Vendor-Conditioned Contrastive Learning for Predicting Organizational Cyber Threat Targets

Benjamin Ampel
Department of Computer Science
Georgia State University
Atlanta, Georgia, USA
bampel@gsu.edu

***Abstract*—Cyberattacks cause billions of dollars in damage annually, with malicious hackers often sharing exploit code and techniques on underground forums. Identifying which organizations are targeted by these exploits is critical for proactive Cyber Threat Intelligence (CTI). To address that gap, we propose Temporal Representation and Classification of Exploits (TRACE), a vendor-conditioned contrastive learning framework built on CySecBERT that jointly optimizes organizational target classification and vendor-coherent representations while evaluating robustness under temporal distribution shift. Unlike prior work limited to small, single-source datasets, we leverage a large-scale, multi-source corpus spanning 9 exploit databases and hacker forums, comprising 352,866 posts collected over three decades, yielding a 129,126-sample dataset across seven organizational categories. In the temporal out-of-distribution evaluation, TRACE achieves macro F1=97.00%, substantially outperforming 17 benchmark classical ML methods, deep learning with GloVe/FastText embeddings, and pretrained transformer models.**

***Index Terms*—hacker forum, exploit classification, organization risk, contrastive learning, temporal distribution shift, cyber threat intelligence**

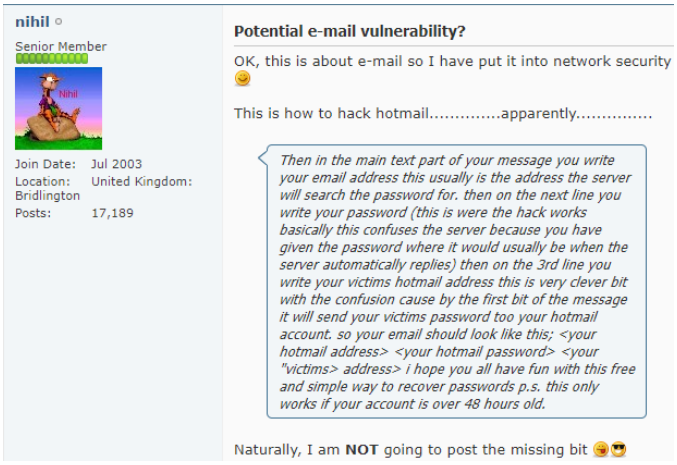

Fig. 1. Example hacker forum post discussing an exploit technique targeting a specific organization's email service.

I. Introduction

Organizations, governments, and individuals face an ever-increasing volume of cyberattacks, with global cybercrime damages estimated at $10.5 trillion in 2025 and projected to reach $15.6 trillion annually by 2029 [1]. To combat this threat, organizations invest heavily in Cyber Threat Intelligence (CTI, the collection, analysis, and dissemination of information about cyber threats) [13]. A key source of proactive CTI lies in hacker forums, where malicious actors congregate to share exploit source code, discuss attack techniques, and trade tools targeting specific organizations [2]. An example of such a forum post is shown in Figure 1.

Prior work on hacker forum exploit analysis has focused primarily on exploit identification [3], MITRE ATT&CK mapping [8], and cross-lingual threat detection [7]. However, a critical gap remains. Which organizations are being targeted by these exploits? Understanding organizational targeting enables companies and sector-specific agencies to prioritize defenses against threats tailored to their industry.

This gap is especially consequential because hacker communities are not static. As new vendors, products, vulnerabilities, and attack techniques emerge, the language used to describe exploits changes as well. A model trained on historical forum posts may therefore learn patterns that become less reliable when applied to future posts. Organizational target prediction must therefore be evaluated under temporal drift, where the vocabulary, targets, and exploit patterns observed during training may differ from those encountered during deployment.

To address this gap, in this paper, we propose Temporal Representation and Classification of Exploits (TRACE), an organizational detection framework with two key contributions:

1) **Large-scale multi-source corpus**: A dataset of labeled 129,126-samples labeled dataset to enable downstream tasks; and
2) **Vendor-conditioned contrastive learning model**: A CySecBERT-based architecture with a supervised contrastive objective over vendor groups, achieving F1=97.00% on temporal OOD evaluation.

II. Related Work

A. Hacker Community Analytics

A survey [13] systematizes the methodologies for studying cybercrime communities, establishing hacker forums as a primary data source for proactive CTI [2]. In this space,

exploit analysis has progressed rapidly. Literature explored deep transfer learning for automated exploit labeling from forum posts [3] and extended exploit detection to non-English forums via adversarial cross-lingual knowledge transfer [7]. Subsequent work mapped exploit code to the MITRE ATT&CK framework using multi-label transformers [8] and profiled the evolution of exploit-sharing hackers through graph embeddings [6]. Finally, NER-based methods have extracted protected health information from multilingual hacker communities [11].

Importantly, a systematic study of dark web language demonstrated hacker communities use unique vocabulary that general-purpose Natural Language Processing (NLP) models struggle to generalize to [25]. Related literature attempted to improve NLP model generalizability through GloVe [4], distilling contextual hacker community embeddings [5], and cyberesecurity domain-specific language models [12], [26], [27]. These models consistently achieve state-of-the-art results on cybersecurity NLP tasks, confirming that cybersecurity domain-specific pretraining is essential for hacker community analytics.

However, prior work primarily classifies exploits by type or tactic rather than organizational target. Thus, while cybersecurity-specific pretraining improves lexical and contextual coverage, it does not explicitly exploit the vendor-level structure inherent in exploit disclosures, nor does it establish whether organizational target classifiers remain reliable under forward temporal shift. This motivates our vendor-conditioned contrastive fine-tuning objective.

B. Contrastive Learning and Temporal Robustness

Contrastive learning has emerged as a powerful paradigm for learning discriminative representations. Literature generalized the self-supervised contrastive framework of SimCLR [15] to the fully-supervised setting [14]. This new Supervised Contrastive (SupCon) loss leverages label information to pull same-class embeddings together while pushing different-class embeddings apart [14]. This approach outperformed standard cross-entropy on image classification benchmarks. Subsequent literatuer adapted SupCon learning to pretrained language model fine-tuning, demonstrating improved generalization on text classification tasks (including sentiment analysis and natural language inference) [16]. It is often observed that contrastive objectives yield state-of-the-art sentence embeddings for downstream NLP tasks [17].

However, a growing body of work has documented the challenge of temporal distribution shift in NLP. For example, research found an average 20% performance degradation when NLP models are evaluated on temporally out-of-distribution (OOD) data [28]. This pattern held in several settings, finding that temporal drift erodes classifier performance even when the underlying task remains identical [29].

In the temporal domain, contrastive methods have been applied to learn representations that capture temporal structure [18], [19]. For example, Temporal Neighborhood Coding adapts temporal proximity as a self-supervised signal for time-series representation learning, while Zhang et al. [19] introduced time-frequency consistency as a contrastive objective for time-series pretraining. However, these approaches are designed primarily for time-series data and do not address the representational challenge posed by cybersecurity text, where exploit posts contain vendor names, product identifiers, version strings, vulnerability references, and code fragments that evolve over time. Existing contrastive approaches also do not specify how auxiliary metadata in exploit disclosures can be used to learn representations that remain discriminative for organizational target prediction under temporal distribution shift.

III. Research Design

Our research design is organized around three challenges in organizational target prediction from hacker community data. The task requires a large and diverse exploit corpus, a high-precision procedure for mapping vendor and product references to organizational categories, and an evaluation protocol that reflects temporal drift in real deployments. Accordingly, we construct a multi-source corpus, extract organizational targets through a curated gazetteer, split the data forward in time, and evaluate TRACE against classical machine learning (ML), deep learning (DL), and pretrained transformer baselines.

A. Dataset

Our design requires that we construct a corpus that captures exploit discussions across heterogeneous CTI-relevant source layers. To create this dataset, we leveraged the HackerSignal dataset (comprised of 35 public sources and 8.7 million posts collected over three decades) [10]. We provide the overall summary of HackerSignal in Table I.

TABLE I
Corpus Summary by Source Layer

Source Layer	Sources	Posts	Date Range
Hacker Forums	21	4,811,505	2001–2026
Exploit Databases	5	3,048,765	1990–2026
Vuln. Advisories	7	385,039	2002–2026
Darknet Markets	1	488,989	2014–2015
Fix Commits	1	8,362	1999–2022
Total	**35**	**8,742,660**	**1990–2026**

For organizational target prediction, we opted to focus on the English exploit-rich sources most likely to reference targeted organizations: exploit databases (e.g., ExploitDB, 0day.today), bug bounty platforms (e.g., HackerOne), and hacker forums (e.g., HackForums, Go4Expert). This HackerSignal subset accounts for 352,866 posts for entity extraction. Exploit titles and descriptions in these sources often follow predictable patterns (e.g., *VendorName ProductName Version – VulnType*), enabling precise vendor extraction via a gazetteer (a curated dictionary of known entity names) of 150+ vendor and product name patterns organized by industry

category. Each post is matched against the title and the opening 500 characters of description text. This gazetteer-based approach yields higher precision than general-purpose NER on noisy exploit code, where register names and technical terms are frequently misclassified as organizations.

Extracted vendors are normalized and mapped to industry categories (e.g., Networking, Gaming). Categories with fewer than 100 samples were removed. Of the 352,866 posts processed, 129,126 (36.6%) contained identifiable organizational targets, forming our gold-standard dataset of vendors mentioned in known exploits (Table II).

TABLE II
GOLD-STANDARD DATASET BY ORGANIZATION CATEGORY

Category	Count	Pct.
CMS / Open Source	44,791	34.7%
Enterprise Software	44,460	34.4%
Web Platform	30,294	23.5%
Networking / IoT	4,672	3.6%
Mobile	3,272	2.5%
Gaming	1,027	0.8%
Database	610	0.5%
Total	**129,126**	**100%**

B. Preprocessing and Temporal Splitting

Each post's raw text is cleaned by removing non-UTF-8 characters and normalizing whitespace. The exploit source code is retained because it contains valuable organizational references (e.g., target hostnames, library imports). We split the data temporally to avoid future leakage. Posts before 2022 form the training set (125,963 samples), 2022–2023 posts form the validation set, and 2024+ posts form the test set. Posts without timestamps are discarded. This temporal out-of-distribution (OOD) protocol requires models to generalize to future and potentially unseen exploit patterns. This data as used as input into our TRACE framework

C. Temporal Representation and Classification of Exploits (TRACE)

We propose TRACE, a vendor-conditioned contrastive learning framework that optimizes organizational target classification while encouraging vendor-coherent representations. The architecture consists of three components, each motivated by the characteristics of exploit text classification under temporal distribution shift.

1. Shared encoder. We build on CySecBERT [12], a BERT-base encoder pretrained on cybersecurity corpora. Domain-adapted pretraining is useful for exploit text that contains specialized vocabulary such as CVE identifiers, shell commands, obfuscated payloads, product names, and version strings. The encoder processes tokenized exploit text and produces a contextualized `[CLS]` representation $\mathbf{h} \in \mathbb{R}^{768}$.

2. Classification head. A single linear layer $W_c \in \mathbb{R}^{768\times 7}$ maps $\mathbf{h}$ to class logits, trained with standard cross-entropy loss $\mathcal{L}_{CE}$. We keep the TRACE classification head minimal so that representational capacity remains concentrated in the shared encoder, where the contrastive objective can shape the learned representation.

3. Contrastive projection head. A two-layer MLP ($768 \to 768 \to 128$ with ReLU) projects $\mathbf{h}$ into a contrastive embedding space, followed by ℓ_2 normalization: $\mathbf{z} = \mathrm{normalize}(g(\mathbf{h})) \in \mathbb{R}^{128}$. Following SupCon learning [14], the contrastive objective pulls positive examples together and pushes negative examples apart. In TRACE, positives are defined by vendor identity rather than by organizational category labels or temporal bins. A SupCon loss pulls posts from the same *vendor* together while pushing posts from different vendors apart:

$$\mathcal{L}_{SC} = -\frac{1}{|B|}\sum_{i\in B}\frac{1}{|P_i|}\sum_{p\in P_i}\log\frac{\exp(\mathbf{z}_i\cdot\mathbf{z}_p/\tau)}{\sum_{j\neq i}\exp(\mathbf{z}_i\cdot\mathbf{z}_j/\tau)} \quad (1)$$

where B is a mini-batch, $P_i = \{p \in B : \mathrm{vendor}(p) = \mathrm{vendor}(i), p \neq i\}$ is the set of same-vendor positives, and $\tau = 0.07$ is the temperature.

The rationale for vendor rather than label-based grouping is twofold. First, exploit posts associated with the same vendor or product family often share vocabulary, code patterns, vulnerability identifiers, and version-specific language that may cut across broader organizational categories. Second, vendor grouping provides a more granular auxiliary supervision signal than the seven-category target label. The classification loss separates broad organizational categories, while the contrastive loss encourages the embedding space to preserve vendor-level structure.

The total training objective combines both losses:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda\mathcal{L}_{SC}, \quad \lambda = 0.1 \quad (2)$$

where $\lambda = 0.1$ down-weights the contrastive term so that classification remains the primary objective. At inference, the classification head produces predictions. The projection head is used only to shape representations during training. Training uses AdamW [32] with learning rate 2×10^{-5}. In our implementation, TRACE is fine-tuned for three epochs and is evaluated using the temporal out-of-distribution test protocol. Figure 2 illustrates the full TRACE pipeline from corpus to evaluation.

D. Benchmark Models

We compare TRACE against benchmark models spanning three paradigms. All models are trained on the full 125,963 training samples and evaluated on the temporal OOD test set.

- **Classical ML** (4 models). TF-IDF features (unigrams + bigrams, sublinear TF) fed to Naïve Bayes, Logistic Regression, Decision Tree, and LinearSVC.
- **Deep Learning** (14 models). Seven architectures (RNN, GRU, LSTM, BiLSTM, CNN-LSTM, BiLSTM with attention, and a shallow Transformer encoder [30]) each trained with two embedding initializations. First, a generic GloVe 6B 300d [20]. Second, a FastText 300d [21] trained on exclusively on our exploit training

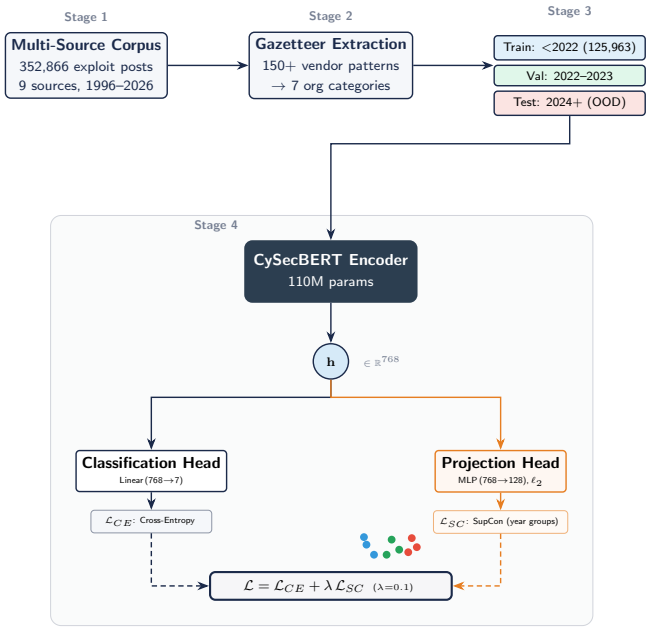

Fig. 2. TRACE research design: multi-source corpus → gazetteer extraction → temporal split → dual-head CySecBERT model with vendor-conditioned contrastive learning (novelty in orange) → temporal OOD evaluation.

corpus. All use 256-dimensional hidden states, 0.3 dropout, the Adam optimizer (lr=10^{-3}), early stopping (patience=2), and train for up to 20 epochs.

- **Pretrained Transformers** (LLMs). BERT [22], RoBERTa [23], ModernBERT [24], and CySecBERT [12], each fine-tuned with early stopping for up to 10 epochs.

IV. Experiments and Results

A. Evaluation Metrics

We report Accuracy, macro F1-Score, macro Precision, and macro Recall on the temporal OOD test set. Models were assessed via paired bootstrap tests (1,000 resamples) to determine the significance of TRACE relative to each benchmark.

B. Results

Table III presents results across all model groups.

Our results yielded several notable insights. First, TRACE achieved the strongest performance (F1=97.00%, Acc=98.65%). The vendor-conditioned contrastive objective encourages the CySecBERT encoder to learn vendor-coherent representations that generalize to future exploit patterns. Second, domain-trained embeddings matter more than architecture for deep learning baselines. FastText embeddings (trained on our exploit corpus) consistently outperformed pretrained GloVe across all architectures (e.g., BiLSTM: F1 81.84% vs. 63.11%). FastText's subword representations handle the out-of-vocabulary tokens prevalent in exploit text (CVE identifiers, tool names, obfuscated strings) and its corpus-specific training captures the distributional semantics

TABLE III
Experiment Results: Temporal OOD Evaluation (%)

Group	Model	Acc.	F1	Prec.	Rec.
Class. ML	Naïve Bayes***	79.52	40.01	55.26	39.45
	Decision Tree***	88.55	62.43	70.26	58.58
	Logistic Reg.***	91.36	73.42	74.16	73.19
	SVM (Linear)***	92.65	75.69	75.50	76.26
DL + GloVe	RNN***	22.42	10.42	11.87	18.54
	LSTM***	71.75	57.33	60.52	57.95
	BiLSTM***	79.42	63.11	64.02	64.39
	GRU***	81.31	65.18	65.32	66.83
	CNN-LSTM***	77.36	68.20	69.64	70.88
	BiLSTM-Attn***	82.60	71.65	71.20	74.51
	Transformer***	77.42	72.38	82.31	69.36
DL + FastText	RNN***	65.64	22.46	22.16	23.08
	LSTM***	90.87	61.45	65.78	59.73
	CNN-LSTM***	94.65	72.33	72.87	71.89
	Transformer***	96.33	77.18	79.84	75.74
	BiLSTM-Attn***	96.43	80.54	80.35	80.80
	GRU***	94.92	81.18	88.06	77.34
	BiLSTM***	93.25	81.84	87.16	79.85
LLM	BERT***	97.78	89.29	90.49	91.91
	RoBERTa***	96.49	84.36	87.84	82.77
	ModernBERT***	97.73	88.11	89.93	86.54
	CySecBERT**	97.51	94.45	98.35	92.32
	TRACE (Ours)	**98.65**	**97.00**	**96.63**	**97.38**

Paired bootstrap vs. TRACE (1,000 resamples):
$^{***}p<0.001$, $^{**}p<0.01$, $^{*}p<0.05$, $^{\dagger}p<0.10$

of hacker discourse. This suggests that domain-specific data is a powerful factor across all model families.

Third, among classical ML models, SVM with TF-IDF (F1=75.69%) outperformed all GloVe-based DL models, confirming that sparse n-gram features capture vendor-specific terminology that generalizes across time. The FastText-based models and pretrained transformers surpass SVM, demonstrating that distributed representations trained on large corpora can provide superior generalization when vocabulary coverage is adequate. Fourth, the vanilla RNN failed to converge reliably with either embedding (F1=10.42% GloVe, 22.46% FastText), confirming the known vanishing gradient problem on long exploit text sequences.

Finally, domain-specific pretraining is decisively necessary. CySecBERT (F1=94.45%) outperforms BERT (89.29%), ModernBERT (88.11%), and RoBERTa (84.36%). CySecBERT's vocabulary, trained on cybersecurity corpora, preserves exploit-specific tokens (CVE IDs, function names, shell commands) as meaningful units rather than fragmenting them into uninformative subwords. Among general-purpose models, RoBERTa's BPE tokenizer fragments these tokens most aggressively, while ModernBERT's architectural improvements (rotary embeddings, flash attention) provide marginal gains but do not overcome the vocabulary bottleneck. TRACE's vendor-conditioned contrastive objective further improves upon standalone CySecBERT fine-tuning (F1:

97.00% vs. 94.45%, a 2.55% gain), demonstrating that the vendor-conditioned SupCon loss provides a substantial complementary signal beyond domain-adapted initialization.

C. Per-Category Analysis

To understand where TRACE's vendor-conditioned contrastive objective improves over standalone CySecBERT fine-tuning, Table IV compares per-category F1 scores on the temporal OOD test set.

TABLE IV
PER-CATEGORY F1 SCORE: CYSECBERT VS. TRACE

Category	Test	CySecBERT	TRACE	ΔF1
CMS / Open Src.	272	96.09	98.55	+2.46
Enterprise Softw.	350	97.88	98.01	+0.13
Web Platform	1,108	98.23	99.32	+1.09
Network / IoT	48	97.87	93.75	−4.12
Mobile	41	98.77	97.56	−1.21
Gaming	30	72.34	91.80	+19.46
Database	2	100.00	100.00	0.00
Macro Avg	**1,851**	**94.45**	**97.00**	**+2.55**

TRACE outperforms CySecBERT across 5 of 7 categories, with the largest gain in Gaming (+19.46% F1). CySecBERT struggles in this low-support category ($F1 = 72.34\%$, 30 test samples), misclassifying nearly half of the instances. TRACE's vendor-conditioned contrastive objective raised Gaming F1 to 91.80% by learning vendor-coherent representations that better separate rare classes under distribution shift. TRACE also improves CMS/Open Source ($+2.46\%$) and Web Platform ($+1.09\%$). The slight regressions on Networking/IoT (-4.12%) and Mobile (-1.21%) reflect trade-offs in the macro-averaged objective, though both categories retain strong absolute performance ($> 93\%$ F1).

TRACE makes only 25 errors out of 1,851 test samples (98.65% accuracy). Examining the misclassifications reveals that most errors occur at genuine category boundaries. For example, the Ruckus IoT Controller (Networking/IoT) is predicted to be CMS/Open Source. Ruckus devices run embedded Linux with web management interfaces, sharing vocabulary with open-source CMS exploits. Similarly, GitLab (Web Platform) is classified as a CMS/Open Source, reflecting its dual identity as both a web application and an open-source project. VMware Cloud Director (Web Platform) is predicted as Enterprise Software, a reasonable confusion given VMware's enterprise positioning. These boundary cases suggest that the 7-category taxonomy, while practically useful, imposes hard boundaries on organizations that straddle multiple categories. Figure 3 visualizes this effect. CySecBERT's representations exhibit substantial inter-category overlap, whereas TRACE produces more tightly clustered, well-separated embeddings.

D. Discussion

Our results yield several practical implications for CTI systems and the broader NLP community. First, TRACE can enable interested stakeholders with a mechanism to quickly understand what exploits are targeting their industry. Second, across all three model families, the single most significant factor is the choice of embedding. Corpus-trained FastText embeddings outperform off-the-shelf GloVe by 10-19% in F1 across identical architectures. CySecBERT outperforms BERT by 5.16% in F1 despite sharing the same architecture. This pattern echoes findings in dark web NLP [25], where general-purpose models struggle with domain-specific jargon. It also aligns with the broader observation that cybersecurity domain adaptation is essential for specialized technical language [26], [27].

Second, our temporal OOD evaluation, motivated by work on distribution shift [28], [29], provides a realistic assessment choice for this domain. Exploit language evolves as new software is released, vulnerabilities are disclosed, and attack techniques shift. A model that memorizes 2019-era exploit patterns (e.g., Flash Player exploits) provides little value when deployed in 2025. Our temporal split ensures that reported metrics reflect genuine forward-looking generalization.

However, we our approach has limitations. Our gazetteer labeling, while high-precision, is imperfect. Vendors not in the gazetteer are missed, and the 7-category taxonomy aggregates organizations into broad industry groups. The Database and Gaming categories have very few test samples (2 and 30, respectively), limiting the reliability of per-category metrics for these classes. Further, our vendor-conditioned contrastive objective assumes vendor identities are meaningful groupings. Alternative metadata signals, such as project families or product families, may yield further improvements.

V. CONCLUSION

We presented TRACE, a vendor-conditioned contrastive learning framework for predicting the type of organization targeted by hacker-forum exploits. Leveraging a corpus of 8.7 million posts from 35 sources, we constructed a 129,126-sample gold-standard dataset. TRACE's CySecBERT encoder with supervised contrastive learning across vendor groups achieves F1=97.00% with only 25 misclassifications out of 1,851 test samples, outperforming all 17 benchmarks in the temporal OOD evaluation. Our analysis revealed that the choice of embedding is critical. This model has implications for stakeholders, enabling them to quickly identify new exploits targeting their industry. Future work should deploy TRACE across the full 8.7 million-post corpus to generate large-scale temporal intelligence on shifting organizational targeting patterns, integrate CVE linkage to connect exploit discussions to specific vulnerabilities, and extend to multilingual hacker communities.

REFERENCES

[1] Statista, "Estimated cost of cybercrime worldwide 2018–2029," *Statista Research Department*, 2024.

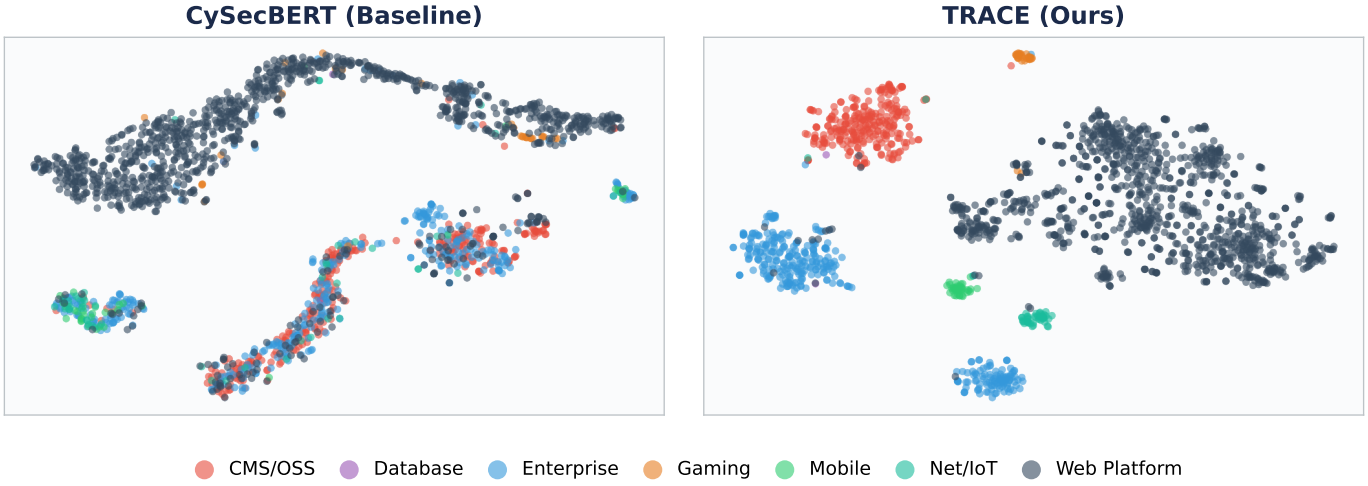

Fig. 3. t-SNE projections of test-set [CLS] representations. CySecBERT baseline embeddings (left) show substantial category overlap. TRACE embeddings (right) form well-separated clusters.

[2] S. Samtani, H. Zhu, and H. Chen, "Proactively identifying emerging hacker threats from the dark web," *ACM Trans. Privacy Security*, vol. 23, no. 4, pp. 1–33, Aug. 2020.

[3] B. M. Ampel, S. Samtani, H. Zhu, S. Ullman, and H. Chen, "Labeling hacker exploits for proactive cyber threat intelligence: A deep transfer learning approach," in *Proc. IEEE ISI*, 2020, pp. 1–6.

[4] B. M. Ampel, S. Samtani, H. Zhu, and H. Chen, "Creating Proactive Cyber Threat Intelligence with Hacker Exploit Labels: A Deep Transfer Learning Approach," in *Management Information Systems Quarterly*, 2024, vol. 48, num. 1, pp. 137-166.

[5] B. M. Ampel and H. Chen, "Distilling contextual embeddings into a static word embedding for improving hacker forum analytics," in *Proc. IEEE ISI*, 2021, pp. 1–6.

[6] K. Otto, B. M. Ampel, S. Samtani, H. Zhu, and H. Chen, "Exploring the evolution of exploit-sharing hackers: An unsupervised graph embedding approach," in *Proc. IEEE ISI*, 2021, pp. 1–6.

[7] M. Ebrahimi, S. Samtani, Y. Chai, and H. Chen, "Detecting cyber threats in non-English hacker forums: An adversarial cross-lingual knowledge transfer approach," in *Proc. IEEE S&P Workshop on Deep Learning and Security*, 2020, pp. 1–7.

[8] B. M. Ampel, B. Vahedi, S. Samtani, and H. Chen, "Mapping exploit code on paste sites to the MITRE ATT&CK framework: A multi-label transformer approach," in *Proc. IEEE ISI*, 2023, pp. 1–6.

[9] B. M. Ampel, K. Otto, S. Samtani, and H. Chen, "Disrupting ransomware actors on the Bitcoin blockchain: A graph embedding approach," in *Proc. IEEE ISI*, 2023, pp. 1–6.

[10] B. M. Ampel and S. Samtani, "HackerSignal: A large-scale multi-source dataset linking hacker community discourse to the CVE vulnerability lifecycle," *arXiv preprint arXiv:2605.03158*, 2026.

[11] C. Dacosta, B. M. Ampel, M. Hashim, and H. Chen, "Automatic extraction of protected health information from multilingual hacker communities," in *Proc. HICSS*, 2026, pp. 1–10.

[12] M. Bayer, P. Kuehn, R. Shanehsaz, and C. Reuter, "CySecBERT: A domain-adapted language model for the cybersecurity domain," *ACM Trans. Privacy Security*, vol. 27, no. 2, 2024.

[13] J. Hughes, S. Pastrana, A. Hutchings, S. Afroz, S. Samtani, W. Li, and E. S. Marin, "The art of cybercrime community research," *ACM Computing Surveys*, vol. 56, no. 6, pp. 1–26, 2024.

[14] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Proc. NeurIPS*, 2020, pp. 18661–18673.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. ICML*, 2020, pp. 1597–1607.

[16] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," in *Proc. ICLR*, 2021.

[17] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proc. EMNLP*, 2021, pp. 6894–6910.

[18] S. Tonekaboni, D. Eytan, and A. Goldenberg, "Unsupervised representation learning for time series with temporal neighborhood coding," in *Proc. ICLR*, 2021.

[19] X. Zhang, Z. Zhao, T. Tsiligkaridis, and M. Zitnik, "Self-supervised contrastive pre-training for time series via time-frequency consistency," in *Proc. NeurIPS*, 2022.

[20] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.

[21] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. ACL*, vol. 5, pp. 135–146, 2017.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[23] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[24] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, N. Cooper, G. Adams, J. Howard, and I. Poli, "Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context fine-tuning and inference," *arXiv preprint arXiv:2412.13663*, 2024.

[25] Y. Jin, E. Jang, Y. Lee, S. Shin, and J.-W. Chung, "Shedding new light on the language of the dark web," in *Proc. NAACL-HLT*, 2022, pp. 5621–5637.

[26] E. Aghaei, X. Niu, W. Shadid, and E. Al-Shaer, "SecureBERT: A domain-specific language model for cybersecurity," in *Proc. EAI SecureComm*, Springer LNICST, vol. 462, 2022.

[27] Y. Park and W. You, "A pretrained language model for cyber threat intelligence," in *Proc. EMNLP Industry Track*, 2023, pp. 113–122.

[28] H. Yao, C. Choi, B. Cao, Y. Lee, P. W. Koh, and C. Finn, "Wild-Time: A benchmark of in-the-wild distribution shift over time," in *Proc. NeurIPS Datasets and Benchmarks*, 2022.

[29] Y. Guo, C. Hu, and Y. Yang, "Predict the future from the past? On the temporal data distribution shift in financial sentiment classifications," in *Proc. EMNLP*, 2023, pp. 1029–1038.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 5998–6008.

[31] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "MITRE ATT&CK: Design and philosophy," *MITRE Technical Report*, 2020.

[32] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.