A CROSS-DOMAIN BENCHMARK AND EVALUATION OF CYBERSECURITY-SPECIFIC BERT MODELS

Laeeba Javed, Aasim Zafar

Department of Computer Science, Aligarh Muslim University, India

ABSTRACT

Cybersecurity language models are typically evaluated under fragmented ways, which impedes meaningful comparison and operational comprehension. Existing cybersecurity-specific BERT models are often tested in isolation, with inconsistent preprocessing, tokenization, and evaluation procedures. This paper presents a unified cross-domain benchmarking study for systematically evaluating cybersecurity-adapted BERT models under identical experimental conditions. The evaluation spans CTI, phishing, logs, and CVE domains using CTI-BERT, SecureBERT, CySecBERT, and SecBERT. Results reveal strong performance convergence across models and highlight domain-driven failure modes rather than architectural superiority. To examine real-world resilience, the study expands on this paradigm with zero-shot and few-shot cross-domain evaluations, revealing asymmetric transfer behavior and domain-dependent adaptation efficiency. A controlled training method ablation is also performed, indicating that aggressive optimization does not always increase performance and can decrease stability in semantically rich domains. Stress filtering further exposes brittle reliance on lexical shortcuts and limited semantic grounding. These findings provide practical guidance for model–domain alignment and real-world cybersecurity deployment. The findings of this study are especially relevant to network security and operational situations when cybersecurity models are deployed across heterogeneous data streams

KEYWORDS

Cybersecurity , NLP, BERT, Threat Intelligence, Domain-Adaptive Pretraining, Network Security.

1. INTRODUCTION

Contemporary cybersecurity analysis is increasingly being carried out as part of larger distributed networked systems rather than as standalone analytic components [1] [2]. Cyber threat intelligence pipelines gather a variety of textual streams from sensors, endpoints, intrusion detection systems, and external sources that are spread across complex network infrastructures [1] [3]. In such cases, the practical utility of language models is determined not only by task-level accuracy, but also by predictability, robustness during domain shift, and stability across diverse data sources. As cybersecurity language models become more integrated into network operations, monitoring workflows, and automated response mechanisms, understanding their behavior across different domains becomes a system-wide problem [4] [5].

The primary challenge that modern cybersecurity professionals struggle to address is how cyber threats are getting more advanced and evolving rapidly [1] [6]. This issue has intensified since 2019 and 2020 as accelerated digitization, remote work, and cloud-based services significantly expanded the attack surface, enabling adversaries to exploit vulnerabilities more rapidly and adaptively while leveraging technological advances to dodge existing detection systems [1] [3].

Adding to this, the scale, speed, and diversity of digital data have become increasingly difficult to manage [4] [3]. Threat intelligence extraction is severely hampered by the continuous generation of large-scale runtime logs and heterogeneous security data from sensors, endpoint agents, and highly unstructured external sources like social media platforms, blogs, threat feeds, dark web forums, and incident reports [1] [7] [2]. As a result, general-purpose language models (LMs) have proven ineffective for this domain due to the highly specialized nature of cybersecurity documentation and the prevalence of threats described in unstructured textual formats [4] [8]. Cybersecurity terms are frequently uncommon in general English (e.g., ransomware, API, keylogger) or function as homographs (words with multiple meanings) that have completely different connotations in a security context (e.g., honeypot, patch, virus) compared to standard English corpora [4] [8]. This situation has rendered traditional, static defense systems inadequate, necessitating the deployment of complex automated techniques like Deep Learning (DL) and Natural Language Processing (NLP) [1] [8] [9].

Despite the effectiveness of general-purpose language models in open-domain natural language processing, their direct application to cybersecurity text is fundamentally ineffective [4].Technical terminology, acronyms, exploit identifiers, and malware family names are examples of highly specialized vocabulary used in cybersecurity documentation that is either rare or non-existent in regular English corpora [10] [11]. As a result, generic language models face a significant domain mismatch, resulting in frequent tokenization failures and insufficient lexical representations of essential security concepts [11]. This difficulty is exacerbated by semantic ambiguity, as several regularly used cybersecurity terms, such as virus, worm, payload, and patch have meanings that range significantly from their general English usage [8]. Without domain understanding, general models often misinterpret such phrases, yielding inappropriate contextual associations [5]. Furthermore, cybersecurity text has structural differences from the clean, narrative-style data used for pretraining generic language models [12]. Threat reports and operational data are often lengthy, noisy, and poorly structured, combining unstructured prose with semi-structured system logs, indicators of compromise, and artifacts from online forums or social media [2]. These properties involve long-range dependencies, uneven formatting, and technical abstractions that ordinary models struggle to represent effectively [13] [14]. As a result, generic language models often fail to capture the full semantics of cybersecurity text, resulting in poor performance for downstream tasks like entity recognition, threat classification, and anomaly detection [13]. These constraints show that using general-purpose language models alone is insufficient for reliable cybersecurity analysis [10].

Building on these constraints, prior studies have increasingly used Domain-Adaptive Pretraining (DAPT) and suggested a number of cybersecurity-specific BERT models to better capture specialized terminology and domain semantics that generic language models are unable to [15]. While these approaches show significant in-domain performance gains [10], they are typically designed for specific cybersecurity sub-domains and evaluated in isolation, with highly inconsistent experimental pipelines that include customized preprocessing, tokenization strategies, loss functions, and evaluation metrics [16] [17]. As a result, claimed improvements are task-specific and scientifically incomparable, with cross-domain transferability generally unknown, leaving practitioners without credible empirical guidance for model selection in real-world operational situations.

Figure 1 summarises this evolution over time to depict these various approaches. To address this gap, our work introduces a unified cross-domain benchmarking study that evaluates multiple cybersecurity-specific BERT models using an identical fine-tuning and evaluation pipeline across multiple cybersecurity text domains. This work provides a reproducible foundation for comparative research by enabling controlled analysis of cross-domain behavior, model-domain

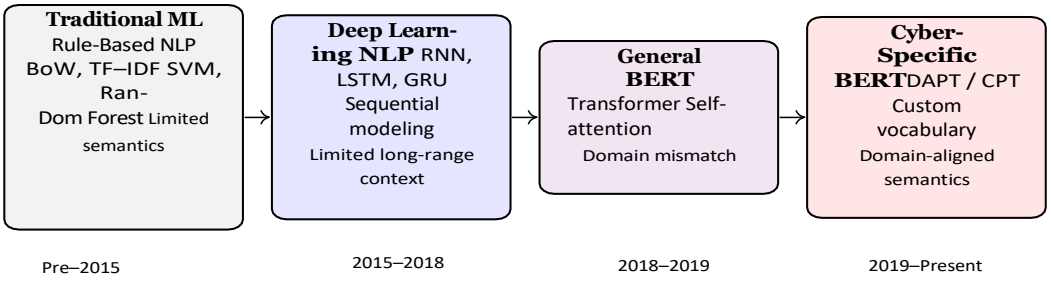


Figure 1: Evolution of NLP models in cybersecurity from traditional machine learning approaches to domain-specific Transformer architectures.

alignment, and practical deployment implications. To the best of our knowledge, this is the first systematic evaluation of cybersecurity-focused BERT models conducted under fully consistent experimental conditions

2. RELATED WORK

Early cybersecurity defense systems focused on rule-based procedures and predefined threat signatures, which were only effective against known attacks and were insufficient for identifying novel or advanced persistent threats [18]. To automate tasks like malware detection [19], intrusion analysis [20], and vulnerability prediction [17], classical machine learning approaches using handcrafted features like TF-IDF [21] [1] and classifiers like Support Vector Machines [18], Na¨ıve Bayes [22], and K-Nearest Neighbors [23] were introduced [24].However, these count-based text representations failed to capture contextual dependencies and semantic relationships between events, frequently omitting significant correlations inherent in cybersecurity text streams [25] [7] [13]. To overcome these constraints, sequential deep learning models like Recurrent Neural Networks and Long Short-Term Memory networks were employed, as they could model event sequences more effectively in tasks like log anomaly detection and threat report entity recognition [26] [27] [25].

Even with notable advances over classical methods, RNN and LSTM based models continued to struggle with long-range dependencies, unidirectional processing, and complex semantic variations inherent in cybersecurity narratives [28] [29] [30]. A major shift was enabled by the introduction of Transformer-based architectures, particularly BERT (Bidirectional Encoder Representations from Transformers), as they capture both preceding and following information, resulting in richer semantic representations for cybersecurity text [9] [18]. BERT and its variants soon became the dominant backbone for modern cybersecurity NLP tasks, as they directly addressed the contextual limitations of earlier models when handling diverse attack descriptions [13].

Despite their success in open-domain NLP, generic BERT models were ineffective for cybersecurity literature due to significant domain mismatch, as they were pretrained on general corpora that lack the specialized vocabulary and structure of security narratives [4] [31] [32]. Key cybersecurity terms such as such as malware families, exploit identifiers, protocol artifacts, and threat actor aliases, were often handled as out of vocabulary terms or poorly tokenised [4] [8]. At the same time, common words like virus, worm, and payload were misinterpreted due to domain-specific semantics, leading to incorrect contextual representations in threat intelligence text [8] [13]. These issus were further amplified by the voluminous, noisy, and weakly structured nature of cybersecurity documents, including CTI reports, vulnerability descriptions, and system logs, as 3 well as continuous semantic drift driven by evolving attack strategies, which static general domain models failed to detect [18] [24].

To overcome these constraints, Domain-Adaptive Pretraining (DAPT) was adopted, in which pretrained language models are further trained on extensive unlabeled cybersecurity corpora [4]. By aligning vocabulary, semantics, and contextual understanding with real-world security data, DAPT consistently improves downstream tasks such as threat classification, named entity recognition, and vulnerability analysis [29] [33]

Table 1: Learning Paradigms in Domain-Adaptive Pretraining (DAPT)

Stage	**Data Type**	**Learning Objective**
I.DAPT	Unlabeled (Raw Text)	Self-supervised learning (MLM) for vocabulary and Semantic alignment[32].
II. Downstream Task	Labeled (Annotated)	Supervised learning for task-specific objectives (e.g., NER, classification)[29,33].

However, DAPT resulted in a scattered ecosystem of cybersecurity specific BERT variant, as shown in Table 2 and Figure 2, each tailored for specific subdomains like CTI, malware analysis, system logs, or vulnerability assessment, thereby limiting cross-domain generalization and comparability across cybersecurity NLP tasks [8] [13].

Table 2: Cybersecurity-Specific BERT Models and Evaluation Scope

Model	**Functioning Domain**	**Task**
CySecBERT[13]	General Cybersecurity Text	Word similarity, clustering, named entity recognition(NER),and relevance classification.
SecureBERT[8]	Cyber Threat Intelligence (CTI)	Masked language modeling(verbs/nouns), Sentiment analysis, and NER.
CTI-BERT[4]	CyberThreatIntelligence (CTI)	ATT&CKtechniqueclassification,malware detection,attributeclassification,andcoarse-andfine-grainedNER.
MalBERT[34]	Android Malware Detection	Binarymalwaredetectionandmulti-label malwarecategoryclassification.
SecBERT[35]	Cybersecurity(General)	EvaluationagainstAPTreportsandgeneral threatintelligence.
DomURLsBERT [14]	Malicious Domains and URLs	Binary and multi-class classification of phishing,malware,DGA,andDNStunneling.
CVSS-BERT[33]	VulnerabilityAssessment	Prediction of CVSS base metrics and vulnerabilityseverityscores.
SecLMNER[36]	Multi-sourceOSINTand ToolOutput	Namedentityrecognitionfornetwork-specific cybersecurityassets.
CyBERT[32]	IndustrialControlSystems (ICS)	Identification and classification of cybersecurityfeatureclaimsintechnical documents.

This fragmentation is intensified by inconsistent experimental pipelines, including customized preprocessing, tokenization, loss functions, and domain-specific evaluation metrics [9] [37] [14],

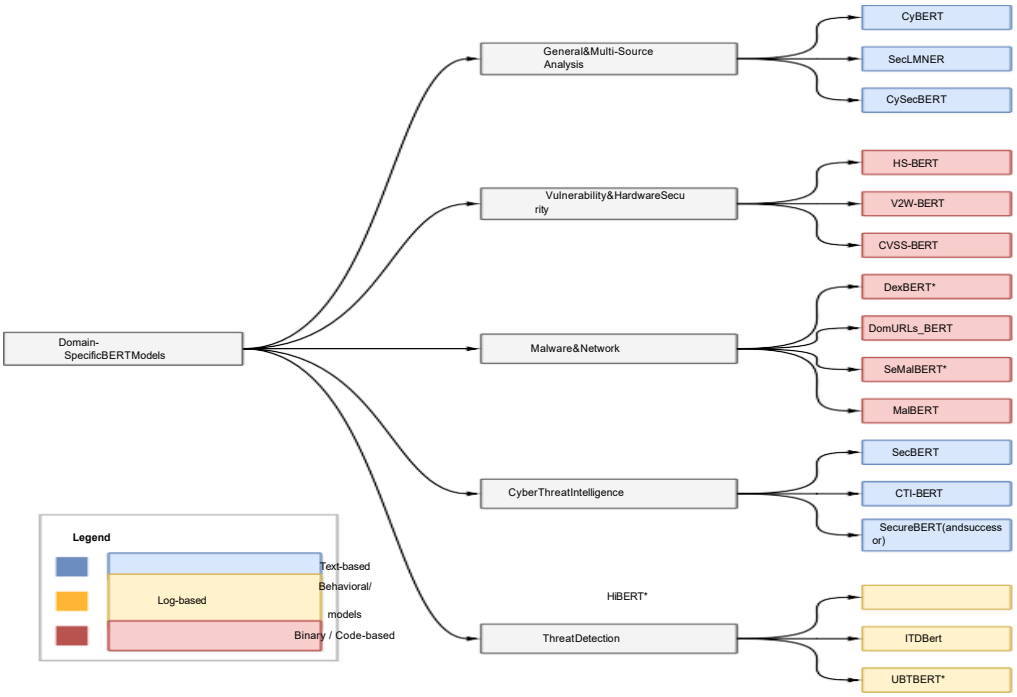


Figure 2: Taxonomy of Cybersecurity BERT Architectures

as discussed in Table 3. As a result, observed performance gains are task-specific and scientifically incomparable across studies and cybersecurity domains [9] [4]. Most cybersecurity-adapted BERT models are assessed primarily within their designated domains, leaving cross-domain transferability largely unexplored [34] [30]. Furthermore, the research lacks a systematic model-domain alignment study under consistent conditions, which limits practical assistance for real-world cybersecurity implementations [1] [37]. In contrast, this study emphasizes cross-domain benchmarking within a standardized experimental framework, explicitly examining model–domain alignment and failure modes through the application of stress-filtered inputs to more accurately simulate operational cybersecurity environments.

Table 3: Identified Research Gaps in Cybersecurity NLP

Gap	**Description**	**Consequence**
Domain Fragmentation	Modelsareevaluatedinisolationonnarrowly defineddomainssuchasmalware,CTI,logs, orvulnerabilities	Results cannot be generalized or compared acrossdomains[4]
Pipeline Inconsistency	Custompreprocessing, tokenization, loss functions,andevaluationmetricsareused acrossstudies	Reported performance is scientifically incomparable acrossmodelsandstudies[8]
Single-Domain Bias	Modelsaretestedonlywithintheirintended domainswithoutcross-domainevaluation	Cross-domaintransferability remainsunexamined[34]
Missing Model–Domain Alignment	Nosystematicanalysislinkingspecialized models to optimal cybersecurity domains	Practitionerslackempirical guidanceformodelselection [13]

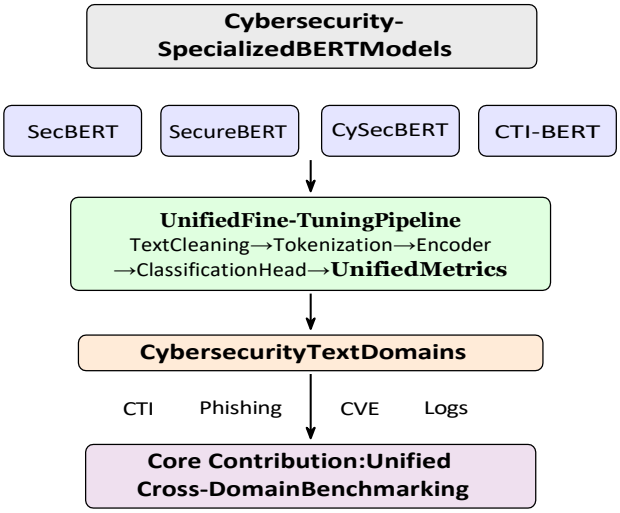


Figure3:Overviewoftheunifiedbenchmarkingstudy.

3. METHODOLOGY

This section describes the experimental methodology as a unified and controlled process designed to enable a fair benchmarking across all evaluated models. Figure 3 provides the graphical representation of the study. The central idea guiding this methodology is that any observed performance differences should originate only from differences in model architecture and pretraining strategy, and not from variations in the experimental pipeline.

In order to accomplish this, all models were assessed under a single and uniform experimental framework. Every pipeline step was kept identical across models and domains. To maintain the original structure and semantics of the data, input texts were subjected to uniform, minimal preprocessing. Only lowercasing, whitespace normalization, and the removal of non-textual artifacts such as malformed tokens were included. With the exception of the stress filtering technique, no language normalization, stemming, lemmatization, or domain-specific cleaning was performed. This ensured that models were exposed to the same raw information content, allowing to evaluate their learned representations without external bias.

Each model employed its own native tokenizer, loaded from the corresponding pre-trained HuggingFace checkpoint. The evaluated models include CTI-BERT [4], SecureBERT [8], CySecBERT [13], and SecBERT [35], which are all cited from their original pretraining publications and related sources. These models incorporate several cybersecurity pretraining approaches from general cybersecurity adaptation, continuous domain-adaptive pretraining, and pretraining from scratch on domain-specific corpora. To maintain consistency with the pretraining assumptions, tokenizer behavior, including special token handling, was left unchanged. All inputs were standardized to a maximum sequence length of 256 tokens using truncation or padding.

Every domain used the same HuggingFace Trainer configuration for evaluation and fine-tuning. The batch size, number of epochs, optimizer, learning rate, and random seed were all kept constant. There was no introduction of model-specific hyperparameter adjustment or early stopping. Any stability-related tweaks were made solely to ensure effective execution and did not alter the optimization goals. Evaluating these architectures under identical downstream conditions enables controlled analysis of how the pretraining technique influences cross-domain behavior, regardless of pipeline or optimization effects

3.1 Domains, Datasets, and Strategy

In this study, each domain is treated as a distinct text classification job and assessed using a single experimental pipeline. The goal is to investigate how various cybersecurity-adapted language models behave under similar and controlled settings rather than to maximize task-specific performance.

The Cyber Threat Intelligence (CTI) domain is made up of curated threat intelligence text obtained from structured reports with ATT&CK style annotations [4]. Multi-class classification of CTI entities, such as ACTOR, MALWARE, TOOL, and TACTIC, is part of the process. After stress filtering, the dataset's initial 15,000 samples are reduced to 2,306 samples. Although there is a moderate decline in TACTIC instances as a result of more semantic constraints, the overall class distribution is still largely stable. The corpus incorporates information from APTNotes threat reports, MITRE ATT&CK knowledge bases, and auxiliary vulnerability-related descriptions.

The phishing domain is based on the Phish No More dataset [38], a consolidated corpus that combines numerous publicly available email datasets. The aim is to distinguish between phishing and legitimate emails using binary classification. Stress filtering generates a compact challenge subset of 2,299 emails from 39,154 original samples, which is then stratified into training and test sets.

System log analysis detects binary anomalies using the HDFS log dataset [39]. Log messages are excessively repetitive and template-driven, with a significant class imbalance favoring normal events. Stress filtering decreases the dataset from 30,000 to 28,160 samples by retaining logs with overlapping anomaly indications while preserving the original task formulation.

In contrast, stress filtering is intentionally not applied in the CVE severity categorization task. CVE severity labels (LOW, MEDIUM, HIGH, and CRITICAL) are based on structured CVSS assessments rather than linguistic clues. Lexical constraints were found to affect label distributions and task validity. As a result, the CVE domain is retained without stress filtering and serves as a diagnostic case for analyzing model failure modes under a text-only formulation.

3.2 Design justification & Threats to Validity

Stress filtering is used specifically for linguistically grounded domains to remove lexically evident samples and create challenging evaluation subsets. A domain-specific lexical indicator set is defined for each applicable domain, and a sample is considered valid if it contains at least two different indicators. Formally, a sample is stress valid if

$$\sum_{i=1}^{n} I\{term_i \in text\} \geq 2 \quad (1)$$

This threshold prevents shortcut learning, enforces contextual interpretation, and simulates analyst level ambiguity. While stress filtering makes diagnosis more difficult, it also reduces dataset size and may change class distributions as shown in Table 4. To reduce this effect, stratified sampling and consistent filtering thresholds are applied to all models in each domain. Furthermore, this study does not claim optimal task performance, but instead prioritizes controlled comparison over score maximization, which aligns with benchmarking purposes rather than state-of-the-art competition.

3.3 Evaluation Metrics

The evaluation technique was developed to ensure that the provided results are meaningful and comparable to class-level performance across all domains and models. Because each domain has

Table 4: Dataset Statistics Before and After Stress Filtering

Domain	Samples(Original)	Samples(Filtered)	Classes
CTI	15,000	2,306	4
Phishing	39,154	2,299	2
Logs	30,000	28,160	2

varied degrees of class imbalance, a unified set of metrics was used consistently to support fair benchmarking.

The key evaluation criteria is the Macro-F1 score, which gives equal weight to all classes regardless of frequency. This option ensures that model performance is not driven by majority classes and provides for a fair assessment of predictive quality across labels. To support this analysis, per-class F1 scores have been included, revealing label-specific strengths and failure patterns that may be disguised by aggregate metrics. Overall accuracy is used as an alternative statistic for completeness, although it is not prioritized due to its susceptibility to class imbalance

$$\text{Macro-F1} = \frac{1}{C}\sum_{c=1}^{C} F1_c \qquad (2)$$

- where C denotes the number of classes.

3.4 Zero-Shot Cross-Domain Training Methodology

While the unified fine-tuning approach ensures controlled comparability across domains, real-world cybersecurity systems often operate in domain mismatch scenarios, in which labeled data from a target domain is absent or highly limited. In such cases, the practical utility of a cybersecurity language model is determined not by its peak in domain accuracy, but by its capacity to transfer previously acquired domain knowledge to unknown security contexts.

To explicitly examine this, we add a zero-shot cross-domain training protocol to the unified benchmarking study, which allows models to be trained on a single source domain and then evaluated on a different target domain without any additional fine-tuning or adaptation. This setup isolates the underlying generalization capacity of each cybersecurity-specific BERT model and directly tackles the gap in cross-domain transfer analysis identified in previous work and in the motivation of this study.

3.4.1. Experimental Design

The zero-shot evaluation employs a train-once, test-elsewhere approach. For every experiment:

- A model is fine-tuned exclusively on a single source domain.
- The trained model is then tested unchanged on a new target domain.
- There are no target-domain samples, labels, or statistics used during training.

This technique assures that any apparent performance on the target domain is the result of semantic and representational overlap acquired during source-domain training, rather than task-

specific memorization

Table 5 depicts the domains analyzed which include Cyber Threat Intelligence (CTI), phishing emails, system logs, and CVE severity descriptions. Each domain keeps its original task formulation, label space, and assessment metrics, as described in the unified pipeline.

Table 5: Zero-Shot Cross-Domain Training–Evaluation Matrix

Source Domain(Train)	**Target Domains(Test)**
CTI	Phishing, Logs, CVE
Phishing	CTI, Logs, CVE
Logs	CTI, Phishing, CVE
CVE	CTI, Phishing, Logs

3.4.2. Data Preparation and Training Protocol

All zero-shot trials employ the stress-filtered datasets from the core benchmarking pipeline, ensuring consistency with the main study and preventing dataset leakage. Stress filtering remains enabled for linguistically grounded domains (CTI, phishing, and logs) but disabled for CVE severity rating, in accordance with the methodological reason presented previously. This ensures that zero-shot findings are directly comparable to the fine-tuned results reported in the unified evaluation.

We use the same training configuration in zero-shot experiments as the unified benchmarking pipeline. No domain-specific hyperparameter adjustment, class reweighting, or early stopping are introduced. Each model is trained individually on a single source domain and tested across all remaining target domains, resulting in a comprehensive cross-domain transfer matrix that includes both strong and weak transfer pathways.

3.5 Few-Shot Cross-Domain Adaptation Methodology

The zero-shot experiments established an important baseline: cybersecurity language models exhibit high domain dependence and asymmetric transfer behavior, even when evaluated using a fully unified pipeline. However, zero-shot configurations do not accurately reflect real-world cybersecurity deployments, in which analysts often have access to a small number of labeled samples from a new domain rather than none at all.

In practice, labeled cybersecurity data is expensive, noisy, and difficult to gather in large quantities. The operational concern now moves from whether a model can generalize without adaptation to how effectively it can adapt when minimal target domain supervision is provided. To investigate this, we extend the zero-shot procedure to include a few-shot cross-domain adaption setting, in which models are trained on a source domain before being exposed to limited labeled data from a target domain.

This investigation focuses on two transfer paths: CTI → phishing and CTI → logs, which show contrasting semantic behavior in the zero-shot condition. This enables us to investigate how domain structure affects adaptation efficiency under constrained supervision.

3.5.1. Experimental Design

In the few-shot approach, each model is fully fine-tuned on the CTI domain before being fine-

tuned on small portions of the target domain. Target-domain supervision is gradually introduced at 10%, 25%, and 50% of the available training data, with evaluation always performed on a predetermined held-out test set.

No architecture changes, hyperparameter tuning, or domain-specific optimizations are implemented

As a result, any observed performance differences can be directly linked to the amount of target-domain supervision available, rather than procedural variance. This design preserves the unified pipeline's experimental controls while allowing for clean comparisons across models and supervision levels.

3.5.2. Data Preparation and Training Protocol

All few-shot experiments reuse the previously defined stress-filtered datasets, assuring compatibility with zero-shot and completely fine-tuned evaluations. The whole stress-filtered CTI dataset is used for source domain supervision. For the target domains, few-shot subsets are generated solely from the training split, whereas the test split remains constant across experiments. To preserve class imbalance characteristics, few-shot samples are stratified when used for phishing. The natural class imbalance in logs is purposely maintained to represent operational contexts. This approach assures that the few-shot results are directly comparable to previous findings.

3.6 Controlled Training Strategy Ablation Study

The unified benchmarking approach demonstrated that cybersecurity-adapted BERT models converge considerably in performance when tested under identical settings. However, this convergence does not explain whether performance similarities are caused by model architecture, domain characteristics, or the downstream training technique itself.

To detangle these aspects, we conduct a controlled ablation research that isolates the influence of the training approach while leaving all other components fixed. Unlike previous research, which frequently conflates architectural uniqueness with training processes, our analysis focuses specifically on how different training configurations affect performance inside the same unified pipeline.

The purpose of this ablation is diagnostic rather than competitive: to discover whether more aggressive or relaxed training procedures reliably extract additional domain signal, or whether simpler and more stable configurations already capture the majority of what can be learned from the data.

3.6.1. Experimental Design and Training Protocol

Three controlled training variants were tested: A, B, and C in all four cybersecurity domains. These variants are not new models, but rather different training settings used evenly within the same experimental context. Only the training technique is changed, therefore any observed differences can be attributed purely to training behavior rather than experimental artifacts.

Variant A represents the baseline configuration utilized in the primary benchmarking trials. It is purposely conservative, focusing on stability and reproducibility as a point of comparison.

Variant B introduces a small relaxation in training dynamics while maintaining the same target

and data exposure. This option determines whether modestly stronger optimization consistently yields in performance gains if domain semantics are already captured.

Variant C significantly lowers training limitations, resulting in a more permissive learning environment. This variation investigates whether lowered constraint results in richer signal extraction or instead increases instability, particularly in domains with limited semantic grounding.

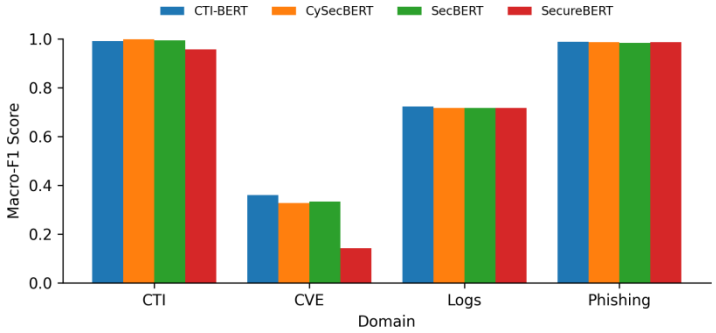


Figure 4: Macro-F1 scores of all evaluated models across cybersecurity domains.

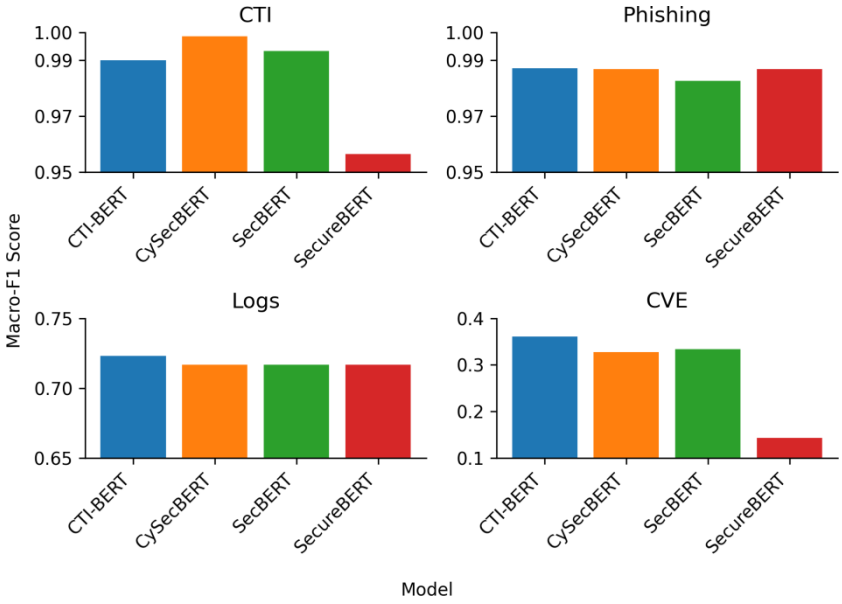


Figure 5: Per-domain Macro-F1 performance of cybersecurity-specialized BERT models.

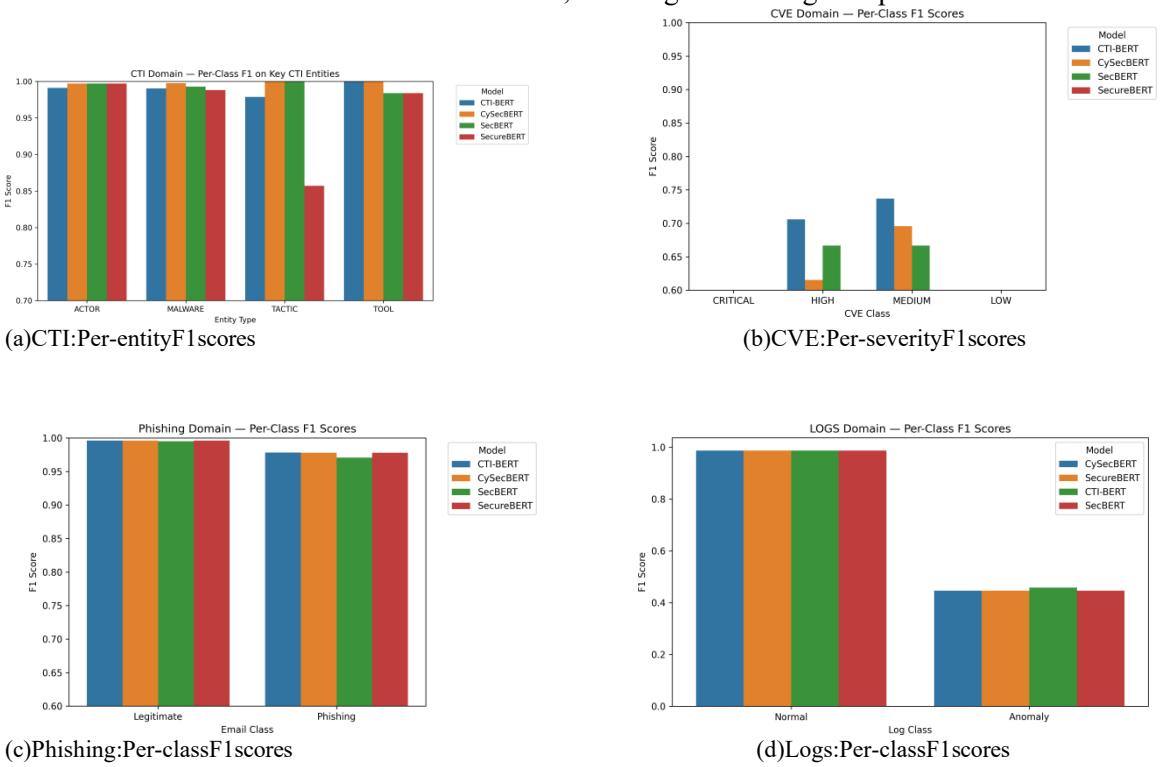
4 RESULTS

This section reports on the experimental results acquired utilizing the previously established unified and stress-filtered evaluation technique. The results are organized by domain and analyzed using a common set of evaluation criteria. The purpose is to understand model behavior, convergence patterns, and domain-specific limits under controlled conditions.

Figure 4 and 5 shows a cross-domain comparison of macro-F1 scores. Macro-F1 is the major statistic used to ensure that all classes contribute equally, especially when there is a class

imbalance. These figures provide a quick visual summary of how models perform across different cybersecurity domains when tested under identical settings. Despite changes in architecture and pretraining technique, the results in the following table 6 indicate little differentiation between models across domains, implying that cybersecurity-specific pretraining does not always translate into higher performance under stress-filtered inputs.

These figures offer a visual summary of how models behave across heterogeneous cybersecurity domains when tested under identical conditions, focusing on convergence patterns and shared



(a)CTI:Per-entityF1scores (b)CVE:Per-severityF1scores

(c)Phishing:Per-classF1scores (d)Logs:Per-classF1scores

Figure 6: Per-class and per-entity F1 score analysis across cybersecurity domains under stress-filtered evaluation.

performance ceilings rather than absolute score differences. Table 6 shows considerable performance convergence across models in phishing and logs, moderate differentiation in CTI, and uniformly low performance in CVE severity classification, confirming that domain features influence model behavior when stress-filtered.

Table 6: Cross-Domain Macro-F1 Scores under Unified Stress-Filtered Evaluation

Domain	**CTI-BERT**	**CySecBERT**	**SecBERT**	**SecureBERT**
CTI	0.9901	0.9987	0.9934	0.9565
Phishing	0.9871	0.9870	0.9827	0.9870
Logs	0.7230	0.7168	0.7168	0.7168
CVE	0.3607	0.3278	0.3333	0.1429

Figure 6 provides per-class F1 scores for selected domains to aid in model behavior analysis. In the CTI domain, ACTOR and TOOL entities are relatively simple to classify across all models, showing that surface-level entity cues persist even after stress filtering. However, the TACTIC classification exhibits significant divergence. CTI-BERT retains stable performance under stress,

whereas SecureBERT degrades significantly, suggesting that continuous pretraining may be less beneficial for abstract tactical semantics that need higher-level contextual reasoning. This behavior demonstrates how stress filtering can uncover representational differences that are not apparent in aggregate metrics.

In the CVE domain, all models show a consistent structural pattern. The MEDIUM and HIGH severity classes get moderate F1 values, whilst the LOW and CRITICAL severities have near-zero F1. This result is consistent across CTI-BERT, SecureBERT, CySecBERT, and SecBERT, implying that text-only severity prediction suffers from fundamental limitations rather than model-specific problems. These findings are consistent with prior observations that vulnerability severity is weakly grounded in natural language descriptions and mainly reliant on structured CVSS metadata rather than free text.

For phishing email detection, per-class F1 scores show near-saturated performance across all models. Legitimate emails regularly get F1 scores more than 0.99, but phishing emails fall in a tight range of 0.97 to 0.98. The low variation across architectures suggests that, under stress-filtered conditions, phishing detection is largely insensitive to differences in cybersecurity-oriented pretraining and is instead dominated by the dataset's strong lexical and stylistic cues [40].

In the logs domain, all models acquire very high F1 scores for the Normal class, indicating robust detection of repetitive and templated log patterns. In contrast, anomalous logs consistently yield low F1 scores, which range between 0.44 and 0.46. This pattern is consistent across architectures, implying a ceiling effect for text-only anomaly detection caused by low semantic diversity and structural repetition in log data. These results align with prior studies that effective log anomaly detection often involves sequence modeling, temporal context, or structured log parsing in addition to plain text representation [25]

Table 7: Cross-domain rank matrix of evaluated models based on Macro-F1.

Domain	CTI-BERT	CySecBERT	SecBERT	SecureBERT
CTI	3	1	2	4
CVE	1	3	2	4
Logs	1	2	2	2
Phishing	1	2	4	2

4.1 Model Domain Alignment

The unified evaluation indicates consistent model domain alignment trends across all assessed tasks. CTI-BERT has the most consistent performance in the CTI domain, especially under stress-filtered settings requiring complex tactical reasoning and contextual threat assessment. This shows that when shallow lexical shortcuts are removed, pretraining from scratch on high-quality CTI corpora delivers a measurable advantage.

In contrast, phishing detection demonstrates performance saturation across all investigated models, implying that cybersecurity-specific pretraining provides little further benefit for this task once strong lexical and stylistic signals are available. Under stress filtering, all models achieve near-identical performance, implying that phishing detection is largely resistant to architectural variation.

In the logs domain, all models achieve great performance on the majority of classes but fail to

effectively model anomalous behavior. This recurrent failure pattern demonstrates a structural weakness of text only approaches when applied to system logs, where anomaly identification is more strongly reliant on temporal dynamics and event sequences than on isolated content.

For CVE severity classification, uniform failure across models particularly for the LOW and CRITICAL classes indicates that vulnerability severity is poorly anchored in natural language and cannot be reliably inferred without structured metadata. The lack of meaningful separation

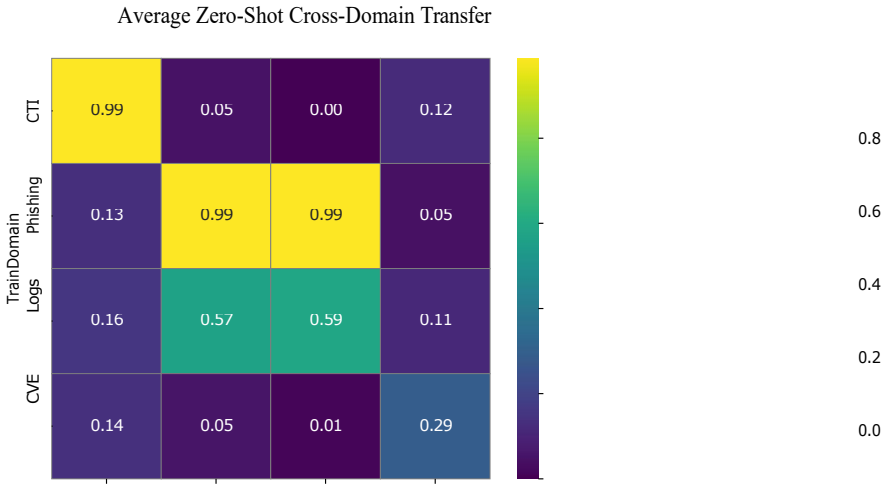


Figure 7: Average Zero-Shot Cross-Domain Transfer Heatmap

across models demonstrates that this constraint is domain-driven rather than architectural.

Collectively, these results show that cross-domain robustness is not always correlated with model specialization. Therefore, rather than assuming that broader pretraining alone ensures superior performance, effective deployment of cybersecurity language models needs careful alignment between model design and domain features. Tables 7 and 8 show that no single model performs best across all domains, highlighting the need to choose models based on domain-specific requirements rather than assuming specialized pretraining will always generalize well.

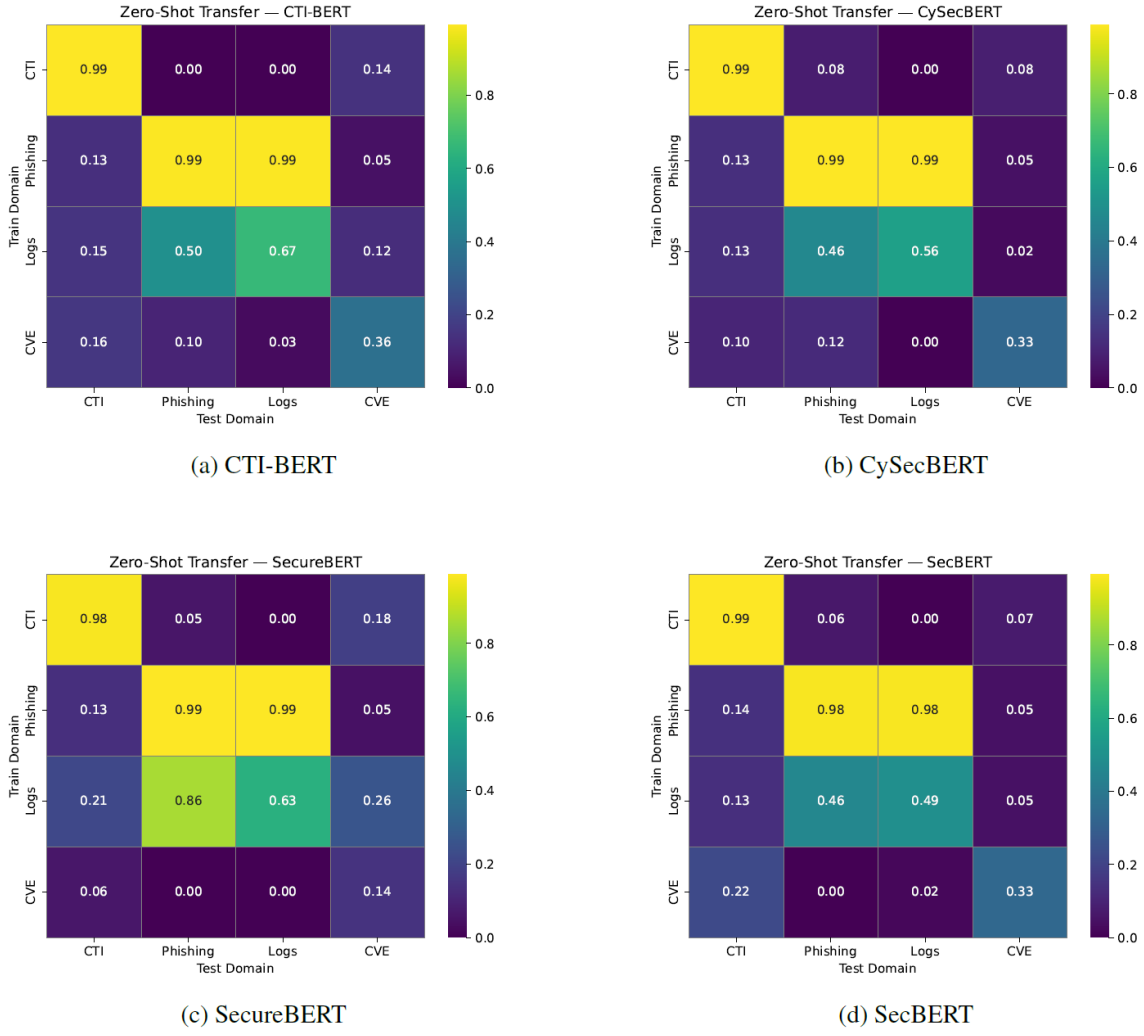
Table 8: Recommended Model Usage Based on Empirical Findings

Domain	**Recommended Model**	**Reason**
CTI	CTI-BERT	Stableunderstress; betterabstracttactical reasoning
Phishing	Any	Tasksaturated;minimaldifferentiationacross models
Logs	None(text-only)	Structuralrepetitionlimitslearningofanomalies
CVE	None(text-only)	Severityweaklygroundedinnaturallanguage

4.2 Zero-Shot Results and Cross-Domain Analysis

Figure 7 depicts the average zero-shot cross-domain transfer performance across all evaluated models. A distinct structural pattern emerges: zero-shot generalization is both domain-dependent and very asymmetric. Models function nearly perfectly when trained and evaluated on the same domain, but cross-domain transfer varies significantly depending on the source-target pair.

Training on CTI results in high self-transfer, but collapses almost entirely when tested on phishing and logs, with only limited transfer to CVE. In contrast, phishing-trained models transfer reliably to logs but have consistently weak transfer to CTI and CVE. Log-trained models show uniformly low transfer in all other domains. CVE-trained models show selective transfer to CTI, but fail to generalize to phishing and logs. No domain serves as a unified source of transferable representations, and no model demonstrates consistent cross-domain robustness. These findings support the idea that domain semantics, rather than model architecture, determine zero-shot behavior



(a) CTI-BERT (b) CySecBERT

(c) SecureBERT (d) SecBERT

Figure 8: Zero-shot cross-domain transfer heatmaps for all evaluated models under stress-filtered evaluation.

Domain-wise zero-shot transfer results from figure 8, show clear and consistent patterns that are driven by domain semantics. Models trained on CTI capture high-level threat ideas that partially align with vulnerability descriptions, allowing limited transfer to CVE. However, they fail to generalize to phishing and logs, which rely on intent-driven and behavioral signals. Phishing trained models transfer effectively to logs, indicating shared surface level and template driven patterns, but perform nearly random on CTI and CVE. Log-trained models had the weakest overall transfer rate, showing that representations formed from repetitive and structurally limited log data do not generalize beyond the source domain. CVE-trained models show minimal transfer to CTI, consistent with shared terminology around vulnerabilities and impact, but remain firmly

constrained to severity specific language. These patterns are consistent across models and are preserved in the averaged transfer matrix, demonstrating that domain structure governs transfer behavior rather than individual model attributes.

At the model level, CTI-BERT exhibits the most asymmetric transfer, performing best when CTI is the source or target domain but failing on phishing and logs, demonstrating that specialization improves depth but restricts breadth. CySecBERT and SecureBERT behave moderately and consistently, with limited transfer where semantic overlap exists. SecBERT exhibits the most variable transfer patterns, showing poor semantic congruence across domains. Across all models, stress filtering reveals significant transfer collapse, confirming that apparent zero-shot generalization frequently relies on shallow lexical overlap rather than transferable semantic understanding.

4.3 Few-Shot Results and Cross-Domain Adaptation Analysis

The few-shot experiments examine how well models adapt to a new domain when only limited target domain supervision is provided. Across both transfer situations, as shown in figure 9, a consistent pattern emerges: adaptation behavior is influenced more by domain characteristics than model architecture.

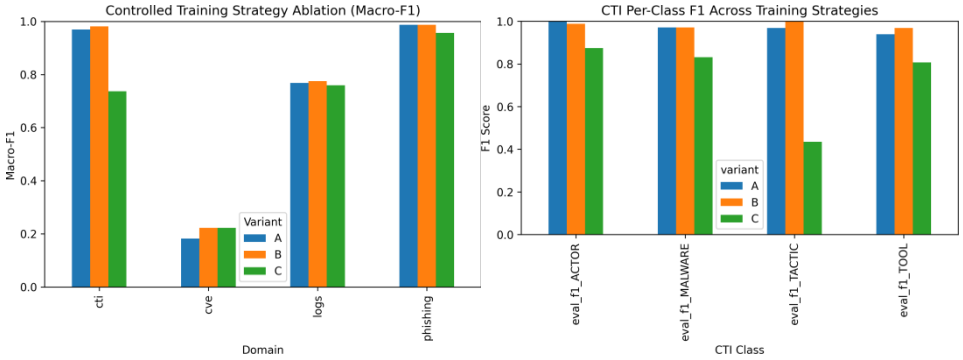
In the CTI → phishing scenario, all models exhibit performance saturation. Even with only 10% target-domain data, Macro-F1 scores are already quite high, and further supervision results in only minor improvements. CTI-BERT, SecureBERT, and CySecBERT converge quickly and remain closely related throughout all supervision levels. SecBERT starts low but gradually improves as additional data is introduced. This trend suggests that phishing detection is extremely data-efficient, with minimal exposure required for models to align with dominant lexical and stylistic cues.

In comparison, the CTI → logs setting provides less adaption advantage. For all models, Macro-F1 remains nearly constant between 10% and 50% target-domain data. While SecBERT exhibits slightly higher stability, no model shows significant improvement with more supervision. This flat response shows log data's semantic resistance, since structural repetition and template-driven patterns limit the use of incremental fine-tuning.

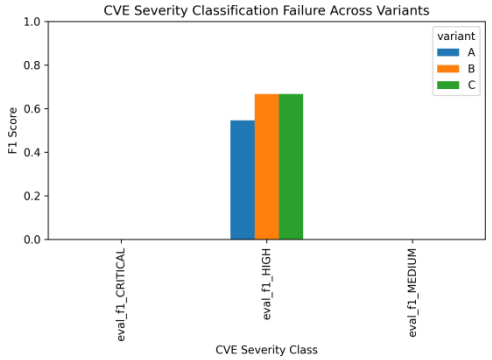
Overall, the few-shot results support this study's primary claim that few-shot success is determined by domain structure rather than model sophistication. Some domains adjust quickly with minimal data, while others stay resistive even with increased supervision.

4.4 Controlled Training Strategy Ablation

The ablation study assesses how different training strategies affect performance within a predefined experimental pipeline. Across domains, a consistent trend emerges, the influence of training technique is heavily dependent on domain semantics rather than uniformly improving results



(a) Cross-domain Macro-F1acros straining variants (b) CTI per-class F1 sensitivity across strategies



(b) CVE severity classification failure a cross variants

Figure 10: Controlled training strategy ablation results under the unified benchmarking pipeline.

At the macro level, phishing achieves near-ceiling Macro-F1 across all versions, demonstrating low sensitivity to training strategy. Logs are also consistent across variants, demonstrating the repetitive and template-driven nature of log data. In contrast, the CTI domain demonstrates clear sensitivity: Variants A and B maintain high Macro-F1, whereas Variant C shows a significant decline in performance. This suggests that relaxed training constraints have a negative impact on semantically rich tasks requiring stable contextual learning. CVE severity classification yields uniformly low Macro-F1 across all variants, demonstrating that changes in training technique do not overcome weak textual grounding.

This behavior is further clarified by CTI's class-level analysis. The ACTOR, MALWARE, and TOOL classes stay stable under Variants A and B, but Variant C constantly degrades. The TACTIC class has the most fluctuation, reaching its peak F1 under Variant B and collapsing rapidly under Variant C, emphasizing its reliance on controlled optimization.

Finally, the variability analysis reveals that Variant A has the lowest performance spread, Variant B has slightly higher dispersion, and Variant C has the most instability. Overall, the results indicate that training stability is more important than aggressive optimization, particularly in semantically complex domains

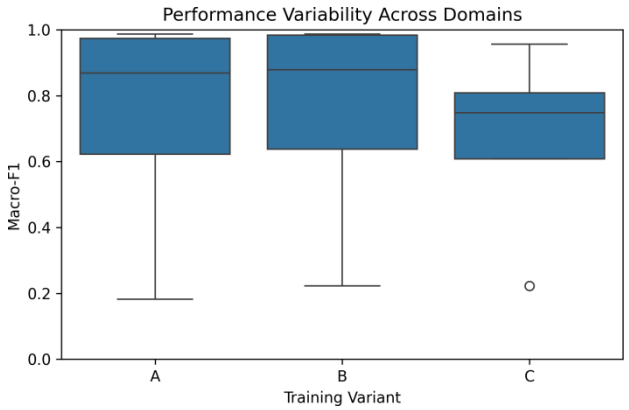


Figure 11: Performance variability across domains for each training strategy variant.

5 CONCLUSION & FUTURE WORK

This paper presented a unified, cross-domain benchmarking approach for assessing cybersecurity specialized BERT models under identical experimental conditions. The analysis shows that performance convergence and shared failure modes dominate across models, regardless of pretraining approach or intended domain focus, by establishing a consistent fine-tuning and evaluation pipeline and incorporating stress-filtered inputs. Across all evaluated areas, model behavior is found to be primarily influenced by data attributes and task formulation rather than architectural variation.

Strong lexical overlap and entity-level cues drive high overall performance in CTI, whereas stress filtering reveals flaws in abstract tactical reasoning. Although classes like ACTOR and TOOL remain reliably identifiable, TACTIC classification shows significant robustness variations, with CTI-BERT remaining stable under higher semantic constraints while SecureBERT degrades. This suggests that domain-specific pretraining increases representation quality but does not ensure robustness once shallow lexical shortcuts are removed.

The CVE domain demonstrates a fundamentally different limitation. Vulnerability severity is not a completely linguistic construct, it is generated from structured CVSS components that are only weakly grounded on natural language. As a result, all assessed models show the same failure pattern, with partial learnability for MEDIUM and HIGH severity levels and near collapse for LOW and CRITICAL classes. This consistent behavior across models suggests a task level limitation rather than a flaw in the pretraining strategy.

In the phishing domain, all models exhibit near saturated performance. Even with stress filtering, phishing and legitimate emails retain significant and repetitive linguistic cues, resulting in little difference between Macro-F1 and per-class F1 scores. This illustrates that text only phishing detection is already well captured by existing transformer representations and is largely unaffected by additional cybersecurity specialization.

The logs domain supports performance convergence. With very repetitious templates and low semantic diversity, all models rely on shortcut patterns, attaining near-perfect F1 on the Normal class while continually failing to represent anomalous behavior. This ceiling effect demonstrates the structural constraints of text based techniques to log anomaly detection.

These findings evaluated using Macro-F1 and per-class F1 scores show that cybersecurity language models converge under controlled, stress-filtered evaluation. The fundamental

contribution of this work is not to evaluate models, but to identify where current models succeed, where they fail, and why.

This study improves our understanding of the practical limitations of cybersecurity specific BERT models by isolating model-domain alignment effects. This study is limited to text-only inputs and a certain range of cybersecurity domains. While stress filtering is diagnostically useful for identifying brittle decision boundaries and suppressing superficial lexical cues, it decreases dataset size and does not capture all types of real world ambiguity found in operational environments. Furthermore, the analysis is limited to fine tuned encoder based models and does not investigate generative, multimodal, or sequence aware architectures. Finally, the evaluation is carried out under static fine tuning conditions, which do not account for temporal drift or evolving threat narratives.

5.1. Future Work

Future research should build on this study's diagnostic findings to solve the structural limitations found by unified, stress-filtered evaluation.

- Adding structured metadata (e.g., CVSS attributes, log templates), temporal signals, and auxiliary artifacts to more accurately portray operational cybersecurity settings.
- A systematic investigation of domain-adaptive and continuous pretraining procedures under controlled benchmarking settings.
- Investigating into vocabulary fragmentation and semantic drift to better understand grounding effects across domains.
- Human-in-the-loop validation, it involves incorporating analyst feedback to qualitatively evaluate model flaws and real world relevance.

CONFLICT OF INTEREST

The authors declare that there are no conflicts of interest regarding the publication of this study.

DECLARATION

The authors assert that all datasets utilized in this work were sourced from publically accessible and widely used cybersecurity text corpora, encompassing materials for cyber threat intelligence reports, phishing emails, system logs, and vulnerability descriptions. At no point during the research process was any private, proprietary, or personally identifiable information acquired or used. The authors generated all of the experiments, analyses, figures, and results given in this publication utilizing a unified experimental pipeline in controlled and reproducible settings. The study has not been submitted to or published in any other journal or conference.

ACKNOWLEDGMENT

The authors thank the Department of Computer Science, Aligarh Muslim University, for academic support.

REFERENCES

[1] M. Macas, C. Wu, and W. Fuertes, "A survey on deep learning for cybersecurity: Progress, challenges, and opportunities," Computer Networks, vol. 212, p. 109032, 5 2022.
[2] S. Yanamandram Kuppuraju, R. Dubey, and M. Kumar, "Threat intelligence automation using nlp and

machine learning," INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS, vol. 13, pp. 2320–2882, 04 2025.
[3] R. Marinho and R. Holanda, "Automated emerging cyber threat identification and profiling based on natural language processing," IEEE Access, vol. 11, pp. 58915–58936, 1 2023.
[4] Y. Park and W. You, "A pretrained language model for cyber threat intelligence," in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track (M. Wang and I. Zitouni, eds.), (Singapore), pp. 113–122, Association for Computational Linguistics, Dec. 2023. 19
[5] M. Buchel, T. Paladini, S. Longari, M. Carminati, S. Zanero, H. Binyamini, G. Engelberg, D. Klein, ¨ G. Guizzardi, M. Caselli, A. Continella, M. van Steen, A. Peter, and T. van Ede, "SoK: Automated TTP Extraction from CTI Reports – Are We There Yet?," 8 2025.
[6] S. Zhang, C. Hu, L. Wang, M. Mihaljevic, S. Xu, and T. Lan, "A malware detection approach based on deep learning and memory forensics," Symmetry, vol. 15, p. 758, 3 2023.
[7] W. Meng, Y. Liu, Y. Zhu, S. Zhang, D. Pei, Y. Liu, Y. Chen, R. Zhang, S. Tao, P. Sun, and R. Zhou, "LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs," pp. 4739–4745, 7 2019.
[8] E. Aghaei, X. Niu, W. Shadid, and E. Al-Shaer, SecureBERT: a Domain-Specific language model for cybersecurity. 1 2023.
[9] H. Tiwari, D. of Computer Science, N. T. U. o. S. Information Engineering, and T. (NTUST), "Advancing Vulnerability Classification with BERT: A Multi-Objective Learning Model," arXiv, 3 2025.
[10] A. Rahali and M. A. Akhloufi, "MalBERTV2: Code aware BERT-Based model for malware identification," Big Data and Cognitive Computing, vol. 7, p. 60, 3 2023.
[11] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, "TTPDrill," pp. 103–115, 12 2017.
[12] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, "Deep learning for anomaly detection in log data: A survey," Machine Learning with Applications, vol. 12, p. 100470, 5 2023.
[13] "CYSECBERT: a Domain-Adapted Language model for the cybersecurity domain," ACM Transactions on Privacy and Security, vol. 27, p. Article 18, 4 2024.
[14] A. El Mahdaouy, S. Lamsiyah, M. Janati Idrissi, H. Alami, Z. Yartaoui, College of Computing, Mohammed VI Polytechnic University, Ben Guerir, Morocco, Department of Computer Science, Faculty of Science, Technology and Medicine, University of Luxembourg, Luxembourg, LISAC Laboratory, Faculty of Sciences Dhar El Mehraz, USMBA, Fez, Morocco, Vanguard Center, Mohammed VI Polytechnic University, Ben Guerir, Morocco, and National Moroccan Computer Emergency and Response Team (maCert), Morocco, "Domurls bert: Pre-trained bert-based model for malicious domains and urls detection and classification," 2024.
[15] S. Gururangan, A. Marasovic, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, ´ "Don't stop pretraining: Adapt language models to domains and tasks," July 2020.
[16] M. Halappanavar, S. S. Das, E. Serra, A. Pothen, and E. Al-Shaer, "Cybersecurity-Tools/V2W-BERT," OSTI OAI (U.S. Department of Energy Office of Scientific and Technical Information), 1 2022.
[17] S. L. Mirtaheri, A. Pugliese, and V. Pascucci, "Automated vulnerability score prediction through lightweight generative AI," Knowledge-Based Systems, vol. 329, p. 114406, 9 2025.
[18] D. Demirol, R. Das, and D. Hanbay, "A Novel Approach for Cyber Threat Analysis Systems Using BERT Model from Cyber Threat Intelligence Data," Symmetry, vol. 17, p. 587, 4 2025.
[19] U. Gupta, S. Kandpal, H. Alamro, M. M. Asiri, M. H. Alanazi, A. M. Al-Sharafi, and S. Sorour, "Efficient malware detection using NLP and deep learning model," Alexandria Engineering Journal, vol. 124, pp. 550–564, 4 2025.
[20] M. Korkmaz, O. K. Sahingoz, and B. Diri, "Detection of phishing websites by using Machine Learning-Based URL analysis," ArXiv, 7 2020.
[21] B. P. Gond, Rajneekant, P. Kishore, and D. P. Mohapatra, "Malware classification leveraging nlp & machine learning for enhanced accuracy," 2025.
[22] T. Peng, I. Harris, and Y. Sawa, "Detecting phishing attacks using natural language processing and machine learning," in 2018 IEEE 12th International Conference on Semantic Computing (ICSC), pp. 300–301, 2018.
[23] A. Alzu'bi, O. Darwish, A. Albashayreh, Y. Tashtoush, J. U. o. S. Department of Computer Science, Technology, I. Security, and E. M. U. Applied Computing, "Cyberattack event logs classification using deep learning with semantic feature analysis," 11 2024.
[24] P. Santos, R. Abreu, M. J. C. S. Reis, C. Serodio, and F. Branco, "A Systematic Review of Cyber ˆ Threat intelligence: The effectiveness of technologies, strategies, and collaborations in combating modern threats," Sensors, vol. 25, p. 4272, 7 2025.

[25] S. Chen and H. Liao, "BERT-Log: Anomaly detection for system logs based on pre-trained language model," Applied Artificial Intelligence, vol. 36, 11 2022.
[26] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog," Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1285–1298, 10 2017.
[27] S.-S. Chen, R.-H. Hwang, C.-Y. Sun, Y.-D. Lin, and T.-W. Pai, "Enhancing cyber threat intelligence with named entity recognition using BERT-CRF," GLOBECOM 2023 - 2023 IEEE Global Communications Conference, pp. 7532–7537, 12 2023.
[28] N. Mohamed, "Artificial intelligence and machine learning in cybersecurity: a deep dive into state-of-the-art techniques and future paradigms," Knowledge and Information Systems, vol. 67, pp. 6969–7055, 4 2025.
[29] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, and G. A. Language, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding."
[30] J. Liu, Y. Zhao, Y. Feng, Y. Hu, and X. Ma, "SeMalBERT: Semantic-based malware detection with bidirectional encoder representations from transformers," Journal of Information Security and Applications, vol. 80, p. 103690, 12 2023.
[31] M. Amine Ferrag, F. Alwahedi, A. Battah, and B. Cherif, "Generative AI in cybersecurity: A 20 comprehensive review of LLM applications and vulnerabilities," 2 2025.
[32] K. Ameri, M. Hempel, H. Sharif, J. Lopez, Jr, and K. Perumalla, "CyBERT: Cybersecurity claim classification by Fine-Tuning the BERT language model," Journal of Cybersecurity and Privacy, vol. 1, pp. 615–637, 11 2021.
[33] M. Shahid and H. Debar, "CVSS-BERT: Explainable Natural Language Processing to Determine the Severity of a Computer Security Vulnerability from its Description," arXiv (Cornell University), 11 2021.
[34] A. Rahali, M. A. Akhloufi, R. Perception, I. M. R. G. (PRIME), and U. d. M. Dept of Computer Science, "MALBERT: USING TRANSFORMERS FOR CYBERSECURITY AND MALICIOUS SOFTWARE DETECTION," 3 2021.
[35] H. Huang and Y. Wang, "SecBERT: Privacy-preserving pre-training based neural network inference system," Neural Networks, vol. 172, p. 106135, 1 2024.
[36] Y. Zhang, J. Liu, X. Zhong, and L. Wu, "SecLMNER: A framework for enhanced named entity recognition in multi-source cybersecurity data using large language models," Expert Systems with Applications, vol. 271, p. 126651, 1 2025.
[37] J. Zhang, H. Bu, H. Wen, Y. Liu, H. Fei, R. Xi, L. Li, Y. Yang, H. Zhu, and D. Meng, "When LLMs meet cybersecurity: a systematic literature review," Cybersecurity, vol. 8, 2 2025.
[38] A. Al-Subaiey, M. Al-Thani, N. A. Alam, K. F. Antora, A. Khandakar, and S. A. U. Zaman, "Novel interpretable and robust web-based ai platform for phishing email detection," 2024.
[39] J. Zhu, S. He, P. He, J. Liu, and M. R. Lyu, "Loghub: A large collection of system log datasets for ai-driven log analytics," 2023.
[40] H. Li, J. Yang, Y. Li, and K. Li, "Email phishing attack detection based on BERT transformer model," International Conference on Optics, Electronics, and Communication Engineering (OECE 2024), p. 114, 11 2024

AUTHORS

Laeeba Javed: Laeeba Javed is currently pursuing an M.Sc. in Cyber Security and Digital Forensics at Aligarh Muslim University. She holds a B.Sc. (Hons.) in Computer Applications from the same institution. Her research interests include cybersecurity, machine learning, and applied computer science.

Prof. (Dr.) Aasim Zafar: Prof. (Dr.) Aasim Zafar is a Professor of Computer Science at Aligarh Muslim University with over 29 years of teaching and research experience. His research covers mobile ad hoc and sensor networks, image and video analytics, information retrieval, e-security, neuro-fuzzy systems, and software engineering. He has led multiple nationally and internationally funded projects and published extensively in reputed journals and conferences, contributing to ICT-enabled education and applied cybersecurity systems