's Performance (Accuracy & F1) on TRAM Bootstrap Dataset

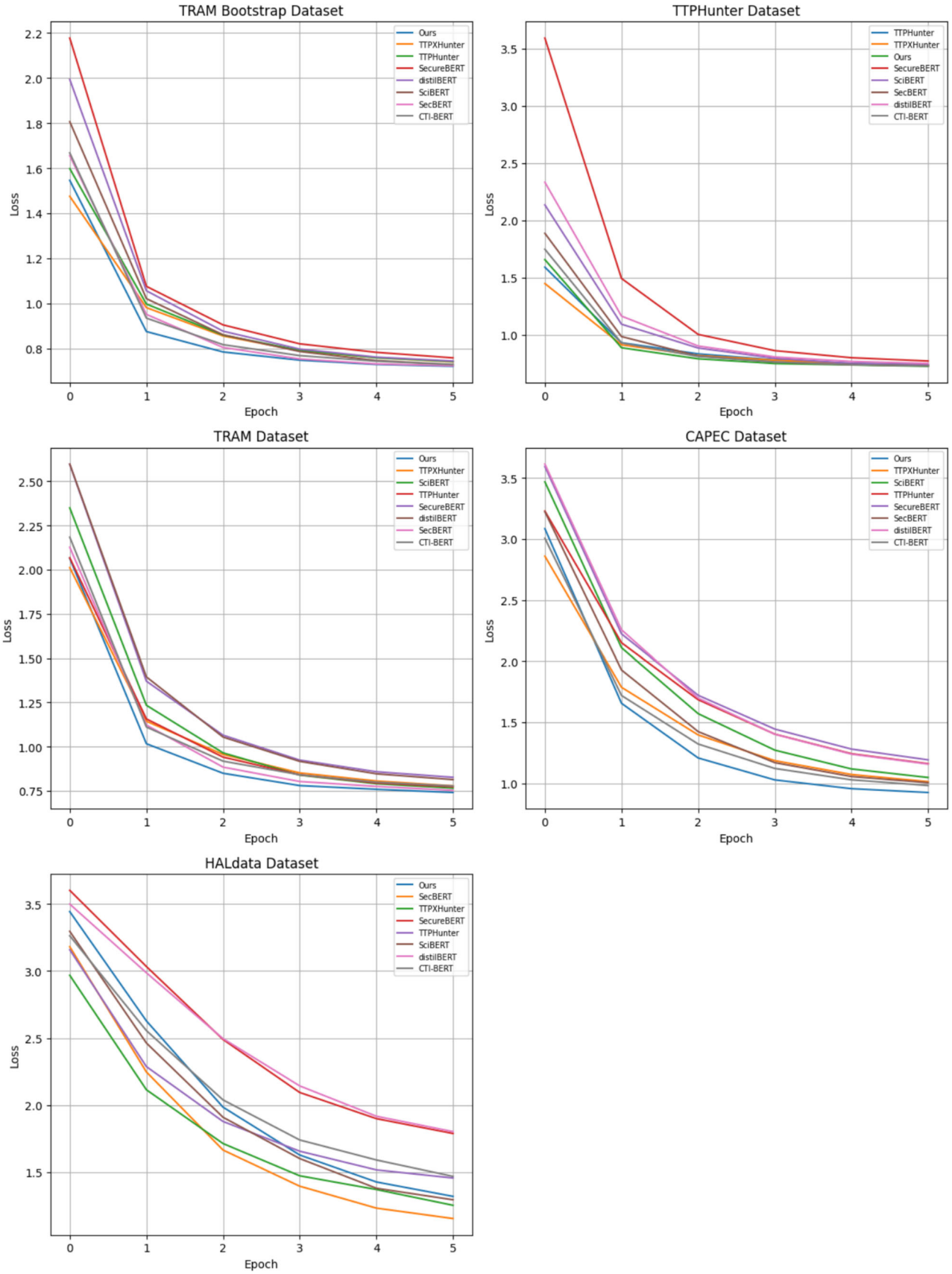

Fig. 10 Comparison of Average Training Loss per Epoch for All Datasets

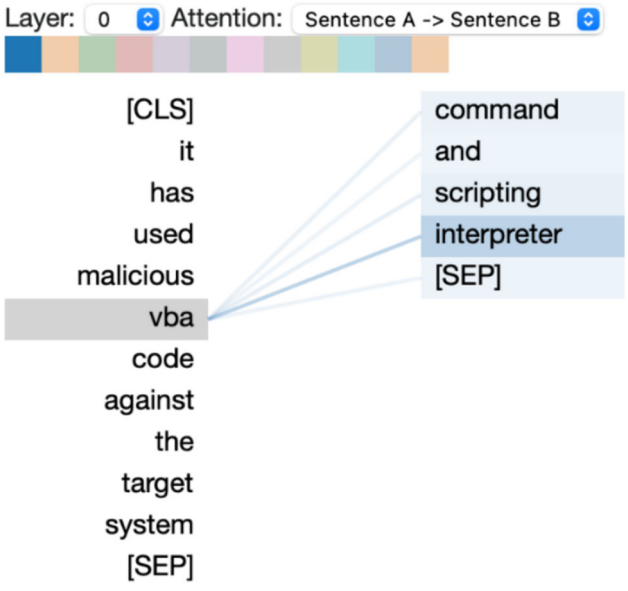

Fig. 11 AC_MAPPER's attention weights between a sentence and its ATT&CK technique from TTPHunter dataset

imbalanced dataset, AC_MAPPER approach led with an accuracy of **71.51%** and macro F1 of **64.52%** (Table 5, Fig. 6).

- **TTPHunter**: The best performance was obtained by TTPHunter (accuracy: **94.76%**, macro F1: **94.05%**), slightly outperforming AC_MAPPER (our approach, accuracy: 93.37%, macro F1: 92.82%), which is expected since the baseline was optimised specifically for this dataset, (Table 6, Fig. 7). The graph in Fig. 10 shows a steep and consistent decrease in loss for all models, indicating that the dataset, with its focused set of 50 techniques, is well-suited for effective model training. AC_MAPPER performs exceptionally well on this dataset. Specifically, the "AC_MAPPER" loss curve converges to one of the lowest final values among all the evaluated models, closely tracking or slightly outperforming the top baselines, such as CTI-BERT and the TTPHunter models. This result provides strong evidence of the proposed model's robustness and efficiency, demonstrating its ability to learn and classify adversarial techniques with high accuracy on a well-structured and balanced dataset.
- **TRAM**: The proposed AC_MAPPER approach achieved the top performance with an accuracy of **90.60%** and macro F1 of **89.75%**, marginally outperforming CTI-BERT and TTPXHunter (Table 7, Fig. 8).
- **TRAM Bootstrap**: On the TRAM Bootstrap dataset, the proposed approach (AC_MAPPER) achieved the highest performance across all datasets, with an accuracy of 93.59% and a macro F1 score of 93.78% (Table 8, Fig. 9). It outperformed CTI-BERT—the next best model—by + 2% in accuracy and + 4% in macro F1, demonstrating superior classification capability. The high F1 score indicates strong performance on both frequent and infrequent techniques, which suggests balanced learning. Additionally, the loss curve for this dataset shows that AC_MAPPER consistently converged faster and to a lower final loss compared to Fig. 10 all baseline models, confirming its efficiency and robustness.

5.1 Model interpretability analysis

To interpret AC_MAPPER's decisions, we extracted attention weights from the transformer encoder and visualised them with the BertViz tool. Figure 11 shows that tokens such as "VBA" and "code" exhibit strong cross-token attention, particularly toward the technique phrase "command and scripting interpreter", which indicates that the model is focusing on scripting/execution cues when predicting an ATT&CK technique. For reproducibility, we also generated a static attention heatmap by averaging heads in the final layer; Fig. 12 concentrates on "malicious," "VBA," and "code/target system," confirming that execution-related terms dominate the model's internal focus and providing interpretable evidence for T1059 (Command and Scripting Interpreter).

5.2 Qualitative predictive performance

Across the four CTI sentences in Table 9, AC_MAPPER correctly maps prose to the expected ATT&CK techniques in three cases: T1553 for the SolarWinds code-signing narrative, T1036 for implants placed in folders named for legitimate software, and T1057 for enumerating local processes. In each, the prediction aligns with salient lexical cues (e.g., code-signing certificates, legitimate software, enumerating processes), indicating that the model reliably grounds technique labels in the informative terms of the sentence. The remaining example (EnvyScout) can collect sensitive NTLM material from a compromised host, is labelled T1005, but predicted T1003. This mismatch illustrates a common challenge in CTI text: a single sentence can contain cues that plausibly support multiple techniques. In our case, local data collection and credential-related activities make disambiguation inherently difficult even for human annotators and lead to occasional cross-technique predictions.

5.3 Ablation studies

To further investigate the influence of prompting on model behaviour, an extended ablation study was conducted using four distinct prompting strategies, as summarised in Table 10. These strategies were designed to explore how varying levels of contextual guidance affect classification accuracy and

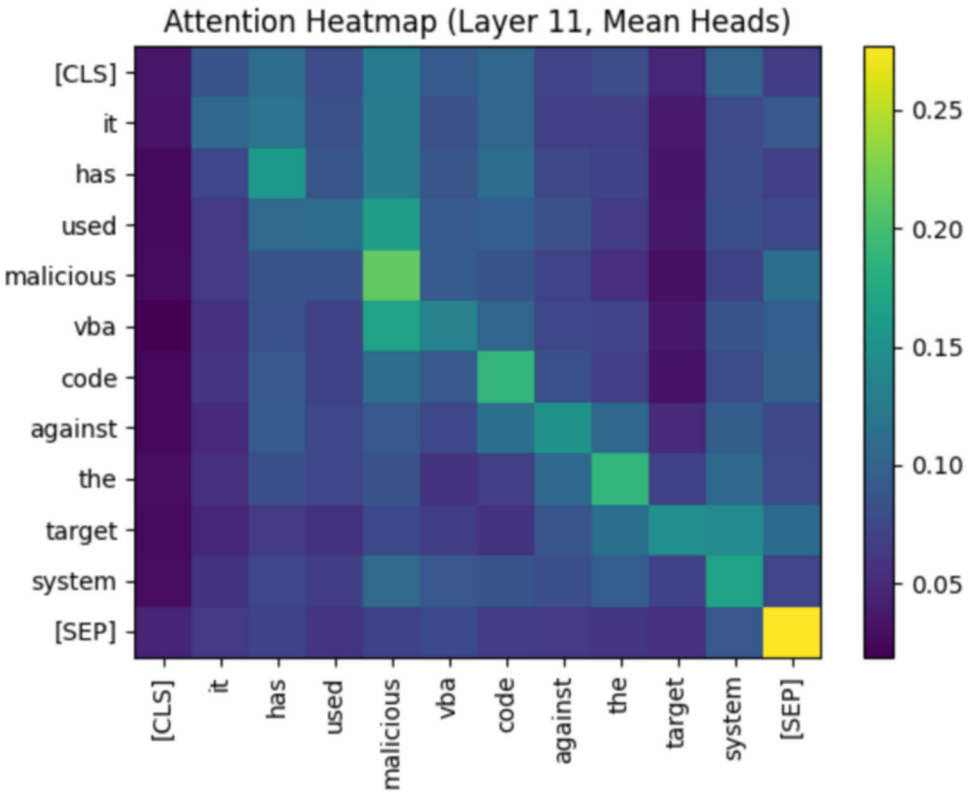

Fig. 12 AC_MAPPER's attention heatmap for a sentence from TTPHunter dataset

inference efficiency across heterogeneous CTI datasets. The evaluated configurations were as follows:

- No Prompt (Baseline): Original input text provided directly to the model without modification.
- Instructional (Q-style):
  - o "Text: {text}\n\nQuestion: Identify the MITRE ATT&CK technique that this text describes."
- Declarative (Contextual cue):
  - o "Given the following text:\n{text}\n\nThis text corresponds to the MITRE ATT&CK technique of:"
- Minimal Tag (Keyword prefix):
  - o "ATT&CK technique: {text}"

The results in Table 10 reveal that prompting exerts a dataset-dependent influence on performance. For structured and concise datasets such as TRAM and TRAM Bootstrap, the inclusion of brief contextual cues—particularly the declarative and keyword prompts—yielded the highest macro F1 scores (0.909 and 0.9426 respectively), improving results by approximately 0.3–0.5 percentage points compared with the baseline. In these datasets, where sentences are short and semantically focused, explicit cues likely help the model attend to technical action phrases (e.g., execute, inject, install), thereby enhancing precision without increasing inference cost.

In contrast, for noisier and linguistically diverse datasets such as CAPEC and HALdata, the impact of prompting was less consistent. The instructional prompt provided modest gains on HALdata (+ 0.04 macro F1), suggesting that explicit question framing can help the model interpret incomplete or context-poor inputs. However, longer declarative or keyword variants sometimes introduced redundant tokens, marginally reducing accuracy due to contextual dilution. For TTPHunter, differences across prompt types were minimal, indicating that the dataset's inherent alignment with ATT&CK semantics already supports strong baseline performance without additional prompting.

Overall, the study demonstrates that prompting can both enhance and hinder model effectiveness depending on dataset structure, sentence length, and contextual clarity. Short, well-targeted prompts tend to improve task focus in structured corpora, whereas verbose or repetitive phrasing may distract the model in complex, real-world CTI text. These findings emphasise the importance of adaptive prompt engineering, where prompt selection and phrasing are dynamically tailored to the characteristics of each dataset. Future work will extend this investigation by exploring learned prompt-tuning and hybrid contextualisation approaches that combine minimal textual cues with trainable prompt embeddings, aiming

Table 9 AC_MAPPER prediction on four sentences

	Sentence
1	apt29 was capable to get sunburst contracted by solarwinds code signing certificates by come ining the malware into the solarwinds orion software lifecycle
	Actual: (T1553: Subvert Trust Controls—Code Signing)
	Predicted:T1553
2	backdoordiplomacy has dropped implants in folders named for legitimate software
	Actual: T1036: Masquerading—Match Legitimate Name or Location)
	Predicted: T1036
3	emotet has been observed enumerating local processes
	Actual: (T1057: Process Discovery)
	Predicted: T1057
4	envyscout can collect sensitive ntlm material from a compromised host
	Actual: (T1005: Data from Local System)
	Predicted: (T1003: OS Credential Dumping)

Table 10 The effect of Prompting the Model across Various Datasets

Dataset		F1 Macro	Inference time (s)	Throughput (samples/s)
TRAM	No Prompt	0.8995	4.3	261.6
	Instructional	0.9008	4.4	256.8
	Declarative	<u>0.9032</u>	4.4	257.2
	keyword	**0.9092**	4.3	261.4
TRAM Bootstrap	No Prompt	0.9371	10.2	260.1
	Instructional	0.9381	10.3	256.5
	Declarative	**0.9426**	10.4	254.1
	keyword	<u>0.9405</u>	10.2	258.9
CAPEC	No Prompt	<u>0.8367</u>	11.1	258
	Instructional	**0.8407**	11.4	253.3
	Declarative	0.8293	11.3	253.3
	keyword	0.8308	11.2	257.5
HALdata	No Prompt	0.6454	0.7	260.8
	Instructional	**0.6866**	0.7	254.7
	Declarative	0.6472	0.7	254.9
	keyword	<u>0.6740</u>	0.7	252.2
TTPHunter	No Prompt	<u>0.9268</u>	7.3	257.9
	Instructional	0.9248	7.3	255.5
	Declarative	**0.9269**	7.4	253.4
	keyword	0.9223	7.3	257.7

Bold indicates the best performance for each metric (e.g., Accuracy, Precision, etc.), while underlined denotes the second-best performance if applicable

to achieve consistent performance gains while maintaining inference efficiency.

6 Discussion

The experimental evaluation across five diverse cyber threat datasets reveals several key insights into the performance and generalisability of the proposed approach, AC_MAPPER. Most notably, the approach consistently achieved the highest performance, particularly excelling in challenging datasets such as HALdata and CAPEC, where

it outperformed all baselines in both accuracy and macro F1. These datasets are characterised by significant class imbalance and a high number of rare techniques, suggesting that the model's design—integrating input augmentation and class rebalancing—effectively enhances learning in low-resource, imbalanced scenarios. On more structured datasets like TRAM Bootstrap, AC_MAPPER achieved the highest scores observed across all experiments (93.59% accuracy and 93.78% macro F1), demonstrating not only peak performance but also strong learning efficiency as reflected in consistently lower loss curves. In contrast, baselines like TTPHunter and CTI-BERT, while strong in some datasets, showed inconsistent generalisation across others, underscoring AC_MAPPER's robustness.

Additionally, the ablation study on prompting revealed that prompting has mixed, dataset-dependent impact when using our exact templates—Instructional (text → question), Declarative (contextual cue), Minimal tag (keyword prefix), and No prompt. On structured corpora such as TRAM and TRAM Bootstrap, concise cues (especially the declarative or minimal tag formats) produced small but consistent gains in macro-F1 over the baseline, with negligible changes in inference time. In contrast, for noisier or more heterogeneous datasets such as CAPEC and HALdata, benefits were inconsistent: the instructional format occasionally helped disambiguate fragmentary sentences (HALdata), whereas additional contextual tokens from declarative or keyword cues could dilute the signal and slightly reduce performance. Differences on TTPHunter were minimal. Collectively, these findings indicate that effective prompt design for CTI is not one-size-fits-all; it is shaped by sentence structure, contextual density, and class overlap. Future work should pursue adaptive prompt selection (e.g., routing by input length/complexity) and learned prompt-tuning to realise robust gains without unintended regression.

7 Conclusion and further work

In this work, a novel approach is proposed for cyber threat technique classification that integrates domain-specific design with a general-purpose transformer backbone. Through extensive experiments across five publicly available datasets, we demonstrated that the AC_MAPPER approach consistently achieves higher performance, outperforming state-of-the-art baselines in both accuracy and macro F1 score.

AC_MAPPER approach showed particular strength in handling datasets with high class imbalance and varying technique diversity, such as HALdata and CAPEC, where it outperformed all other models by significant margins. Even in datasets where specialised baselines like TTPHunter and TTPXHunter excelled, our model remained highly competitive, highlighting its robust generalisation capabilities.

The comprehensive evaluation, supported by quantitative tables (4, 5, 6, 7, and 8) and visual comparison graphs (Figs. 5,6,7,8 and 9), underscores the reliability and adaptability of our approach across heterogeneous cyber datasets.

Future work can explore prompt engineering investigation in addition to extending the model to multi-label classification settings and incorporating external knowledge graphs to enhance the semantic understanding of threat techniques. This work contributes toward practical AI-driven threat intelligence systems.

Author Contributions M.A. conducted the experiments, developed the model, and led the implementation and evaluation of the proposed approach. A.A. contributed to the literature review, data analysis, and interpretation of results. S.J. supervised the research, provided critical feedback, and contributed to the structuring and revision of the manuscript. All authors reviewed and approved the final manuscript.

Funding Open access funding provided by Lund University.

Data availability Data is provided within the manuscript.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Institute IBMS-P (2023) Cost of a data breach report 2022
2. Nazir, A., He, J., Zhu, N., et al.: Empirical evaluation of ensemble learning and hybrid CNN-LSTM for IoT threat detection on heterogeneous datasets. J. Supercomput. **81**, 775 (2025). https://doi.org/10.1007/s11227-025-07255-1
3. Nazir, A., He, J., Zhu, N., et al.: Enhancing IoT security: a collaborative framework integrating federated learning dense neural networks and blockchain. Cluster Comput. **27**, 8367–8392 (2024). https://doi.org/10.1007/s10586-024-04436-0
4. Nazir, A., He, J., Zhu, N., et al.: Collaborative threat intelligence: enhancing IoT security through blockchain and machine learning integration. J. King Saud Univ. Comput. Inf. Sci. **36**, 101939 (2024). https://doi.org/10.1016/j.jksuci.2024.101939
5. Sun, N., Ding, M., Jiang, J., et al.: Cyber threat intelligence mining for proactive cybersecurity defense: a survey and new perspectives. IEEE Commun. Surv. Tutor. (2023). https://doi.org/10.1109/COMST.2023.3273282

6. Shackleford, D. (2015) Who's using cyberthreat intelligence and how. SANS Institute
7. Rahman, M.R., Hezaveh, R.M., Williams, L.: What are the attackers doing now? automating cyberthreat intelligence extraction from text on pace with the changing threat landscape: a survey. ACM Comput. Surv. **55**, 1–36 (2023)
8. Conti, M., Dargahi, T., Dehghantanha, A.: Cyber threat intelligence: challenges and opportunities. Springer (2018)
9. Strom, BE., Applebaum, A., Miller, DP., et al. (2018) Mitre att&ck: Design and philosophy. In: technical report. The MITRE corporation
10. Wagner, T.D., Mahbub, K., Palomar, E., Abdallah, A.E.: Cyber threat intelligence sharing: survey and research directions. Comput. Secur. **87**, 101589 (2019)
11. Strom, BE., Battaglia, JA., Kemmerer, MS., et al. (2017) Finding cyber threats with ATT&CK-based analytics. The MITRE corporation, Bedford, MA, Technical Report No MTR170202
12. MITRE (2025) Enterprise Matrix
13. Hubbard, J. (2020) Measuring and improving cyber defense using the MITRE ATT & CK framework. SANS Whitepaper
14. Corporation, M. (2021) Bad Rabbit - Enterprise | MITRE ATT&CK. Accessed 7 Oct 2025
15. Chamkar, S.A., Maleh, Y., Gherabi, N.: Security operations centers: use case best practices coverage and gap analysis based on MITRE adversarial tactics techniques and common knowledge. J. Cybersecur. Privacy **4**, 777–793 (2024)
16. Jisoo, J., Jung, S., Ahn, M., et al.: Research on quantitative prioritization techniques for selecting optimal security measures. IEEE Access (2024). https://doi.org/10.1109/ACCESS.2024.3433404
17. Al-Sada, B., Sadighian, A., Oligeri, G.: MITRE ATT&CK: state of the art and way forward. ACM Comput. Surv. **57**, 1–37 (2024)
18. Zhao, WX., Zhou, K., Li, J., et al. (2023) A survey of large language models. arXiv preprint arXiv:230318223
19. Chernyavskiy A, Ilvovsky D, Nakov P (2021) Transformers:"the end of history" for natural language processing? In: machine learning and knowledge discovery in databases. research track: European conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, Proceedings, Part III 21. 677–693
20. Khurana, D., Koli, A., Khatter, K., Singh, S.: Natural language processing: state of the art current trends and challenges. Multimed. Tools Appl. **82**, 3713–3744 (2023)
21. Rosenfeld, R.: Two decades of statistical language modeling: where do we go from here? Proc. IEEE **88**, 1270–1278 (2000)
22. Weizenbaum, J.: Eliza—a computer program for the study of natural language communication between man and machine. Commun. ACM **9**, 36–45 (1966)
23. Pahune, S., Chandrasekharan, M.: Several categories of large language models (LLMs): a short survey. Inter. J. Res. Appl. Sci. Eng. Technol. (2023). https://doi.org/10.22214/ijraset.2023.54677
24. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017) Attention is all you need. In: Guyon I, Luxburg U Von, Bengio S, et al (eds) advances in neural information processing systems. Curran Associates, Inc.
25. Pilipiszyn, A. (2021) GPT-3 powers the next generation of apps—openai.com
26. Gu, Y., Tinn, R., Cheng, H., et al.: Domain-specific language model pretraining for biomedical natural language processing. ACM Transactions Comput. Healthcare (HEALTH) **3**, 1–23 (2021)
27. Liu, Y., Ott, M., Goyal, N., et al. (2019) Roberta: a robustly optimized bert pretraining approach. arXiv preprint arXiv:190711692
28. Amine Ferrag, M., Battah, A., Tihanyi, N., et al. (2023) SecureFalcon: the next cyber reasoning system for cyber security. arXiv e-prints arXiv–2307
29. Aghaei, E., Niu, X., Shadid, W., Al-Shaer, E. (2022) Securebert: a domain-specific language model for cybersecurity. In: international conference on security and privacy in communication systems. 39–56
30. Bayer, M., Kuehn. P., Shanehsaz, R., Reuter, C. (2022) CySecBERT: a domain-adapted language model for the cybersecurity domain. arXiv preprint arXiv:221202974
31. Liao, X., Yuan, K., Wang, X., et al. (2016) Acing the ioc game: toward automatic discovery and analysis of open-source cyber threat intelligence. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. pp 755–766
32. Legoy, V. (2021) vlegoy/rcATT: a python app to predict ATT&CK tactics and techniques from cyber threat reports
33. Husari, G., Al-Shaer. E., Ahmed, M. et al. (2017) Ttpdrill: automatic and accurate extraction of threat actions from unstructured text of cti sources. In: Proceedings of the 33rd annual computer security applications conference. pp 103–115
34. Husari, G., Niu, X., Chu, B., Al-Shaer, E. (2018) Using entropy and mutual information to extract threat actions from cyber threat intelligence. In: 2018 IEEE international conference on intelligence and security informatics (ISI). pp 1–6
35. Zhang, H., Shen, G., Guo, C., et al.: Ex-action: automatically extracting threat actions from cyber threat intelligence report based on multimodal learning. Secur. Commun. Networks **2021**, 5586335 (2021)
36. Satvat, K., Gjomemo, R., Venkatakrishnan, VN. (2021) Extractor: extracting attack behavior from threat reports. In: 2021 IEEE European symposium on security and privacy (EuroS&P). pp 598–615
37. Alves PMMR, Geraldo Filho, PR., Gonçalves, VP. (2022) Leveraging BERT's power to classify TTP from unstructured text. In: 2022 workshop on communication networks and power systems (WCNPS). pp 1–7
38. Grigorescu, O., Nica, A., Dascalu, M., Rughinis, R.: Cve2att&ck: bert-based mapping of cves to mitre att&ck techniques. Algorithms **15**(9), 314 (2022)
39. Kurniawan, K., Ekelhart, A., Kiesling, E., et al.: KRYSTAL: knowledge graph-based framework for tactical attack discovery in audit data. Comput. Secur. **121**, 102828 (2022)
40. Li, Z., Zeng, J., Chen, Y., Liang, Z. (2022) AttacKG: constructing technique knowledge graph from cyber threat intelligence reports. In: European symposium on research in computer security. pp 589–609
41. Orbinato, V., Barbaraci, M., Natella, R., Cotroneo, D. (2022) Automatic mapping of unstructured cyber threat intelligence: an experimental study:(practical experience report). In: 2022 IEEE 33rd International symposium on software reliability engineering (ISSRE). pp 181–192
42. Kim, H., Kim, H.: Comparative experiment on TTP classification with class imbalance using oversampling from CTI dataset. Secur. Commun. Networks **2022**, 5021125 (2022)
43. MITRE (2023) Threat report ATT&CK mapper (TRAM). https://ctid.mitre.org/projects/threat-report-attck-mapper-tram/. Accessed 7 Oct 2025
44. Beltagy, I., Lo, K., Cohan, A. (2019) SciBERT: a pretrained language model for scientific text. arXiv preprint arXiv:190310676
45. Mendsaikhan, O., Hasegawa, H., Yamaguchi, Y., Shimada, H. (2020) Automatic mapping of vulnerability information to adversary techniques. In: the fourteenth international conference on emerging security information, systems and technologies SECUREWARE2020
46. Kuppa, A., Aouad, L., Le-Khac, N-A. (2021) Linking cve's to mitre att&ck techniques. In: Proceedings of the 16th international conference on availability, reliability and security. pp 1–12
47. Ampel, B., Samtani, S., Ullman, S., Chen, H. (2021) Linking common vulnerabilities and exposures to the mitre att&ck framework: a self-distillation approach. arXiv preprint arXiv:210801696

48. Branescu, I., Grigorescu, O., Dascalu, M.: Automated mapping of common vulnerabilities and exposures to mitre att&ck tactics. Information **15**, 214 (2024)
49. Panwar, A. (2017) igen: toward automatic generation and analysis of indicators of compromise (iocs) using convolutional neural network. Arizona State University
50. Abdeen, B., Al-Shaer, E., Singhal, A., et al. (2023) Smet: semantic mapping of cve to att&ck and its application to cybersecurity. In: IFIP annual conference on data and applications security and privacy. pp 243–260
51. Wang, L., Sun, L., Shang, D., et al. (2024) Automatic mapping based on CVE and ATT&CK. In: international conference on computer network security and software engineering (CNSSE 2024). pp 326–331
52. Liu, X., Tan, Y., Xiao, Z., et al.: Not the end of story: an evaluation of ChatGPT-driven vulnerability description mappings. Findings Assoc. Comput. Linguistics: ACL **2023**, 3724–3731 (2023)
53. You, W., Park, Y. (2024) Cyber-attack technique classification using two-stage trained large language models. arXiv preprint arXiv:241118755
54. Li, L., Huang, C., Chen, J.: Automated discovery and mapping ATT&CK tactics and techniques for unstructured cyber threat intelligence. Comput. Secur. **140**, 103815 (2024)
55. Rani, N., Saha, B., Maurya, V., Shukla, S.K.: Ttpxhunter: actionable threat intelligence extraction as ttps from finished cyber threat reports. Digital Threats: Res. Practice **5**, 1–19 (2024)
56. You, Y., Jiang, J., Jiang, Z., et al.: TIM: threat context-enhanced TTP intelligence mining on unstructured threat data. Cybersecurity **5**, 3 (2022)
57. Kumarasinghe, U., Lekssays, A., Sencar, HT., et al. (2024) Semantic ranking for automated adversarial technique annotation in security text. In: Proceedings of the 19th ACM Asia conference on computer and communications security. pp 49–62
58. Zhou, Y., Tang, Y., Yi, M., et al.: CTI view: APT threat intelligence analysis system. Secur. Commun. Networks **2022**, 9875199 (2022)
59. Aduma, J. (2021) SecBERT: domain-specific bert model for cybersecurity text
60. Ranade, P., Piplai, A., Joshi, A., Finin, T. (2021) Cybert: contextualized embeddings for the cybersecurity domain. In: 2021 IEEE international conference on big data (Big Data). pp 3334–3342
61. Park, Y., You, W. (2023) A pretrained language model for cyber threat intelligence. In: Proceedings of the 2023 conference on empirical methods in natural language processing: industry track. pp 113–122
62. Zhang, J., Wen, H., Li, L., Zhu, H. (2024) UniTTP: A unified framework for tactics, techniques, and procedures mapping in cyber threats. In: 2024 IEEE 23rd international conference on trust, security and privacy in computing and communications (TrustCom). pp 1580–1588
63. Hassan, E., Saber, A., El-kenawy, E-SM., et al. (2024) Early detection of black fungus using deep learning models for efficient medical diagnosis. In: 2024 international conference on emerging techniques in computational intelligence (ICETCI). IEEE, pp 426–431
64. Hassan, E., Saber, A., El-Sappagh, S., El-Rashidy, N.: Optimized ensemble deep learning approach for accurate breast cancer diagnosis using transfer learning and grey wolf optimization. Evol. Syst. **16**, 59 (2025). https://doi.org/10.1007/s12530-025-09686-w
65. Nguyen, K-D., Chu, H-C., Nguyen, Q-V., et al, (2024) From data to action: cti analysis and att&ck technique correlation. In: 2024 IEEE 23rd international conference on trust, security and privacy in computing and communications (TrustCom). pp 141–148
66. El Jaouhari, S., Tamani, N., Jacob, RI. (2024) Improving ML-based solutions for linking of CVE to MITRE ATT &CK techniques. In: 2024 IEEE 48th annual computers, software, and applications conference (COMPSAC). pp 2442–2447
67. Müller, R., Kornblith, S., Hinton, GE. (2019) When does label smoothing help? Adv Neural Inf Process Syst 32:
68. Della Penna S, Natella R, Orbinato V, et al (2025) CTI-HAL: a human-annotated dataset for cyber threat intelligence analysis. arXiv preprint arXiv:250405866
69. Rani N, Saha B, Maurya V, Shukla SK (2023) TTPHunter: automated extraction of actionable intelligence as ttps from narrative threat reports. In: Proceedings of the 2023 australasian computer science week. pp 126–134
70. CrowdStrike. (2021) CrowdStrike partners with MITRE engenuity center for threat-informed defense to develop TRAM
71. Sanh, V., Debut, L., Chaumond, J., Wolf, T. (2019) DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. ArXiv abs/1910.01108:
72. Yu, Y-C., Chiang, T-H., Tsai, C-W., et al. (2025) Primus: a pioneering collection of open-source datasets for cybersecurity LLM training