EventHunter: Dynamic Clustering and Ranking of Security Events from Hacker Forum Discussions

Yasir ECH-CHAMMAKHY*†, Anas Motii*, Anass Rabii†, Jaafar Chbili‡
*College of Computing, Mohammed VI Polytechnic University (UM6P), Ben Guerir, Morocco
†Deloitte Morocco Cyber Center, Casablanca, Morocco
‡Deloitte Conseil, Paris, France
Emails: *{Yasir.ECH-CHAMMAKHY, Anas.MOTII}@um6p.ma; †arabii@deloitte.fr; ‡jchbili@deloitte.fr

***Abstract*—Hacker forums provide critical early warning signals for emerging cybersecurity threats, but extracting actionable intelligence from their unstructured and noisy content remains a significant challenge. This paper presents an unsupervised framework that automatically detects, clusters, and prioritizes security events discussed across hacker forum posts. Our approach leverages Transformer-based embeddings fine-tuned with contrastive learning to group related discussions into distinct security event clusters, identifying incidents like zero-day disclosures or malware releases without relying on predefined keywords. The framework incorporates a daily ranking mechanism that prioritizes identified events using quantifiable metrics reflecting timeliness, source credibility, information completeness, and relevance. Experimental evaluation on real-world hacker forum data demonstrates that our method effectively reduces noise and surfaces high-priority threats, enabling security analysts to mount proactive responses. By transforming disparate hacker forum discussions into structured, actionable intelligence, our work addresses fundamental challenges in automated threat detection and analysis.**

***Index Terms*—Cyber Threat Intelligence, Hacker Forums, Event Detection**

I. Introduction

The evolving landscape of cyber threats demands increasingly sophisticated detection and response capabilities [13], [20], [38]. Hacker forums have emerged as pivotal sources of cyber threat intelligence (CTI), facilitating the exchange of illicit tools, tactics, and sensitive information critical to executing advanced cyber attacks. These platforms have become significant marketplaces, generating substantial revenues, such as $300 million annually from credit card fraud and $864 million from DDoS-for-hire schemes [62]. Therefore, continuous and effective monitoring of hacker forums is essential for proactive threat detection.

Researchers have explored various techniques to derive actionable intelligence from these platforms, including identifying influential hackers [4], [5], [49], mapping community structures [39], extracting cybersecurity neologisms [34], and classifying text to filter relevant discussions [17], [22], [31], [40]. However, these approaches rarely focus on the automated detection and aggregation of security-related events as they unfold within forum discussions. Aligning with similar conceptualizations in security text analysis [15], [55], we define a *security event* in this context as a collection of related forum activities, such as discussions, announcements, transactions or information sharing that center on a specific cybersecurity occurrence. This includes, but is not limited to, the emergence of zero-day vulnerabilities, the distribution or sale of malware and exploits, the disclosure of breached data, or the offering of illicit attack services.

Identifying these events is critical because hacker forums often serve as crucial early warning systems, surfacing discussions related to significant security incidents well before they appear in conventional threat intelligence reports or public disclosures [42], [50]. Because threat actors directly leverage these platforms to initiate, propagate, or orchestrate attacks, monitoring these emerging events offers potentially vital lead time for defensive measures. However, this research gap persists partly because CTI efforts have often prioritized platforms like Twitter, which offer relatively centralized data streams compared to the fragmented ecosystem of numerous and disparate hacker forums [15], [47], [60]. Furthermore, monitoring these forums poses inherent challenges, including user anonymity, widespread jargon, rapid content velocity, and deliberate obfuscation tactics [42].

Despite their analytical value, the volume, noise and fragmented nature of discussions across numerous forums render manual monitoring and timely identification of critical events operationally infeasible [16], [29]. This challenge directly contributes to analyst overload and alert fatigue within Security Operations Centers (SOCs) [13], [29]. Even after initial filtering identifies potentially relevant forum posts, the large volume can exceed analysts' capacity for investigation. Therefore, analysts often lack the resources to thoroughly investigate all potentially true signals, not just filter false positives [48]. Addressing this matter requires moving beyond simple filtering to effective prioritization, allowing analysts to focus limited resources on the events posing the greatest potential risk. Therefore, our primary objective is to develop EventHunter, an automated framework designed for both detecting emerging security events through clustering fragmented discussions and enabling systematic event ranking based on quantifiable metrics reflecting potential impact and importance. This approach aims to provide analysts with timely, actionable intelligence, facilitating focused investigation and proactive response. The main contributions of this paper are threefold:

1) We propose EventHunter, an unsupervised framework for dynamically detecting security events by clustering

fragmented forum posts using a novel entity-aware contrastive text embedding model specifically designed for the linguistic characteristics of hacker forums.
2) We introduce a systematic, data-driven event prioritization methodology that objectively ranks detected event clusters based on quantifiable metrics for timeliness, relevance, credibility, and completeness, providing analysts with actionable intelligence.
3) We present a comprehensive empirical evaluation and measurement study using large-scale, real-world hacker forum data, demonstrating EventHunter's effectiveness in identifying coherent security events and prioritizing high-impact threats, supported by detailed performance metrics and case studies.

The rest of this paper is organized as follows: Section II discusses related work. Section III details the EventHunter framework and its components. Section IV presents the experimental setup and evaluation results. Section V discusses the findings, limitations, and implications, including the identification of potential key actors. Finally, Section VI concludes the paper.

II. Related Work

Event detection is a central task in information retrieval and natural language processing, aimed at identifying and characterizing evolving events from data streams. This task has been explored across various domains, including natural disasters [7], [27], public health surveillance [43], [61], and real-time news tracking [3]. A core challenge in event detection lies in extracting semantically coherent occurrences from high-volume, unstructured data, while effectively managing constraints related to scale, velocity, and noise [23].

In cybersecurity, identifying events early is fundamental for effective cyber-threat intelligence (CTI) [13]. Online platforms, particularly social media like Twitter and underground forums, serve as critical CTI sources, often providing early indicators of vulnerabilities, threat actor activities, and illicit services [50], [62].

Recent work has validated the importance of this area; notably, Paladini et al. [42] conducted a large-scale longitudinal study demonstrating that forum discussions often precede official security reports. Their work provides crucial measurement-based validation for monitoring these platforms, answering "*are forums still early?*". Our work addresses a different, complementary problem: we propose EventHunter, an unsupervised framework designed not for historical analysis, but to answer "*what specific event on this forum should I investigate right now?*", thereby shifting the focus to operational intelligence generation.

Early detection methods frequently used trigger-based techniques using predefined keyword sets to identify potentially relevant security posts [52], [53]. While computationally efficient, such static approaches are not efficient in adversarial environments like hacker forums, where obfuscated language, neologisms, and evolving terminology limit their recall and robustness.

More recent approaches have advanced beyond simple keyword matching by incorporating semantic representations derived from contextual embeddings [15], [55], [60]. This innovation enables better detection of novel or linguistically varied threats missed by keyword-based systems. However, many such methods were primarily developed or evaluated on relatively centralized platforms (e.g., Twitter) or more formal text sources. Their direct applicability is limited when dealing with the unique characteristics of the hacker forum ecosystem, which is marked by high noise levels, fragmented discussions across numerous disparate platforms, and rapid shifts in language and topics [42]. Furthermore, much existing research has concentrated on supervised classification of individual posts, such as identifying security incident reports [21], [32], [60] or discussions about data breaches [19], [22]. While useful for initial filtering, classifying individual posts differs significantly from identifying broader thematic trends (topic modeling) or, as addressed in our work, aggregating fragmented posts into coherent, actionable *events*. Our work tackles this more complex challenge: unsupervised clustering of semantically related posts into distinct events, a capability particularly crucial given the dynamic nature of threat landscapes where specific incidents unfold across multiple discussions over time.

While standard semantic embeddings capture general context, their precision for distinguishing specific security events can be reduced by the noise and ambiguity prevalent in forum data. Named Entities (NEs)—such as specific malware names, vulnerability identifiers (CVEs), threat actor handles, and targeted organizations—serve as crucial anchors in cybersecurity discussions, often defining the core elements of a security event [6], [47]. Leveraging NEs to link and group related posts offers a potentially more robust approach to event detection by focusing on these shared critical elements [33]. For instance, discussions across multiple threads or forums referencing the same specific vulnerability identifier (e.g., CVE-2025-XXXX) likely pertain to the same underlying security event. EventHunter directly incorporates this insight by employing an unsupervised, entity-aware contrastive learning approach tailored to generate embeddings sensitive to these key entities within noisy forum text.

A second significant limitation in the existing literature is the general absence of principled event prioritization mechanisms. While some studies perform post-clustering ranking [9], [15], they often rely on simple heuristics like cluster size or recency, which may correlate poorly with the actual impact or urgency of the detected event. High-severity events might initially generate only sparse discussion, while lower-relevance topics could produce substantial post volume. Effective CTI necessitates not only detection but also triage capabilities to help analysts prioritize events according to operational relevance and potential risk.

Current CTI quality frameworks emphasize dimensions such as timeliness, credibility, completeness, and relevance as essential for actionable intelligence [26], [56], [63]. However, existing automated event detection and ranking models largely

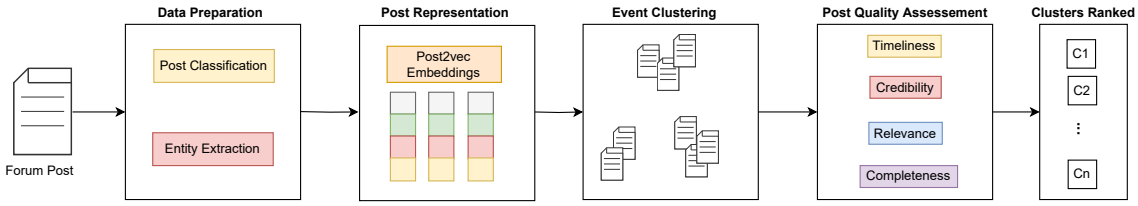

Fig. 1. Overview of the EventHunter framework: Forum posts are processed through data preparation (including classification and entity extraction) and transformed into dense, entity-aware post embeddings. These embeddings are then used to group related posts into event clusters. Finally, these clusters are assessed based on key quality metrics (Timeliness, Credibility, Relevance, Completeness) and ranked for analyst prioritization.

overlook the systematic integration of these factors. Our work addresses this gap through a dedicated prioritization module that evaluates discovered event clusters based on quantifiable metrics aligned with these four key dimensions. This methodology aims to shift CTI systems from purely detection-focused tools towards operationally valuable intelligence platforms that actively support analyst decision-making and resource allocation in time-sensitive SOC environments.

III. Methodology: The EventHunter Framework

This section presents the EventHunter framework, which describes our unsupervised methodology to dynamically discover and prioritize security events within hacker forums. We describe the process of transforming fragmented forum posts into rich representations, clustering related discussions into coherent event timelines, and ranking these events for analyst attention.

A. Problem Statement

The distributed and dynamic nature of hacker forums presents distinct challenges for security event detection compared to traditional CTI sources. Although vendor reports typically consolidate comprehensive details within well-defined documents, forum discussions produce fragmented and continuously evolving information dispersed across numerous individual posts. This fragmentation complicates the detection and aggregation of security events, requiring a formal framework for tracking and connecting related content.

Formally, we approach this challenge by processing a continuous stream of hacker forum posts:

$$\mathcal{P} = \{p_1, p_2, \ldots, p_n\},$$

Our primary objectives are to:

1) Derive *security event clusters*:

$$\mathcal{C} = \{c_1, c_2, \ldots, c_m\},$$

where each cluster c_i groups forum posts $p \in \mathcal{P}$ discussing a specific security incident.

2) Generate a prioritized list of these events:

$$\mathcal{R} = \text{sort}\left(\{(c_1, s_1), (c_2, s_2), \ldots, (c_m, s_m)\}\right),$$

where s_i represents the calculated **Priority Score** for cluster c_i, and the list $\mathcal{R}$ is ordered by s_i to guide the analyst's attention.

Formally, this process can be expressed as:

$$\mathcal{C} = f_{\text{cluster}}(\mathcal{P}), \quad \mathcal{R} = f_{\text{rank}}(\mathcal{C}),$$

where f_{cluster} encapsulates the automated clustering mechanism that uses content similarity and contextual features, and f_{rank} quantifies the priority of each cluster based on metrics reflecting the timeliness, relevance, credibility, and completeness derived from the constituent posts.

B. System Overview

As depicted in Figure 1, the EventHunter framework operates through five sequential stages:

Data Acquisition. We utilized the CrimeBB dataset [44], encompassing over 91 million posts from 43 distinct cybercrime forums. The scale of this dataset provides a comprehensive foundation for observing global cybercriminal activities and discussions.

Data Preparation. Raw forum data is inherently noisy and unstructured. To prepare the acquired posts for effective representation and analysis, we perform essential preprocessing steps. Initially, post classification filters the data stream to retain relevant cybersecurity discussions and reduce noise [17]. After that, Named Entity Recognition (NER) is applied to extract structured cybersecurity entities from these posts [12], [41]. Both the classifications and the extracted entities serve as crucial inputs, significantly informing the downstream clustering process and the final event prioritization ranking.

Post Representation. Each incoming forum post p is transformed into a meaningful vector representation. As described in Section III-D, we generate a dense, entity-aware embedding $\mathbf{h}_p$ for each post using a fine-tuned Transformer model trained with a contrastive learning objective to capture semantic and entity-based relatedness necessary for identifying security events.

Event Clustering. With posts represented as dense embeddings, this stage groups semantically related discussions. As detailed in Section III-E, we employ an unsupervised clustering algorithm, specifically HDBSCAN [10], to identify distinct event clusters $\mathcal{C}$ by grouping proximal post embeddings in the vector space while designating noise points.

Event Prioritization Ranking. Finally, to guide the analyst's attention to the most critical findings, the framework performs a ranking of all active event clusters. This ranking is based on a calculated Priority Score for each cluster, derived from

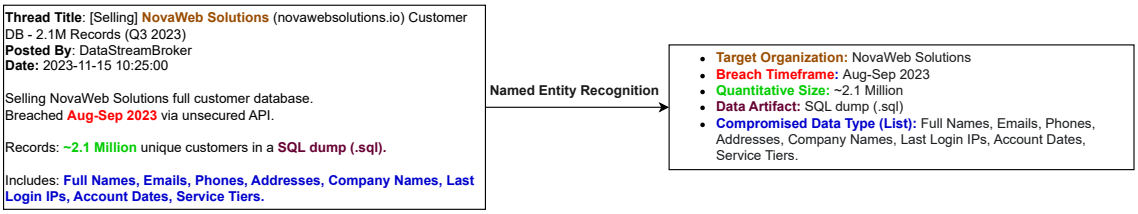

Fig. 2. Illustration of CTI entity identification within a forum post announcing a data breach.

metrics assessing its timeliness, relevance, credibility, and completeness [11], [26].

C. Data Preparation

Post Classification.

CTI extraction increasingly relies on unstructured data from sources such as hacker forums. However, these platforms contain substantial non-relevant content, with actionable intelligence constituting only a small fraction of posts. This necessitates effective classification methodologies that extend beyond traditional keyword-based filtering [15], [42]. Recent advances leverage Transformer-based models (e.g. BERT) to identify security-relevant content. While much prior work focused on binary relevance filtering [22], [40], [42], this treats all relevant content as homogeneous. Recognizing the need for finer granularity, other approaches, similar to ours, have employed multi-class classification to assign posts to specific security categories (e.g., vulnerabilities, malware, data breaches) from a predefined set [15], [17]. This level of detail, distinguishing between different types of security discussions, is critical for the accurate event clustering and meaningful feature construction used in downstream event prioritization, which are central goals of our work.

To support fine-grained threat differentiation, we perform multi-class classification by assigning each relevant post to its primary security category. To address the computational constraints of processing the full corpus, we first apply keyword-based filtering using terms drawn from prior research [42], [53] and domain-specific vocabulary identified during preliminary exploration. From this filtered subset, we constructed a labeled dataset of 14,386 posts sampled from various forums within the CrimeBB corpus [44]. This dataset was manually annotated, with each post assigned to the most appropriate category based on its main topic. The annotation schema includes six categories informed by established cybersecurity taxonomies and common underground forum themes [15], [17], [52], [53]: *Irrelevant*, *DataBreach*, *Malware/Ransomware*, *Vulnerability*, *Fraud/Phishing*, *DoS/DDoS*, and *BrandMonitoring*. Table I provides the distribution of these categories in our dataset, along with annotation guidelines and illustrative examples. The *BrandMonitoring* category was specifically introduced to capture threats targeting specific organizations, such as attack plans, which are highly relevant to CTI but frequently underrepresented in technical taxonomies.

By assigning specific threat types, this categorization allows for more targeted analysis and noise reduction compared to basic relevance filtering.

Named Entity Recognition (NER). Following post classification, we apply Named Entity Recognition (NER) to extract structured information from forum posts. In the cybersecurity domain, NER identifies entities such as malware names, vulnerability identifiers, threat actors, tools, IP addresses, hashes, and targeted organizations [6], [47]. This structured extraction supports the core components of the EventHunter framework, including both event clustering and prioritization.

Applying standard NER models to hacker forums is non-trivial due to domain-specific challenges. Although specialized cybersecurity NER datasets exist (for example, APTNER [58], DNRTI [59]), they are primarily derived from formal texts such as CTI reports and blogs. As a result, these models face two main limitations when transferred to forums.

1) **Linguistic Disparity:** Forum language includes slang, misspellings, abbreviations, and informal syntax [42]. Models trained on formal language often fail to recognize entities expressed in this irregular form.
2) **Information Granularity:** Forum content frequently contains operational details, such as payloads, compromised databases, or specific attack vectors. Existing NER schemas typically abstract these details, focusing instead on high-level threat indicators. Figure 2 provides an example that highlights the extraction of such operational entities from a data breach announcement.

To address these limitations and avoid the cost of developing large annotated forum-specific corpora, we explore zero-shot NER using open Large Language Models (LLMs). Previous work shows that LLMs, when guided by prompt engineering, can yield competitive results on NER tasks [8], [14], including in few-shot and zero-shot settings. Their pretraining on diverse corpora enables them to generalize across writing styles, making them well-suited for processing informal forum content.

The entities extracted via NER serve two primary functions within our pipeline. First, they provide essential signals for training the entity-aware dense embedding model, enabling it to capture event-specific semantics more effectively and link posts containing shared indicators. Second, the extracted entities and their associated metadata (e.g., entity type, frequency) are used by the prioritization module to assess event characteristics such as topical relevance and potential impact.

TABLE I
DISTRIBUTION OF CATEGORIES IN THE FORUM POST CLASSIFICATION DATASET WITH ANNOTATION GUIDELINES.

Category	# Posts (Ratio)	Classification Criteria	Example Post Excerpt
Irrelevant	5,925 (41.19%)	Forum posts unrelated to actionable security intelligence, including general discussions, advertisements, etc.	*"Looking for pentesting team members with experience in wireless networks and physical security assessments."*
Security-Relevant	8,461 (58.81%)		
DataBreach	3,219 (22.38%)	Posts discussing unauthorized access to systems, exfiltration of sensitive information, or exposure of personal/corporate data.	*"The database contains over 8.4M records including full names, email addresses, and hashed credentials from the compromised financial institution."*
Malware/Ransomware	2,674 (18.59%)	Posts about malicious software, including detection, analysis, development, distribution, or ransomware activity.	*"New loader bypasses Windows Defender by leveraging legitimate DLL sideloading technique and encrypting its C2 communications."*
Vulnerability	722 (5.02%)	Posts revealing or discussing security weaknesses in software, hardware, or protocols, including zero-days and recently patched flaws.	*"The buffer overflow in the authentication module can be triggered with a specially crafted request to achieve remote code execution."*
Fraud/Phishing	816 (5.67%)	Posts related to social engineering attacks, credential harvesting campaigns, or financial scams.	*"Selling fresh phishing kit that mimics the new Microsoft 365 login page with anti-bot protection and integrated Telegram notifications."*
DoS/DDoS	890 (6.19%)	Posts discussing denial-of-service attacks, botnets, amplification techniques, or mitigation methods.	*"Our booter service now supports layer 7 attacks with custom headers and bypass for common WAF implementations."*
BrandMonitoring	140 (0.97%)	Posts about targeted threats, impersonation, leaked assets, or attacks linked to a specific brand/organization.	*"Selling access to internal network of [Company Name], including employee credentials and source code."*
Total	**14,386 (100.0%)**		

D. Post Representation

A critical step for effective event clustering is transforming raw hacker forum posts into meaningful vector representations that capture their core content and relatedness in the security domain. Our objective is to map each post p to a single, dense vector $\mathbf{h}_p \in \mathbb{R}^d$ such that posts discussing the same underlying security event are mapped to vectors that are proximal in the embedding space, while vectors for posts about different events are distant.

Entity-Aware Dense Embeddings. Dense contextual embeddings, generated by large pre-trained Transformer models like BERT [18], excel at capturing rich semantic meaning and contextual nuances. Leveraging this capability, these embeddings form the foundation of our post representation strategy. However, standard Transformer embeddings distribute attention relatively uniformly. This can underemphasize the crucial signal carried by specific, often rare, domain-specific Named Entities (NEs)—like unique CVE IDs, malware hashes, or threat actor handles—which are fundamental identifiers for distinguishing concrete security events [51], [54]. Consequently, relying solely on standard dense representations may make it difficult to effectively differentiate posts discussing related concepts generally from those pertaining to a specific, actionable event defined by key NEs.

To explicitly enhance the representation's sensitivity to these critical identifiers, we employ an entity-aware BERT model (illustrated in Figure 3).Inspired by approaches integrating entity information into representations for related tasks [36], [51], we adapt the standard Transformer architecture to integrate information about identified NEs. As extracted in Section III-C, each token in a post is labeled based on whether it belongs to a recognized security-relevant entity. This entity presence information is then incorporated directly into the model's input layer. We train a dedicated entity presence embedding layer, mapping tokens to distinct embeddings based on their entity status (e.g., belonging to an entity, not belonging to an entity, or padding). During the forward pass, the standard input embeddings (token, segment, and position embeddings from the base Transformer) are augmented by summing them with the corresponding entity presence embedding for each token, *before* feeding the composite embedding into the Transformer encoder layers:

$$\mathbf{E}_{\text{input}} = \mathbf{E}_{\text{token}} + \mathbf{E}_{\text{segment}} + \mathbf{E}_{\text{position}} + \mathbf{E}_{\text{entity presence}}$$

This input-level augmentation encourages the model's internal representation process to give specific consideration to tokens marked as part of an entity, without requiring changes to the core Transformer layers themselves. After the Transformer encoder, the final hidden states are pooled using the CLS token representation, and then passed through an MLP layer to produce the fixed-size dense post embedding vector $\mathbf{h}_p$.

Contrastive Fine-Tuning Objective. To further ensure the resulting entity-aware embeddings are optimally structured for event clustering—where proximity in the vector space directly corresponds to event relatedness—we fine-tune the model using a combined contrastive learning objective. Contrastive learning aims to pull representations of "positive" pairs (e.g., posts from the same event) closer while pushing "negative" pairs (e.g., posts from different events) apart. Our approach integrates two complementary post-level contrastive loss functions. Figure 4 illustrates this fine-tuning process.

The first component is a Post Triplet Margin Loss ($\mathcal{L}_t$). This loss operates on triplets of (anchor post, positive post, negative post), where the positive post belongs to the same ground-truth event as the anchor, and the negative post belongs

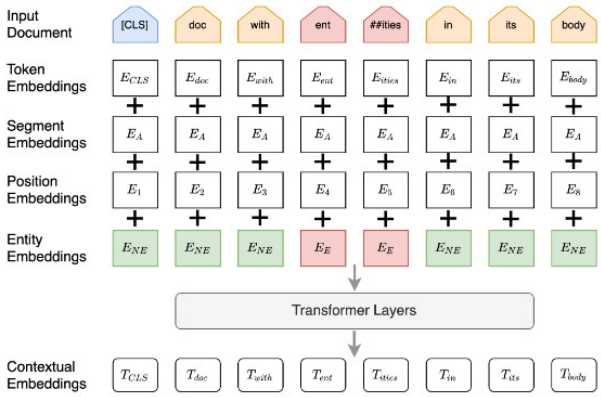

Fig. 3. Illustration of the entity-aware BERT model architecture, showing the addition of entity presence embeddings to the standard input embeddings before the Transformer layers.

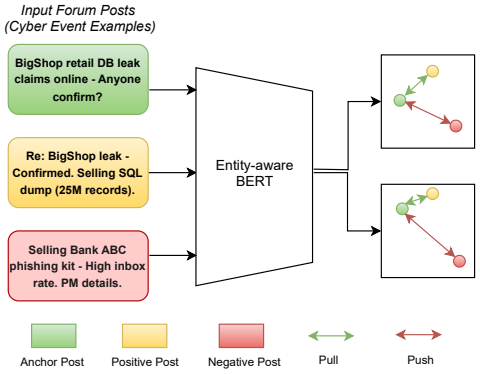

Fig. 4. Illustration of the combined contrastive fine-tuning process for the entity-aware BERT model. The model is trained using Post Triplet ($\mathcal{L}_t$) and Post Pairwise ($\mathcal{L}_p$) losses to structure the embedding space such that embeddings of posts from the same event are closer than those from different events.

to a different ground-truth event. The objective is to ensure that the embedding of the anchor post ($\mathbf{h}_a$) is closer in the embedding space (using cosine similarity $\text{sim}(\cdot,\cdot)$, where higher similarity implies smaller effective distance) to the embedding of the positive post ($\mathbf{h}_p$) than it is to the embedding of the negative post ($\mathbf{h}_n$), by at least a defined margin α. The loss is formulated as:

$$\mathcal{L}_t = \max\left(\text{sim}(\mathbf{h}_a, \mathbf{h}_n) - \text{sim}(\mathbf{h}_a, \mathbf{h}_p) + \alpha, 0\right) \quad (1)$$

Minimizing this loss pulls $\mathbf{h}_a$ and $\mathbf{h}_p$ together while pushing $\mathbf{h}_a$ and $\mathbf{h}_n$ apart based on a sampled hard negative post.

The second component is a Post Pairwise Cosine Loss ($\mathcal{L}_p$), inspired by SimCSE [25]. This loss leverages in-batch negatives. For a batch of N anchor-positive pairs (derived from N distinct original posts), resulting in $2N$ post embeddings ($\mathbf{h}_{a,1}, \ldots, \mathbf{h}_{a,N}$ and $\mathbf{h}_{p,1}, \ldots, \mathbf{h}_{p,N}$), the similarity matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$ is computed where $S_{ij} = \text{sim}(\mathbf{h}_{a,i}, \mathbf{h}_{p,j})/\tau$, scaled by a temperature parameter τ. The loss is then the standard cross-entropy between this similarity matrix and the identity matrix, effectively training the model to predict that $\mathbf{h}_{a,i}$ corresponds to its positive pair $\mathbf{h}_{p,i}$ among all positive embeddings in the batch:

$$\mathcal{L}_p = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\text{sim}(\mathbf{h}_{a,i}, \mathbf{h}_{p,i})/\tau}}{\sum_{j=1}^{N} e^{\text{sim}(\mathbf{h}_{a,i}, \mathbf{h}_{p,j})/\tau}} \quad (2)$$

This loss encourages intra-pair compactness and inter-pair separation using a dynamic set of negatives within each batch.

The overall fine-tuning objective is a weighted sum of these two post-level contrastive loss components: $\mathcal{L}_{\text{total}} = w_t\mathcal{L}_t + w_p\mathcal{L}_p$, where w_t, w_p are configurable weights. This combined strategy leverages the complementary strengths of hard negative sampling ($\mathcal{L}_t$) and in-batch negative sampling ($\mathcal{L}_p$) to produce dense, entity-aware post embeddings that effectively capture event relatedness for the subsequent clustering phase.

E. Event Clustering

With each post transformed into a dense, entity-aware embedding $\mathbf{h}_p$, the next step is to group these representations into clusters that correspond to discrete security events discussed across the forum corpus (f_{cluster} in the Problem Statement). Given the unsupervised nature of the problem—we do not know the number or exact boundaries of events beforehand—and the potential for noise and variations in cluster density within real-world forum data, we require a robust clustering algorithm capable of discovering arbitrary cluster shapes and effectively identifying noise points.

Based on these requirements, we employ Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [10]. HDBSCAN is a powerful density-based clustering algorithm. Unlike partition-based methods, HDBSCAN transforms the space into a hierarchy of density-based clusters. It then uses a technique to extract a flat partitioning from this hierarchy. This unique capability allows HDBSCAN to discover clusters of varying densities within the data and is particularly effective at distinguishing core clusters from background noise.

In our framework, after generating the dense embeddings for a set of posts, we apply the HDBSCAN algorithm to the embedding space. The clustering algorithm is configured to group a minimum number of related posts considered sufficient to constitute a potential ”event” cluster. Posts that the algorithm does not assign to any cluster are treated as noise.

The output of this clustering process is the set of discovered event clusters $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$. Each cluster c_i comprises the post IDs whose embeddings were grouped together by the algorithm. These clusters then serve as the input for the subsequent event prioritization phase, enabling us to rank the discovered events based on their estimated operational significance.

F. Event Prioritization Ranking

A core contribution of EventHunter is its ability not only to detect security event clusters but also to prioritize them, guiding security analysts towards the most critical or impactful ongoing events discussed within hacker forums. Drawing from established CTI quality frameworks [26], [56], [63], our prioritization mechanism is designed to integrate four key dimensions operationally relevant to security analysts: Timeliness (T), Relevance (V), Credibility (R), and Completeness (C). This score, calculated daily as a **Priority Score** s_i for each active event cluster $c_i \in \mathcal{C}$, is designed to reflect the potential operational importance of the event.

We quantify each dimension for a given cluster c_i based on properties derived from its constituent posts $\{p \mid p \in c_i\}$, associated metadata from the CrimeBB schema (e.g., member reputation, post timestamps), and entities extracted by our NER component.

Timeliness (T): This metric captures both the recency and the level of activity associated with the event cluster. It combines two components:

- *Recency (Rec)*: Measures how recently the latest post in the cluster appeared, using an exponential decay function:

$$Rec(c_i) = \exp\left(-\frac{t_{\text{now}} - t_{c_i}^{\max}}{\tau_{Rec}}\right) \tag{3}$$

  where t_{now} is the current time, $t_{c_i}^{\max}$ is the timestamp of the latest post in cluster c_i, and τ_{Rec} is a hyperparameter controlling the decay rate.
- *Activity (Act)*: Reflects the volume of discussion within the cluster, measured by the logarithm of the number of posts $N(c_i)$ in the cluster:

$$Act(c_i) = \log(1 + N(c_i)) \tag{4}$$

The final Timeliness score is a weighted combination:

$$T(c_i) = w_{Rec} \cdot Rec(c_i) + w_{Act} \cdot Act(c_i) \tag{5}$$

where w_{Rec} and w_{Act} are non-negative weights. This formulation prioritizes events that are both recent and have generated significant discussion.

Relevance (V): This metric assesses the cluster's relevance to specific analyst interests, which can be represented by an optional query Q. When a query is provided, the relevance $V(c_i, Q)$ is calculated based on the semantic similarity between the cybersecurity entities present in the cluster and those specified or implied by the query. It is computed as:

$$V(c_i, Q) = f_{\text{match}}(\text{Entities}(c_i), \text{Entities}(Q)) \tag{6}$$

where $\text{Entities}(c_i)$ denotes the set of unique cybersecurity entities extracted from the posts in cluster c_i (via NER, Section III-C), and $\text{Entities}(Q)$ represents the set of target entities derived from the analyst query Q. The function f_{match} specifically calculates the semantic similarity between these two sets of entities. If no query Q is provided by the analyst, this component defaults to zero, i.e., $V(c_i, \emptyset) = 0$.

Credibility (R): This metric estimates the reliability of the information within the cluster based on the reputation or standing of the contributing authors within the forum community. Let $\mathcal{A}(c_i)$ denote the set of unique authors who contributed posts to cluster c_i. The credibility score is computed as the average reputation score across these unique authors:

$$R(c_i) = \frac{1}{|\mathcal{A}(c_i)|} \sum_{a \in \mathcal{A}(c_i)} \text{Reputation}(a) \tag{7}$$

where $|\mathcal{A}(c_i)|$ is the number of unique authors in the cluster. The `Reputation(a)` for each author is derived directly from user metadata available in the CrimeBB schema, which include metrics such as post count, join date, or explicit reputation points awarded by other forum members. A higher average reputation score among the cluster's contributors suggests potentially more credible or influential discussions.

Completeness (C): This metric evaluates the richness and diversity of information contained within the event cluster, based on the variety and quantity of extracted cybersecurity entities. It is calculated as:

$$\begin{aligned} C(c_i) = {} & w_\alpha \cdot \log(1 + |\text{unique\_entities}(c_i)|) \\ & + w_\beta \cdot \log(1 + |\text{unique\_entity\_types}(c_i)|) \end{aligned} \tag{8}$$

where $|\text{unique\_entities}(c_i)|$ is the count of unique entity instances (e.g., specific CVEs, malware hashes) extracted from posts in c_i by the NER component, $|\text{unique\_entity\_types}(c_i)|$ is the count of distinct entity types (e.g., 'Vulnerability', 'Malware', 'Threat-Actor') present among the extracted entities, and w_α, w_β are weighting parameters. A higher score indicates a more developed event narrative with diverse indicators.

G. Priority Score Calculation

The final Priority Score s_i for an event cluster c_i is determined by aggregating the normalized scores [0, 1] from the four key dimensions: Timeliness (T), Relevance (V), Credibility (R), and Completeness (C). This aggregation is performed using a weighted linear combination:

$$s_i(Q) = w_T T(c_i) + w_V V(c_i, Q) + w_R R(c_i) + w_C C(c_i) \tag{9}$$

Here, the coefficients w_T, w_V, w_R, w_C represent non-negative weights assigned based on the perceived importance of each dimension for operational CTI. These weights are configurable and can be optimized based on empirical analysis or direct analyst input to reflect specific monitoring priorities.

Finally, on a regular basis, the active event clusters $c_i \in \mathcal{C}$ are ranked in descending order based on their calculated Priority Scores $s_i(Q)$, generating the prioritized list $\mathcal{R}$ intended for analyst review. This allows analysts to focus on the clusters deemed most operationally significant according to the defined criteria.

IV. Experiments and Evaluation

To rigorously evaluate the performance of the EventHunter framework in identifying and prioritizing security events

TABLE II
CHARACTERISTICS OF ANALYZED HACKER FORUMS

Forum	Posts	Threads	Members	Boards	Time Span
HackForums	42,474,325	4,148,196	716,058	212	Jan 2007 - Apr 2023
Nulled	6,675,497	591,830	1,647,057	168	Apr 2013 - Jun 2023
Cracked	2,977,800	419,517	897,760	163	Apr 2018 - Jun 2023
BreachForums	737,922	34,412	119,260	72	Mar 2022 - Mar 2023
KernelMode	26,815	3,606	1,668	11	Mar 2010 - Nov 2019

within complex hacker forum data, we conducted a comprehensive set of experiments. This section details the experimental methodology, including the datasets used, the specific configurations tested, and the metrics employed to assess each stage of the framework, from data preparation to final event ranking.

A. Experimental Setup

Dataset Description:

Our empirical evaluation leverages data from five prominent English-language forums within the CrimeBB dataset [44]: *Nulled* and *Cracked*[1], *BreachForums*[2], *KernelMode*[3], and *HackForums*[4].

These platforms were specifically selected due to their central roles within the cybercriminal ecosystem and their rich, diverse discussions spanning malware development, vulnerability exploitation, data breaches, and advanced hacking methodologies. As detailed in Table II, our dataset encompasses millions of posts across multiple years, providing a robust and representative cross-section of contemporary underground security discourse. This carefully curated data selection enables rigorous assessment of EventHunter's performance in accurately identifying and prioritizing significant security events from extensive and noisy forum interactions.

Evaluation Data & Ground Truth: Evaluating unsupervised clustering algorithms, particularly in specialized domains like cybersecurity event detection from noisy forum data, necessitates the creation of reliable ground truth. Standard labeled datasets for fine-grained security *events* spanning multiple fragmented posts are largely unavailable. Therefore, to assess the ability of EventHunter to group related posts into coherent event clusters, we constructed a ground truth dataset based on well-documented, distinct security incidents discussed within the forums used in our study.

Specifically, we identified a set of 70 known, significant security events that generated traceable discussions within our forum corpus. These included incidents such as major vulnerability disclosures (e.g., CVE-2022-42475, a critical FortiOS RCE vulnerability), specific ransomware campaigns (e.g., discussions surrounding LockBit 3.0 operations), and widely publicized data breaches (e.g., the leak involving the Turkish Public Health System, HSYS). For each such reference event, we meticulously curated a ground truth cluster by: (1) Identifying posts definitively discussing that specific event using unique and unambiguous markers (e.g., canonical vulnerability identifiers like 'CVE-2022-42475', established malware family names, specific targeted entity names like 'HSYS' coupled with incident context) within a relevant time window. (2) Verifying through **manual inspection** that the collected posts genuinely pertain to the same underlying security incident and were not merely tangential mentions.

This process yielded a collection of disjoint sets of forum posts, where each set represents a single, distinct real-world security event (e.g., one set for all posts discussing CVE-2022-42475, another for the HSYS leak). This curated collection serves as the ground truth against which we evaluate the clustering component of EventHunter. While acknowledging the inherent challenges in creating exhaustive ground truth for dynamic events, this approach provides a robust benchmark based on verifiable real-world occurrences discussed in the forums.

Implementation Details: We implemented the EventHunter framework using Python (version 3.8), leveraging core libraries including PyTorch [45] for deep learning models and scikit-learn [46] for traditional methods and evaluation metrics. The embedding size for all Transformer models was 768. For clustering, HDBSCAN was configured with a `min_cluster_size` of 5. To ensure reproducibility, key components were configured as follows: Transformer models (used for post classification and generating dense embeddings, e.g., DarkBERT, RoBERTa) were fine-tuned using the AdamW optimizer [37] with a learning rate of 2×10^{-5}, a dropout rate of 0.1, and trained for 5 epochs. The margin α for the contrastive triplet loss component was set to 0.5, a common value in the contrastive learning literature. TF-IDF vectors were generated using standard scikit-learn configurations. The LLM-based NER component utilized zero-shot prompting via the Ollama library[5] with the *mistral-nemo* model. The source code for this paper is available at: https://github.com/yasirech-chammakhy/EventHunter.

Evaluation Metrics: We evaluate clustering performance using five standard metrics: Adjusted Rand Index (ARI) [28], which measures the similarity between two clusterings while adjusting for chance; Normalized Mutual Information (NMI) [57], which quantifies the mutual dependence between the predicted and true clusters. These metrics range from 0 to 1, with higher values indicating better clustering quality. Our implementation uses the scikit-learn library's clustering metrics module for consistent and reproducible evaluation.

B. Multi-Category Classification Model Evaluation

To focus downstream analysis on relevant content, EventHunter first employs a multi-category classification step. We evaluated several pre-trained Transformer models for this task: *BERT-base* [18], *RoBERTa* [35], *SecureBERT* [1], *CySecBERT* [24], and *DarkBERT* [30]. Each model was fine-tuned using a standard architecture comprising the base

[1]Major cybercrime forums taken down in Jan 2025 post-collection.
[2]Known for trading breached data/exploits; data precedes takedowns.
[3]Technical discussions on malware, system internals, vulnerabilities.
[4]Long-running platform covering diverse hacking topics.
[5]https://ollama.com/

TABLE III
PERFORMANCE COMPARISON OF TRANSFORMER MODELS ON FORUM POST CLASSIFICATION. RESULTS REPORTED AS F1 SCORES FOR EACH CATEGORY.

	Irrelevant	DataBreach	Malware	Vulnerability	FraudPhishing	DosAttack	BrandMonitoring	Avg. F1
BERT	0.7266	0.9367	0.8762	0.7326	0.5449	**0.9003**	0.2632	0.7115
RoBERTa	0.7478	**0.9478**	0.8721	0.7397	0.5359	0.8920	0.2727	0.7154
SecureBERT	0.7548	0.9414	0.8698	**0.7755**	0.4828	0.8857	**0.4091**	0.7313
CySecBERT	**0.7559**	0.9455	**0.8844**	0.7529	0.5231	**0.9003**	0.3043	0.7238
DarkBERT	0.7540	0.9399	0.8841	0.7626	**0.5460**	0.8952	0.3902	**0.7389**

Transformer followed by a classification head (dropout and linear layer on the [CLS] token representation) predicting one of the seven security-relevant categories identified in Table I.

Training utilized a weighted cross-entropy loss function to mitigate class imbalance, with optimizer, learning rate, and other hyperparameters detailed in Section IV-A. Table III summarizes the classification performance. Consistent with expectations for domain-specific text, models pre-trained on cybersecurity corpora (DarkBERT, CySecBERT) demonstrated superior effectiveness. DarkBERT achieved the highest average F1 score across categories and was therefore selected as the classification component for the EventHunter pipeline. This step yields a filtered and categorized set of posts pertinent to security discussions, preparing the data for representation learning and event clustering.

C. NER Performance

Named Entity Recognition (NER) is crucial for EventHunter to identify key actors, tools, vulnerabilities, and targets that anchor security events within forum discussions. Extracted entities inform both representation learning and event prioritization. To select an effective NER component without requiring extensive fine-tuning, we evaluated several recent open Large Language Models (LLMs) using zero-shot prompting on the standard CyNER dataset [2]. Specifically, we assessed *gemma2*[6], *mistral-nemo*[7], and *mistral:7b-instruct*[8].

Performance was measured by mention-level F1 score. *Mistral-nemo* achieved the highest F1 (52.13), outperforming *gemma2* (49.39) and *mistral:7b-instruct* (42.10). None of these LLMs were fine-tuned on CyNER or any cybersecurity-specific corpus prior to this zero-shot evaluation. Notably, this zero-shot performance surpasses the supervised RoBERTa-base benchmark reported in [15] (F1 score of 39.7), despite using no task-specific supervision. Consequently, we adopt *mistral-nemo* with zero-shot prompting, aligned to our predefined entity schema, as the default NER component in EventHunter. The entities extracted by this component serve key roles within the pipeline, providing crucial signals for representation learning and metadata for event prioritization.

[6]https://blog.google/technology/developers/google-gemma-2/
[7]https://mistral.ai/news/mistral-nemo
[8]https://mistral.ai/news/announcing-mistral-7b
[1]https://blog.google/technology/developers/google-gemma-2/
[2]https://mistral.ai/news/mistral-nemo
[3]https://mistral.ai/news/announcing-mistral-7b

TABLE IV
FINDING 1: TRANSFORMER EMBEDDINGS VS. BASELINES (BEST NMI PER TRANSFORMER MODEL TYPE SHOWN, USING STANDARD CONTRASTIVE LOSS)

Model / Method	ARI	NMI	Clusters Found
Baselines			
TF-IDF	0.264	0.638	12
Word2Vec (GloVe)	-0.024	0.222	4
Transformers (Best NMI - Standard Contrastive Loss)			
BERT (Pairwise)	0.159	0.534	9
DarkBERT (Pairwise)	0.347	0.712	14
RoBERTa (Triplet+Pairwise)	0.402	0.714	11
CySecBERT (Pairwise)	0.377	0.689	12

D. Clustering Results

We evaluated the effectiveness of EventHunter's event clustering component using the ground truth dataset derived from the curation process described in Section IV-A. From the initially identified set of 70 distinct security events, we randomly selected a subset of 21 events to serve as the test set for this clustering evaluation. This curated test set, representing a diverse range of incidents discussed in the forums, served as the ground truth against which we assessed how well different post embedding configurations group related posts. While this test set provides a valuable benchmark for relative comparison, evaluating performance against larger-scale event datasets remains an area for future work.

Finding 1: Fine-tuned Transformer embeddings significantly outperform traditional baselines. Learned contextual embeddings offer significant gains over traditional vectorization approaches. As shown in Table IV, TF-IDF and Word2Vec underperform (NMI 0.638 and 0.222 respectively), while RoBERTa and DarkBERT achieve substantially higher scores (NMI 0.714 and 0.712), validating the impact of deep contextual representation and contrastive fine-tuning.

Finding 2: Domain-Specific Pre-training Enhances Clustering. Comparing the best configurations for each Transformer architecture (Table V, using Pairwise Loss results for comparison) highlights the benefit of domain adaptation. Models pre-trained on relevant corpora (DarkBERT, CySecBERT) consistently outperform general-purpose models (BERT, RoBERTa). DarkBERT achieves the highest NMI, while CySecBERT attains the highest ARI. This indicates that familiarity with cybersecurity jargon, topics, and linguistic patterns acquired during pre-training enables these models to learn more discriminative embeddings for forum data.

Finding 3: Pairwise or combined contrastive objectives

TABLE V
FINDING 2: DOMAIN-SPECIFIC VS. GENERAL MODELS (BEST STANDARD CONTRASTIVE LOSS CONFIGURATION PER MODEL FROM INPUT ENTITY EXPERIMENTS)

Model Type (Best Standard Loss)	ARI	NMI	Clusters Found
General Domain			
BERT (Pairwise)	0.159	0.534	9
RoBERTa (Pairwise)	0.314	0.664	13
Cybersecurity Domain			
DarkBERT (Pairwise)	0.347	**0.712**	14
CySecBERT (Pairwise)	**0.377**	0.689	12

TABLE VI
FINDING 3: COMPARISON OF STANDARD CONTRASTIVE LOSSES (INPUT ENTITY EXPERIMENTS)

Model	Loss Configuration	ARI	NMI	Clusters Found
BERT	Triplet Loss	0.083	0.340	6
	Pairwise Loss	**0.159**	**0.534**	9
	Combined Loss	0.112	0.383	5
DarkBERT	Triplet Loss	0.240	0.652	12
	Pairwise Loss	**0.347**	**0.712**	14
	Combined Loss	0.303	0.696	12
RoBERTa	Triplet Loss	0.319	0.613	8
	Pairwise Loss	0.314	**0.664**	13
	Combined Loss	**0.342**	0.654	9
CySecBERT	Triplet Loss	0.328	0.665	10
	Pairwise Loss	**0.377**	**0.689**	12
	Combined Loss	0.327	0.668	11

outperform triplet-only loss. Contrastive loss formulations significantly influence clustering performance. Table VI and the NMI heatmap in Figure 5 show that triplet-only loss generally underperforms compared to pairwise or combined losses across models (using standard inputs, from Input entity experiments). For instance, DarkBERT achieves NMI 0.652 (triplet), 0.712 (pairwise), and 0.696 (combined), reinforcing the value of in-batch negative sampling over hard negative mining alone.

Finding 4: Input-Level Entity-Aware Mechanism Shows Model-Specific Benefits. We evaluated an input-level entity-aware mechanism, modifying input embeddings with entity indicators, against a standard contrastive baseline (Table VII). The results ('Input Entity+' vs. 'Standard Input') show benefits that depend on the base model and loss function. Notably, the mechanism significantly improved BERT performance with Combined Loss (NMI increased from 0.383 to 0.460) and yielded competitive results with DarkBERT (achieving higher ARI under Pairwise and Combined losses, e.g., 0.368 vs. 0.347 with Pairwise). However, the standard baseline generally performed better for RoBERTa and CySecBERT in our tested configurations (e.g., RoBERTa NMI 0.664 vs. 0.637 with Pairwise loss). While not universally superior in these tests, the input-level entity-aware approach demonstrates clear potential depending on the specific model and setup, suggesting a promising direction for further tuning.

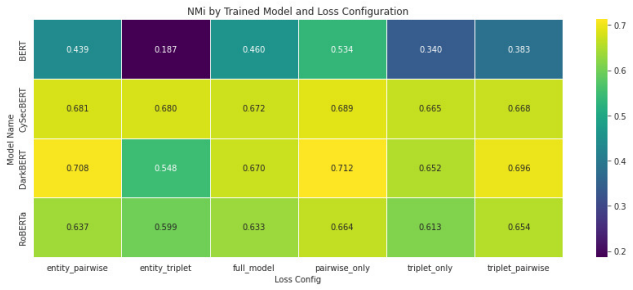

Fig. 5. NMI Scores by Model and Loss Configuration (Input Entity Experiments).

Finding 5: Output-Level Entity-Aware Loss Did Not Yield Benefits in Tested Configurations.

As an alternative entity-aware strategy, we explored augmenting the standard post-level contrastive objectives with an auxiliary *output-level entity contrastive loss* ($\mathcal{L}_e$). This loss aims to make the final post embedding $\mathbf{h}_p$ explicitly aware of the named entities within the post by pulling it closer to embeddings of mentioned entities (e^+) and pushing it away from embeddings of unmentioned entities (e^-). We evaluated adding this entity loss ($\mathcal{L}_e$, weighted by w_e) to the standard objectives ($\mathcal{L}_t + w_e\mathcal{L}_e$, $\mathcal{L}_p + w_e\mathcal{L}_e$, $\mathcal{L}_t + \mathcal{L}_p + w_e\mathcal{L}_e$).

The results indicate that incorporating this specific output-level entity-aware loss did not improve clustering performance and frequently led to substantial degradation compared to the standard contrastive objectives alone. For instance, comparing the standard Pairwise Loss ($\mathcal{L}_p$ only) to the configuration adding the entity loss ($\mathcal{L}_p + w_e\mathcal{L}_e$), RoBERTa's NMI dropped significantly from 0.702 to 0.120, and DarkBERT's NMI decreased from 0.592 to 0.339 (detailed results omitted for brevity). This suggests that directly forcing alignment between overall post embeddings and specific entity embeddings via this auxiliary loss interfered with learning representations optimal for the primary task of clustering posts based on the overall event context. Figure 6 visually compares the impact of the input-level versus output-level entity-aware mechanisms specifically for the BERT model across the different loss configurations. As shown by the orange bars (Output Entity Mechanism), performance (both NMI and ARI) is consistently low and generally worse than the corresponding results for the Input Entity Mechanism (blue bars), which itself showed mixed results (Finding 4). The output-level entity-aware mechanism, in particular, struggled to surpass even baseline levels achieved by standard contrastive fine-tuning. Consistent with Finding 4, the specific entity-aware mechanisms explored here, whether at the input or output level, proved less effective overall than standard contrastive fine-tuning for this task.

Overall Performance Summary. Fine-tuned Transformer embeddings effectively cluster security events from forums. The best performance came from standard contrastive fine-tuning on domain-adapted models: DarkBERT (Pairwise Loss) achieved the highest NMI (0.712), and CySecBERT (Pairwise Loss) the highest ARI (0.377). While an input-level entity-aware mechanism significantly improved BERT (Combined Loss) and yielded competitive results for DarkBERT (higher ARI), it wasn't universally better across all tested models

TABLE VII
FINDING 4: PERFORMANCE COMPARISON: STANDARD INPUT VS. INPUT-LEVEL ENTITY-AWARE MECHANISM

Model	Mechanism	Triplet Loss		Pairwise Loss		Combined Loss (Full)	
		ARI	NMI	ARI	NMI	ARI	NMI
BERT	Standard Input	0.083	0.340	0.159	0.534	0.112	0.383
	Input Entity+	0.050	0.187	0.059	0.439	**0.230**	**0.460**
DarkBERT	Standard Input	0.240	0.652	0.347	**0.712**	0.303	0.696
	Input Entity+	0.206	0.548	**0.368**	0.708	0.31	0.670
RoBERTa	Standard Input	0.319	0.613	0.314	**0.664**	**0.342**	0.654
	Input Entity+	0.278	0.599	0.217	0.637	0.286	0.633
CySecBERT	Standard Input	0.328	0.665	**0.377**	**0.689**	0.327	0.668
	Input Entity+	0.335	0.680	0.279	0.681	0.330	0.672

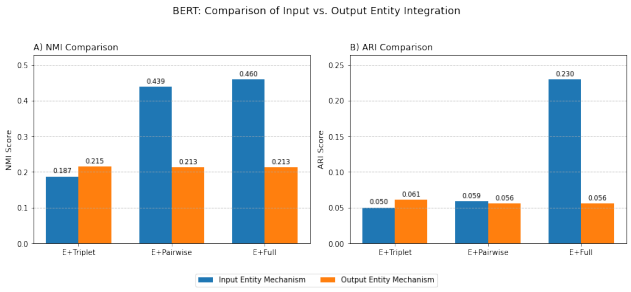

Fig. 6. Comparison of NMI (A) and ARI (B) for BERT using Input vs. Output-level entity integration across contrastive loss types.

and configurations. An output-level entity-aware loss proved ineffective. These results highlight the importance of domain adaptation and contrastive objectives, while suggesting the input-entity approach has model-specific potential that could be optimized in future work.

E. Ranking Performance

A key objective of EventHunter is to move beyond simple event detection towards actionable intelligence by prioritizing the discovered event clusters based on their potential operational significance. This addresses the critical challenge of analyst overload when faced with numerous alerts generated from noisy data sources like hacker forums [29]. We evaluate the prioritization mechanism by analyzing the characteristics of events ranked at different positions using qualitative case studies and by comparing the overall ranking order produced by EventHunter against simpler heuristics.

For this evaluation, we applied the EventHunter ranker using a default weighting scheme ($w_T = 0.35, w_V = 0.25, w_R = 0.20, w_C = 0.20$) designed to prioritize event timeliness, reflecting the operational need to surface recent and active threats, while giving balanced consideration to other factors. The relevance component was disabled ($V = 0$) to assess the general, query-agnostic ranking performance. The reference time was set to the latest post timestamp in the dataset, and the ranker was applied directly to the 70 curated ground truth event clusters. This allowed us to assess how the prioritization score behaves across a set of known, distinct events.

TABLE VIII
RANKING SCORES FOR CASE STUDY EVENTS.

Rank	Event (Approx. Date)	Score	T	R	C
1	Optus Breach (Sep/Oct 22)	0.414	0.630	0.584	0.382
19	ThaiTradeFair (Dec 22)	0.202	0.204	0.275	0.379
37	LockBit 3.0 (Jun-Dec 22)	0.090	0.010	0.350	0.083

1) Qualitative Analysis: Case Studies: To qualitatively assess the ranking's ability to surface operationally relevant events, we examined specific event clusters ranked at different positions (top, middle, bottom) by EventHunter. Table VIII presents the calculated priority scores (Overall Score, Timeliness T, Credibility R, Completeness C) for three representative examples from our curated ground truth set.

Rank 1: Optus Data Breach (Sep/Oct 2022). This high-profile event, associated with 52 posts in our dataset, achieved the top rank (Score: 0.414). As shown in Table VIII, its ranking is primarily driven by an excellent Timeliness score (T=0.630), reflecting recent and sustained activity, combined with strong Credibility (R=0.584) based on contributing authors. Its Completeness (C=0.382) was moderate. This outcome aligns with the goal of prioritizing major, ongoing incidents involving reputable forum members.

Rank 19: ThaiTradeFair.com Leak (Dec 2022). Representing a mid-tier rank (Score: 0.202), this less prominent data leak discussion comprised only 6 posts. Its lower Timeliness (T=0.204) and significantly lower author Credibility score (R=0.275) contributed to its moderate ranking. The event showed reasonable Completeness (C=0.379), indicating some relevant entities were mentioned despite the low post count.

Rank 37: LockBit 3.0 Infrastructure Discussion (June-Dec 2022). this cluster discussed older LockBit infrastructure and policies across 7 posts. Despite moderate author Credibility (R=0.350), its very low score stems directly from extremely poor Timeliness (T=0.010) and low Completeness (C=0.083). This correctly deprioritizes older, less detailed discussions relative to more current or richer events.

These case studies demonstrate that the Priority Score integrates multiple dimensions to produce a nuanced ranking. High ranks are not solely determined by activity (post count)

TABLE IX
COMPARISON OF TOP 3 RANKED EVENTS BY DIFFERENT METHODS

Rank	EventHunter	Recency Rank	Activity Rank
1	Optus Data Breach	Optus Data Breach	HSYS Leak
2	HSYS Leak	HSYS Leak	Banorte Leak
3	DDoS Tool Method	DDoS Tool Method	Optus Data Breach

or recency alone, but by a combination reflecting potential significance based on timeliness, credibility, and information richness. The system effectively differentiates between major ongoing events, smaller or less credible incidents, and older, less complete discussions based on the calculated metrics.

2) Comparison with Baseline Ranking Orders: We also compared the ranking order produced by EventHunter against two simpler baseline heuristics:

- **Recency Rank**: Events ranked solely by the timestamp of their latest post (most recent first).
- **Activity Rank**: Events ranked solely by the total number of posts within their cluster (highest count first).

Observing the top-ranked events for each method (Table IX) highlights the different perspectives provided. EventHunter's top rank (Optus) is also top-ranked by Recency but only third by Activity. HSYS, ranked second by EventHunter and first by Activity (due to its high post count of 116), is ranked second by Recency. The DDoS tool discussion, third in EventHunter's ranking, appears third in Recency but only eighth in Activity. This illustrates that EventHunter's multi-dimensional score produces a distinct ordering that balances factors beyond simple recency or volume, potentially offering a more holistic view of event significance compared to unidimensional heuristics.

Summary. The evaluation demonstrates EventHunter's event prioritization mechanism generates a plausible ranking based on Timeliness, Credibility, and Completeness. Qualitative case studies confirm the interpretability of the ranking and its ability to differentiate events based on these factors. Comparison with baseline methods (IV-E2) shows that EventHunter produces a distinct event ordering by integrating multiple dimensions, offering a potentially more comprehensive assessment of event significance than rankings based solely on recency or activity. The presented results highlight the functionality and potential utility of the proposed ranking approach for CTI analysts navigating high-volume forum data.

V. DISCUSSION

Our results demonstrate that EventHunter can effectively identify and cluster security events from noisy forum data. The strong performance of domain-specific Transformers fine-tuned with contrastive objectives highlights the importance of contextual representation for this task. The framework's ability to aggregate fragmented discussions into coherent, prioritized events offers a significant step towards managing the deluge of data from underground forums. However, the work has several limitations that open important avenues for future research.

A. Limitations

A primary limitation is the static nature of our current evaluation. The clustering is performed on a snapshot of the data and does not explicitly model the temporal evolution of events, which is critical for tracking threats as they unfold. Second, our exploration of entity-aware mechanisms, while showing model-specific promise (Finding 4), did not yield a universally superior integration strategy. A deeper investigation into how entity information can robustly enhance event clustering is still needed.

Third, the Priority Score's ranking is sensitive to the configured weights of its components (Timeliness, Credibility, etc.). Our evaluation used a default set of weights, but a comprehensive sensitivity analysis is required to understand how different operational priorities would alter the final event ordering. Finally, our metrics for credibility and timeliness are derived solely from internal forum metadata. They do not incorporate external ground truth, which could provide a more objective measure of an event's real-world impact and the timeliness of its detection.

B. Future Work and Broader Impact

These limitations highlight several promising research directions. Addressing the static nature of the evaluation, future work should focus on incorporating timestamps and the temporal relationships between posts to enable dynamic event tracking and more refined clustering. This would allow the system to model the entire lifecycle of a security event, from its inception to its resolution.

Regarding the prioritization mechanism, we acknowledge that our weighted linear combination is a baseline approach. Future research could explore more advanced multi-objective ranking techniques, such as skyline queries, to provide analysts with a more nuanced set of non-dominated "best" events across different dimensions, rather than a single linear ranking. Furthermore, the ranking could be validated and enriched by correlating its outputs with external data sources. For example, the framework's timeliness could be benchmarked against public threat intelligence reports to quantify the early-warning value of forum monitoring, and event completeness could be compared against official CVSS scores.

The framework also provides a foundation for identifying significant actors, by calculating an actor's aggregated priority score based on their participation in high-ranking events ($\text{ActorScore}(a) = \sum_{c_i \in \mathcal{C}_a} s_i$), one could develop an event-centric measure of user influence. Validating this data-driven approach against known threat actor profiles is a key next step.

Finally, scaling the framework for real-time, high-volume data ingestion and conducting empirical evaluations of its effectiveness in a live operational setting are essential for transitioning EventHunter from a research prototype to a practical tool for security analysts. Such a deployment would also allow for a direct assessment of how the system helps mitigate analyst workload and improves response times to emerging threats.

C. Ethical Considerations

Our research uses the CrimeBB academic dataset [44], sourced from public forums under a Data Use Agreement (DUA), which prohibits sharing the raw data. This data is sensitive and was handled ethically: analysis used only public text and involved no user interaction. To support reproducibility despite data restrictions, the source code for the EventHunter framework covering embedding model training, clustering, and prioritization is available at https://github.com/yasirech-chammakhy/EventHunter.

VI. Conclusion

This paper introduced EventHunter, an unsupervised framework designed to automatically detect and prioritize security events within noisy, fragmented hacker forum discussions. Our approach addresses fundamental challenges in extracting actionable intelligence by integrating entity-aware contrastive embeddings tailored for security semantics, robust density-based clustering for aggregating related posts into events, and a systematic prioritization mechanism based on CTI quality metrics. EventHunter demonstrates a scalable methodology for moving from raw forum data to prioritized, operationally relevant threat intelligence, enabling security analysts to focus effectively on the most significant emerging risks discussed within the cybercrime ecosystem.

Acknowledgment

This work was supported by a collaborative PhD program between Mohammed VI Polytechnic University (UM6P) and the Deloitte Morocco Cyber Center. We particularly thank Oussama Azrara from Deloitte for his continuous support and insightful operational feedback throughout this project. We gratefully thank the anonymous reviewers for their constructive feedback and the creators of the CrimeBB dataset.

References

[1] Ehsan Aghaei, Xi Niu, Waseem Shadid, and Ehab Al-Shaer. Securebert: A domain-specific language model for cybersecurity. *arXiv*, October 2022.

[2] Md Tanvirul Alam, Dipkamal Bhusal, Youngja Park, and Nidhi Rastogi. Cyner: A python library for cybersecurity named entity recognition, 2022. Accessed: 2025-03-15.

[3] Sarah A. Alkhodair, Steven H. H. Ding, Benjamin C. M. Fung, and Junqiang Liu. Detecting breaking news rumors of emerging topics in social media. *Information Processing & Management*, 57(2):102018, March 2020.

[4] Abdoul Nasser Hassane Amadou, Anas Motii, Saida Elouardi, and El Houcine Bergou. EUREKHA: Enhancing User Representation for Key Hackers Identification in Underground Forums. In *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 387–398. IEEE, 2024.

[5] Abdoul Nasser Hassane Amadou, Anas Motii, and Mohammed Jouhari. HC-HackerRank: Identifying Key Hackers in Cybercrime Social Network Forums. In *2024 7th International Conference on Advanced Communication Technologies and Networking (CommNet)*, pages 1–8. IEEE, 2024.

[6] Marco Arazzi, Dincy R. Arikkat, Serena Nicolazzo, Antonino Nocera, Rafidha Rehiman K. A., P. Vinod, and Mauro Conti. NLP-Based Techniques for Cyber Threat Intelligence. *arXiv preprint*, arXiv:2311.08807, November 15 2023.

[7] Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. Using twitter data to monitor natural disaster social dynamics: A recurrent neural network approach with word embeddings and kernel density estimation. *Sensors*, 19(7):1746, 2019.

[8] Dhananjay Ashok and Zachary C. Lipton. PromptNER: Prompting For Named Entity Recognition. *arXiv*, June 2023. arXiv preprint arXiv:2305.15444.

[9] Avishek Bose, Vahid Behzadan, Carlos Aguirre, and William H. Hsu. A novel approach for detection and ranking of trendy and emerging cyber threat events in twitter streams. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 871–878. ACM, 2019.

[10] Ricardo JGB Campello, Davoud Moulavi, and Joerg Sander. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining*, volume 7819, pages 160–172. Springer, 2013.

[11] Sheng-Shan Chen, Ren-Hung Hwang, Asad Ali, Ying-Dar Lin, Yu-Chih Wei, and Tun-Wen Pai. Improving quality of indicators of compromise using stix graphs. *Computers & Security*, 144:103972, September 2024.

[12] Sheng-Shan Chen, Ren-Hung Hwang, Chin-Yu Sun, Ying-Dar Lin, and Tun-Wen Pai. Enhancing cyber threat intelligence with named entity recognition using bert-crf. In *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, pages 7532–7537, Kuala Lumpur, Malaysia, 2023. IEEE.

[13] Yiren Chen, Mengjiao Cui, Ding Wang, Yiyang Cao, Peian Yang, Bo Jiang, Zhigang Lu, and Baoxu Liu. A survey of large language models for cyber threat detection. *Computers & Security*, 145:104016, October 2024.

[14] Qi Cheng, Liqiong Chen, Zhixing Hu, Juan Tang, Qiang Xu, and Binbin Ning. A Novel Prompting Method for Few-Shot NER via LLMs. *Natural Language Processing Journal*, 8:100099, September 2024.

[15] Jian Cui, Hanna Kim, Eugene Jang, Dayeon Yim, Kicheol Kim, Yongjae Lee, Jin-Woo Chung, Seungwon Shin, and Xiaojing Liao. Tweezers: A framework for security event detection via event attribution-centric tweet embedding. arXiv preprint arXiv:2409.08221, 2024.

[16] Cybersixgill. Cyber threat intelligence survey, 2024. Accessed: 2025-02-27.

[17] Isuf Deliu, Carl Leichter, and Katrin Franke. Collecting cyber threat intelligence from hacker forums via a two-stage, hybrid process using support vector machines and latent dirichlet allocation. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 5008–5013, Seattle, WA, USA, 2018. IEEE.

[18] Jacob Devlin et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[19] Fangzhou Dong, Shaoxian Yuan, and Liang Liu. New cyber threat discovery from darknet marketplaces. In *2018 IEEE International Conference on Big Data and Artificial Intelligence (ICBDAI)*, pages 69–72. IEEE, 2018.

[20] Saida Elouardi, Anas Motii, Mohammed Jouhari, Abdoul Amadou, and Mustapha Hedabou. A survey on hybrid-cnn and llms for intrusion detection systems: Recent iot datasets. *IEEE Access*, PP:1–1, 2024. Published: November 26, 2024.

[21] Yong Fang, Jian Gao, Zhonglin Liu, and Cheng Huang. Detecting cyber threat event from twitter using idcnn and bilstm. *Applied Sciences*, 10(17):5922, 2020.

[22] Yong Fang, Yusong Guo, Cheng Huang, and Liang Liu. Analyzing and identifying data breaches in underground forums. *IEEE Access*, 7:48770–48777, 2019.

[23] Mateusz Fedoryszak, Brent Frederick, Vijay Rajaram, and Changtao Zhong. Real-time event detection on social data streams. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2774–2782, New York, NY, USA, 2019. Association for Computing Machinery.

[24] Nicolas Fiorini et al. Cysecbert: Cybersecurity language model. In *IEEE International Conference on Big Data*, 2023.

[25] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

[26] Thomas Geras and Thomas Schreck. The 'big beast to tackle': Practices in quality assurance for cyber threat intelligence. In *The 27th International Symposium on Research in Attacks, Intrusions and Defenses*, pages 337–352, Padua, Italy, 2024. ACM.

[27] Mohamed Hagras, Ghada Hassan, and Nadine Farag. Towards natural disasters detection from twitter using topic modelling. In *2017 European*

Conference on Electrical Engineering and Computer Science (EECS), pages 272–279. IEEE, 2017.

[28] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[29] SANS Institute. 2023 cti survey: Keeping up with a changing threat landscape, 2023.

[30] Seungjun Jin et al. Darkbert: A language model for the dark side of the internet. In *ACL*, 2023.

[31] Masashi Kadoguchi, Shota Hayashi, Masaki Hashimoto, and Akira Otsuka. Exploring the dark web for cyber threat intelligence using machine learning. In *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 200–202. IEEE, 2019.

[32] Rupinder Paul Khandpur, Taoran Ji, Steve Jan, Gang Wang, Chang-Tien Lu, and Naren Ramakrishnan. Crowdsourcing cybersecurity: Cyber attack detection using social media. https://doi.org/10.48550/arXiv.1702.07745, 2017. arXiv:1702.07745.

[33] Basak Kömeçoglu and Burcu Yilmaz. Event graph-based news clustering: The role of named entity-centered subgraphs. *IEEE Access*, PP:1–1, January 2024.

[34] Ying Li, Jiaxing Cheng, Cheng Huang, Zhouguo Chen, and Weina Niu. Nedetector: Automatically extracting cybersecurity neologisms from hacker forums. *Journal of Information Security and Applications*, 58:102784, May 2021.

[35] Yinhan Liu et al. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[36] Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. Zero-shot entity linking by reading entity descriptions. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460, Florence, Italy, 2019. Association for Computational Linguistics.

[37] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2019.

[38] Cyber Magazine. The rapidly evolving threat landscape of 2024, 2024. Accessed: 2025-02-27.

[39] Dalyapraz Manatova, Charles DeVries, and Sagar Samtani. Understand your shady neighborhood: An approach for detecting and investigating hacker communities. *Decision Support Systems*, 184:114271, September 2024.

[40] Felipe Moreno-Vera, Mateus Nogueira, Cainã Figueiredo, Daniel Sadoc Menasché, Miguel Bicudo, Ashton Woiwood, Enrico Lovat, Anton Kocheturov, and Leandro Pfleger de Aguiar. Cream skimming the underground: Identifying relevant information points from online forums. arXiv preprint arXiv:2308.02581, 2023. Accessed: 2025-03-28.

[41] Inoussa Mouiche and Sherif Saad. Entity and relation extractions for threat intelligence knowledge graphs. *Computers & Security*, 148:104120, January 2025.

[42] Tommaso Paladini, Lara Ferro, Mario Polino, Stefano Zanero, and Michele Carminati. You might have known it earlier: Analyzing the role of underground forums in threat intelligence. In *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. ACM, 2024.

[43] Tanmay Parekh, Anh Mac, Jiarui Yu, Yuxuan Dong, Syed Shahriar, Bonnie Liu, Eric Yang, et al. Event detection from social media for epidemic prediction. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5758–5783, Mexico City, Mexico, 2024. Association for Computational Linguistics.

[44] Sergio Pastrana, Daniel R. Thomas, Alice Hutchings, and Richard Clayton. CrimeBB: Enabling Cybercrime Research on Underground Forums at Scale. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, pages 1845–1854. ACM Press, 2018.

[45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[46] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[47] Md Rayhanur Rahman, Rezvan Mahdavi-Hezaveh, and Laurie Williams. What are the attackers doing now? automating cyber threat intelligence extraction from text on pace with the changing threat landscape: A survey. *ACM Computing Surveys*, 55(12):1–36, 2023.

[48] Lukáš Sadlek, Muhammad Mudassar Yamin, Pavel Čeleda, and Basel Katt. Severity-based triage of cybersecurity incidents using kill chain attack graphs. *Journal of Information Security and Applications*, 89:103956, March 2025.

[49] Sagar Samtani, Ryan Chinn, Hsinchun Chen, and Jay Nunamaker. Exploring emerging hacker assets and key hackers for proactive cyber threat intelligence. *Journal of Management Information Systems*, 34(4):1023–1053, 2017.

[50] Anna Sapienza, Alessandro Bessi, Saranya Damodaran, Paulo Shakarian, Kristina Lerman, and Emilio Ferrara. Early warnings of cyber threats in online discussions. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 667–674, New Orleans, LA, 2017. IEEE.

[51] Kailash Karthik Saravanakumar, Miguel Ballesteros, Muthu Kumar Chandrasekaran, and Kathleen McKeown. Event-driven news stream clustering using entity-aware contextual embeddings. In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2330–2340, Online, 2021. Association for Computational Linguistics.

[52] Quentin Sceller, Elmouatez Karbab, Mourad Debbabi, and Farkhund Iqbal. SONAR: Automatic Detection of Cyber Security Events over the Twitter Stream. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM)*, 2017.

[53] Hyejin Shin, WooChul Shim, Jiin Moon, Jae Seo, Sol Lee, and Yong Hwang. Cybersecurity event detection with new and re-emerging words. In *Proceedings of the 2020 ACM SIGKDD Workshop on Cybersecurity and Intelligence*. ACM, 2020.

[54] Todor Staykovski, Alberto Barrón-Cedeño, Giovanni Da San Martino, and Preslav Nakov. Dense vs. sparse representations for news stream clustering. In *Proceedings of the 2nd International Workshop on Narrative Extraction from Texts (Text2Story@ECIR)*, pages 43–48, Cologne, Germany, 2019. Workshop held at ECIR 2019.

[55] Mengmeng Tang, Yuanbo Guo, Qingchun Bai, and Han Zhang. Trigger-free cybersecurity event detection based on contrastive learning. *The Journal of Supercomputing*, 79:20984–21007, 2023.

[56] Andrea Tundis, Samuel Ruppert, and Max Mühlhäuser. A feature-driven method for automating the assessment of osint cyber threat sources. *Computers & Security*, 113:102576, 2022.

[57] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. In *Journal of Machine Learning Research*, volume 11, pages 2837–2854, 2010.

[58] Xuren Wang, Songheng He, Zihan Xiong, Xinxin Wei, Zhengwei Jiang, Sihan Chen, and Jun Jiang. Aptner: A specific dataset for ner missions in cyber threat intelligence field. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 1233–1238, Hangzhou, China, 2022. IEEE.

[59] Xuren Wang, Xinpei Liu, Shengqin Ao, Ning Li, Zhengwei Jiang, Zongyi Xu, Zihan Xiong, Xiong Mengbo, and Xiaoqing Zhang. Dnrti: A large-scale dataset for named entity recognition in threat intelligence. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1632–1639. IEEE, 2020.

[60] Semih Yagcioglu, Mehmet Saygin Seyfioglu, Begum Citamak, Batuhan Bardak, Seren Guldamlasioglu, Azmi Yuksel, and Emin Islam Tatli. Detecting cybersecurity events from noisy short text. pages 1366–1372, 2019.

[61] Samira Yousefinaghani, Rozita Dara, Zvonimir Poljak, Theresa Bernardo, and Shayan Sharif. The assessment of twitter's potential for outbreak detection: Avian influenza case study. *Scientific Reports*, 9, 2019.

[62] Yiming Zhang, Yujie Fan, Shifu Hou, Jian Liu, Yanfang Ye, and Thirimachos Bourlai. iDetector: Automate underground forum analysis based on heterogeneous information network. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1071–1078, Barcelona, 2018. IEEE.

[63] Adam Zibak, Clemens Sauerwein, and Andrew C. Simpson. Threat intelligence quality dimensions for research and practice. *Digital Threats: Research and Practice*, 3(4):1–22, 2022.