

# EUREKHA: Enhancing User Representation for Key Hackers Identification in Underground Forums

Abdoul Nasser Hassane Amadou, Anas Motii, Saida Elouardi, EL Houcine Bergou

College of Computing

University Mohammed VI Polytechnic

Benguerir, Morocco

{abdoul.amadou, anas.motii, saida.elouardi, elhoucine.bergou}@um6p.ma

**Abstract**—Underground forums serve as hubs for cybercriminal activities, offering a space for anonymity and evasion of conventional online oversight. In these hidden communities, malicious actors collaborate to exchange illicit knowledge, tools, and tactics, driving a range of cyber threats—from hacking techniques to the sale of stolen data, malware, and zero-day exploits. Identifying the key instigators (i.e., key hackers), behind these operations is essential but remains a complex challenge. This paper presents a novel method called *EUREKHA* (Enhancing User Representation for Key Hacker Identification in Underground Forums), designed to identify these key hackers by modeling each user as a textual sequence. This sequence is processed through a large language model (LLM) for domain-specific adaptation, with LLMs acting as feature extractors. These extracted features are then fed into a Graph Neural Network (GNN) to model user structural relationships, significantly improving identification accuracy. Furthermore, we employ BERTopic (Bidirectional Encoder Representations from Transformers Topic Modeling) to extract personalized topics from user-generated content, enabling multiple textual representations per user and optimizing the selection of the most representative sequence. Our study demonstrates that fine-tuned LLMs outperform state-of-the-art methods in identifying key hackers. Additionally, when combined with GNNs, our model achieves significant improvements, resulting in approximately 6% and 10% increases in accuracy and F1-score, respectively, over existing methods. *EUREKHA* was tested on the *Hack-Forums*<sup>1</sup> dataset, and we provide open-source access to our code<sup>2</sup>.

**Index Terms**—Key Hacker Identification; Underground Forums; Large Language Model; Social Network Analysis; Topic Modeling

## I. INTRODUCTION

To stay ahead of the ever-changing threat landscape, embracing proactive security measures such as Cyber Threat Intelligence (CTI) is essential. CTI involves gathering, analyzing, and utilizing data on security threats, threat actors, malware, vulnerabilities, exploits, and indicators of compromise (IoCs). An effective CTI must deliver relevant and precise information, identify credible threats, and offer actionable insights for threat mitigation. Through rigorous examination, recent research highlights the increasing importance of underground markets and communities [15], [17], delving into their socio-economic and geographical ramifications.

These underground forums frequented by hackers serve as valuable sources of CTI. A study [10] reveals that the intelligence from hacker forums is underutilized in addressing cyber threats. This reluctance can be attributed to the overwhelming volume of unprocessed data in these forums, which poses a significant processing challenge. Many of these data are irrelevant or repetitive, making the manual analysis of hacker forum data a laborious, resource-intensive, and error-prone process. We define a "key hacker" as an influential user involved in cybercriminal activities, including distributing malware, offering ready-made denial-of-service attack tools, or using these tools to launch attacks on others. Identifying key hackers in underground forums is crucial; by pinpointing them, security teams can efficiently extract valuable CTI while ensuring the authenticity of the extracted content and its sources, all within a time-efficient framework.

This research is motivated by the following factors: (1) the critical importance of such information for effective incident response and detection within organizations, (2) the substantial volume of data originating from underground forums, and (3) the need to optimize security teams resources by focusing on key hackers, whose posts may provide valuable insights, including details on malware, techniques, and zero-day vulnerabilities.

Despite the strides made by previous studies [38], [53] in automating CTI extraction through the semantic interpretation of forum data, the persistent challenge remains in validating profiles before CTI extraction. In [11], researchers developed a system that employs behavioral analysis and account recognition techniques via social networks and semantic analyses of users to identify cybercriminals. Moreover, the authors in [44] used centrality measures to categorize user types, although their reliance on outdated data highlights the need for real-time data analysis for effective CTI. In [69], the authors introduced an attributed heterogeneous information network designed to identify key hackers using a Graph Convolutional Network (GCN), achieving an accuracy rate of approximately 90%. Other research efforts have tackled similar challenges in various domains, including the dark web [2], [5], [8], organizational security [12], public exploits [16], and deceptive CTI [47].

Recent hacker identification methods using GNNs [69], [70] have demonstrated potential but are limited by the un-

<sup>1</sup><https://hackforums.net/>

<sup>2</sup><https://github.com/jumbo110/EUREKHA>

derutilization of user attributes and lengthy training times due to shallow text embedding techniques, such as skip-gram [35] and bag-of-words (BoW) [21]. Additionally, reliance on embedding techniques like Doc2Vec [30] and Word2Vec [27] struggles to capture the full complexity of users' semantic features, especially for those with extensive posting histories. This leads to biased and low-quality embeddings due to the dilution of attribute information during node feature extraction. In contrast, multi-layer LLM models provide a more sophisticated and nuanced approach to text representation. Recently, the authors in [37], evaluated the efficiency of hacker forums as a source of Cyber Threat Intelligence, using fine-tuned BERT-based models to conduct the analysis.

We introduce a novel approach, Enhancing User Representation for Key Hackers Identification in Underground Forums (*EUREKHA*), which leverages LLM for domain adaptation and feature extraction. We first convert the metadata, threads, and replies of each user into a unified textual sequence for representation. These sequences are then submitted to the LLM model, where we perform fine-tuning for domain adaptation to help the model acquire relevant domain knowledge. However, processing extensive user interaction data, such as years' worth of activity, is challenging due to the complexity and noise in lengthy user sequences. The variability in user content over time further complicates pattern recognition. To address this issue, we use BERTopic [20] to condense large posting histories into concise topics, reducing computational demands while preserving key information. By enhancing user profiles with a deeper semantic understanding, we enable GNNs to effectively combine textual attributes with nuanced user characteristics and graph topology. This integration creates a comprehensive source of information that greatly enhances key hacker identification. We summarize our contributions as follows:

- We utilize BERTopic to extract personalized topics from user-generated content, summarizing extensive user history into relevant patterns and enabling the creation of four distinct user sequence representations. We evaluate these representations to identify the most effective one for key hacker identification.
- To the best of our knowledge, we are the first to explore LLM-based methods for key hacker identification. We fine-tune several LLM models for this task and find that they outperform traditional GNN-based methods. Our added value includes extracting features with LLMs instead of using classical embedding techniques such as Word2Vec and Doc2Vec.
- Our experiments show that *EUREKHA* achieves state-of-the-art performance, outperforming existing methods by approximately 6% in accuracy and 10% in F1-score.

The remainder of the paper is organized as follows: Section II provides a review of existing literature. Section III provides a brief overview of the *EUREKHA* framework. Section IV presents the proposed methodology in more detail, including experimental datasets, user representation, topic mod-

eling of content, the fine-tuning of LLMs, and GNN training. Section V describes the experimental setup and evaluates the performance of different LLM and GNN models, comparing our results with recent work to demonstrate that we achieve the highest accuracy for key hacker identification. Section VII discusses the limitations and suggests potential directions for future research. Finally, Section VIII summarizes our contributions and concludes the paper.

## II. RELATED WORK

Table I provides a comprehensive summary of related works focused on identifying key hackers in underground forums. Several methods have been proposed, which can be divided into three main categories: content-based methods, centrality-based methods, and graph neural network-based methods.

1) **Content-based methods:** The data generated by users of underground forums is substantial and includes a variety of elements, such as threads, posts, comments, and uploaded attachments. A content-based method involves mining this data and constructing user evaluation metrics to identify key hackers. The most common evaluation metrics are activity level and content quality. In [33], the authors analyzed content features, seniority features, and social network features within the context of underground forums. An optimization meta-heuristic was employed to identify key hackers, and a systematic reputation-based method was proposed to validate the findings. Fang et al. [18] developed a framework incorporating topic models to extract popular topics, track topic evolution, and identify key hackers along with their specialties. In [67], the authors investigated the transfer of knowledge among user posts in underground forums and classified users into four categories: expert, casual, learning, and novice hackers. Expert hackers are knowledgeable and respected members of the community who increasingly serve as knowledge providers. While content-based analysis provides comprehensive metrics reflecting user influence, it is also more complex, necessitating professional involvement and verification in the selection of evaluation metrics.

2) **Centrality-based methods:** Centrality measures are commonly used to identify influential hackers based on social network structures. These studies often begin by using content filtering techniques, such as Latent Dirichlet Allocation (LDA) [26], to categorize content based on topics that align with specific skill levels and to filter out users who do not have a certain level of expertise. After filtering, social network structures are built by analyzing user interactions, followed by the application of centrality measures to identify influential hackers. In [49], the authors identify key hackers in a large English hacker forum who distribute keyloggers using only degree and betweenness centrality measures within a social network. Similarly, in [51], the authors predicted key hackers by analyzing various malicious hacker tools, relying on these same centrality measures. Recently, Huang et al. [24] introduced HackerRank, a tool designed to analyze and identify key hackers in underground forums. The tool uses LDA to filter out users with low hacking skills and then uses PageRank to

highlight highly skilled hackers. In [39], the authors introduce an AI-driven framework (INSPECT) to identify key hackers in underground forums based on five centrality measures (degree, betweenness, closeness, eigenvector, and PageRank).

However, these approaches have significant limitations. Content filtering methods, such as LDA, can oversimplify user categorizations and miss nuanced skill indicators. Traditional LDA models may fail to capture these contextual differences. Additionally, while centrality-based methods focus on network structure, they overlook important user characteristics, such as trust, reputation, and prestige, which are crucial in underground forums. To overcome these challenges, emerging trends such as Graph Neural Networks (GNNs) [71] offer promising solutions.

3) **GNN-based methods:** Graph Neural Networks (GNNs) offer a robust alternative for detecting influential hackers, surpassing traditional centrality-based methods. GNNs excel at capturing complex, nonlinear relationships in the data. They can integrate user features, including behavioral patterns, interaction histories, and contextual attributes such as trust, reputation, and prestige. In [64], the authors used a heterogeneous graph attention network (HGHAN) [60] to develop a system for identifying hacker groups. In [70], the authors use heterogeneous graph representation and meta-path [25] approaches to establish user relationships on *Hack-Forums*. They introduced ActorHin2vec to learn low-dimensional user representations within the Heterogeneous Information Network (HIN) framework and identify key hackers. However, this approach does not consider crucial user attributes such as profiles, post content, and threads, which are essential for accurately representing users. To address this limitation, the authors of [69] propose an Attributed Heterogeneous Information Network (AHIN) as a solution. This method integrates meta-paths to assess user relationships and introduces Player2Vec for efficient node representation in identifying key hackers. By incorporating user attributes into their model, they improve the accuracy and performance of user characterization in complex networks compared to ActorHin2vec.

Despite the performance of GNNs in learning user representations in underground forums, they are limited by shallow text embeddings that fail to capture complex user attributes, especially for users with extensive posting histories, leading to biased and low-quality embeddings due to the dilution of attribute information during node feature extraction. Our approach uses LLMs for feature extraction, generating enriched user representations as features for GNNs. This hybrid method enhances user representations, significantly improving key hacker identification in underground forums. However, processing extensive user interaction data, such as years' worth of activity, is challenging due to the inherent complexity and noise in lengthy user sequences. To address this challenge, we employ BERTopic [20] to extract personalized topics from user-generated content, thereby reducing data complexity and enhancing LLM training for a deeper understanding of user profiles in underground forums.

### III. SYSTEM OVERVIEW

In this section, we provide the system overview of *EU-REKHA* depicted in Fig. 1. The approach consists of the following steps:

**Dataset and Preprocessor:** We used CrimeBB [41], one of the most well-known datasets in underground forums. This dataset has been utilized in various works [39], [8], [45], [62]. We focused our study on *Hack-Forums*, and due to the presence of noise, we preprocessed the users' posted content to ensure it was thoroughly cleaned. In addition, the dataset is labeled to establish a reliable ground truth for our analysis. The annotation process is detailed in Section V-A2.

**User Sequence Representation:** The user representation consists of two steps :

**Topics Modeling:** Using pre-processed content from the previous module, we apply BERTopic [20] to generate topic distributions for each user's threads and replies, resulting in thread topics and reply topics.

**User as a Textual Sequence:** Based on the generated topics from threads and replies, we now have metadata, thread topics, threads, replies, and reply topics for each user account. We evaluated various strategies for user sequence representation, exploring methods to manage long text sequences and assessing the effectiveness of these representations in enhancing model efficiency. The user sequence representations are detailed in Section IV-B2 and their assessment is detailed in Section V-C.

**LLM Fine-tuning and Feature Extraction:** To improve model stability and performance, we first perform domain adaptation. The user's text sequence is processed through the LLM, which acts as a feature extractor to produce user embeddings. These embeddings are then passed to the GNN model.

**GNN Training on Enriched Features:** The GNN utilizes user embeddings generated by LLM as input, along with a graph structure that represents various types of interactions, to effectively learn user representations. The interactions include: quoted replies, threads, and contract relationships. Based on these learned representations, the final user representations obtained from the GNN are then used to predict whether a user is a key hacker.

### IV. METHODOLOGY

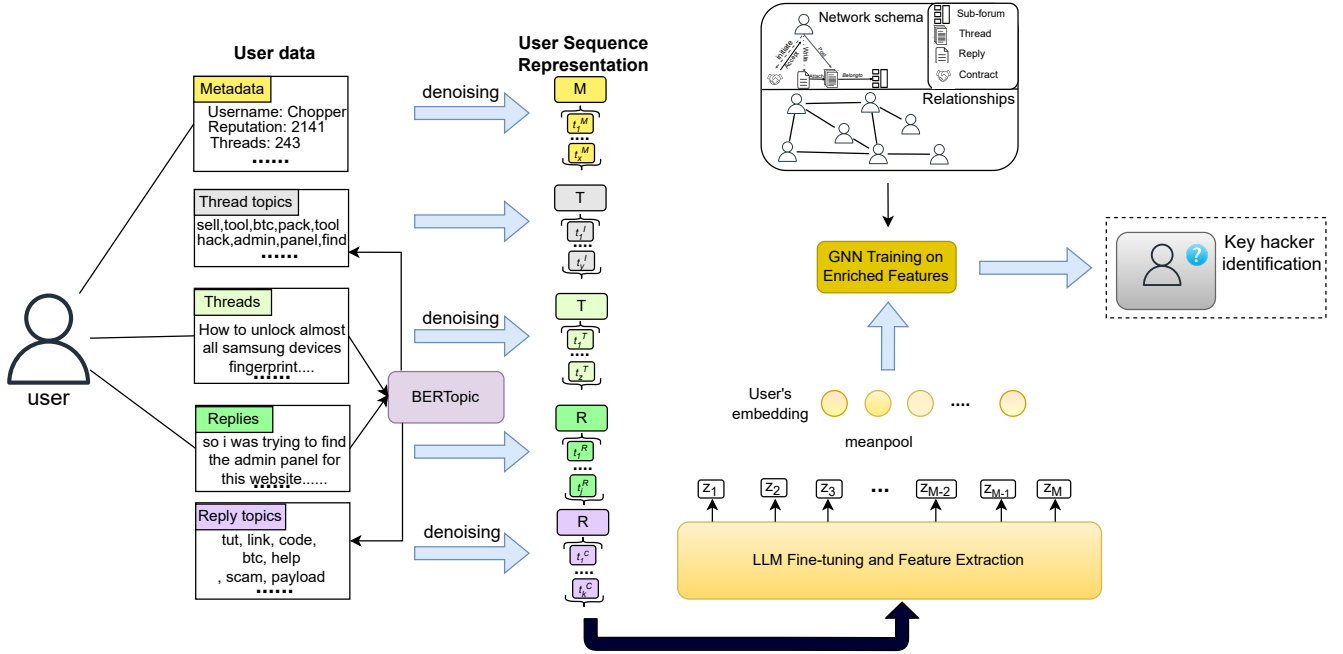
In this section, we provide details about our approach and explain how users are represented in underground forums. In addition, LLM fine-tuning and GNN training are discussed.

#### A. Dataset and Preprocessor

We use the latest version of the CrimeBB dataset collected on 21/06/2023. It contains data from several underground forums, with a focus on *Hack-Forums*, the largest and longest-running hacking and cybercrime forum. This forum has more than 42.5 million posts and 4 million threads created by 736,000 user accounts spanning over 15 years. *Hack-Forums* is structured into nine main categories: Hacking, Coding, Gaming, Technology, Market, Money, Graphics, and Common

TABLE I: Existing works on identifying key hackers in underground forums.

Reference	Methods			Summary	Limitations
	Content	Centrality	GNN		
[33]	✓			They used a meta-heuristic to analyze content and seniority to identify key hackers.	While content-based analysis offers comprehensive metrics that reflect user influence, it is more complex and requires professional involvement and verification in selecting evaluation metrics.
[18]	✓			They developed a framework with topic models to identify key hackers and their specialties.	
[67]	✓			The authors examined knowledge transfer in user posts and categorized users.	
[49]		✓		The authors used degree and betweenness centrality to identify key hackers distributing keyloggers.	Centrality-based methods emphasize network structure but fail to account for essential user characteristics like trust, reputation, and prestige, which are vital in underground forums.
[24]		✓		They introduced HackerRank, a tool that uses an improved PageRank algorithm to identify key hackers.	
[39]		✓		The authors introduced INSPECT, an AI-driven framework that identifies key hackers using five centrality measures.	
[64]			✓	The authors used HGHAN to develop a system for to identify hacker groups.	Due to the fusion strategy HGHAN dilutes node features.
[70]			✓	The authors introduced ActorHin2vec identify key hackers.	This approach overlooks important user attributes.
[69]			✓	The authors propose an AHIN, improving user characterization in identifying key hackers.	This approach fails to capture complex user attributes.


 Fig. 1: Overview of our proposed framework *EUREKHA*.

(which includes subforums for discussions on various topics, such as entertainment or politics, as well as forums for rules and suggestions). In [40], [68], the authors reported that cybercrime subforums are organized into distinct categories, such as 'Hacking' which covers techniques and tools for unauthorized access, and 'Market' where illegal goods and services are traded [50]. Therefore, we focus our study on these two subforums. Table II provides key metrics for the Hacking and Market categories within the *Hack-Forums* platform. The preprocessing step removes special characters and

extra spaces, further cleaning the user content. It also replaces certain textual elements such as URLs, cited posts, quotations, and source code with the labels URL, CITE, QUOTE, and CODE. This enhances clarity and makes the analysis easier.

### B. User Sequence Representation

The user representation process involves two steps: first, managing large volumes of user content, and second, representing the user as a textual sequence in multiple ways, as detailed hereunder.

TABLE II: Metrics for Hacking and Market categories in *Hack-Forums*.

Metric	Value
Number of Subforums	35
Number of Users	465,385
Number of Posts	12,630,580
Number of Threads	1,555,354
Number of Contracts	348,340
Average Posts per User	27.14
Average Threads per User	3.34
Replies per Thread	7.12
Average Length of Posts (words)	29.13
Average Length of Threads (words)	5.19
Language	EN

1) *Topics Modeling*: Topic modeling reveals the thematic structure within a text corpus by categorizing documents into distinct topics based on related words. This technique is crucial for understanding context, community types, user expertise, and content in forums [61], [23], [57]. Various text analysis methods, such as Latent Semantic Indexing (LSI) [48], Probabilistic Latent Semantic Analysis (PLSA) [6], Latent Dirichlet Allocation (LDA) [26], Non-Negative Matrix Factorization [31], Top2Vec [3], and BERTopic [20], are suited to different text analysis scenarios.

Unlike traditional methods that rely on word frequency and often struggle to capture complex semantic relationships and contextual nuances, BERTopic leverages advanced language understanding of BERT and c-TF-IDF [46] for more coherent topics. We applied BERTopic [20] to extract personalized topics from users' threads and replies, generating two distinct sets of topics for each user: thread topics and reply topics. For the first, by analyzing users' thread posts to extract recurring themes and central ideas. These thread topics represent the key subjects the user tends to initiate discussions on. For the latter, We applied the same process to users' replies. We identified distinct reply topics that reflect how users engage in discussions started by others.

BERTopic automatically determines the optimal number of topics, eliminating the need for manual specification. Fig. 2 shows the workflow of BERTopic and the details of each step are as follows:

- **Document embedding**. Each document is transformed into a vector using BERT, which captures contextual nuances and provides richer, more meaningful representations.
- **Dimensionality reduction**. BERT embeddings are high-dimensional (often 768 dimensions). BERTopic uses UMAP [34] to reduce these dimensions while preserving key structural aspects, which prepares the data for effective clustering.
- **Clustering**. With reduced dimensions, BERTopic uses Hierarchical Density-Based Spatial Clustering of Ap-

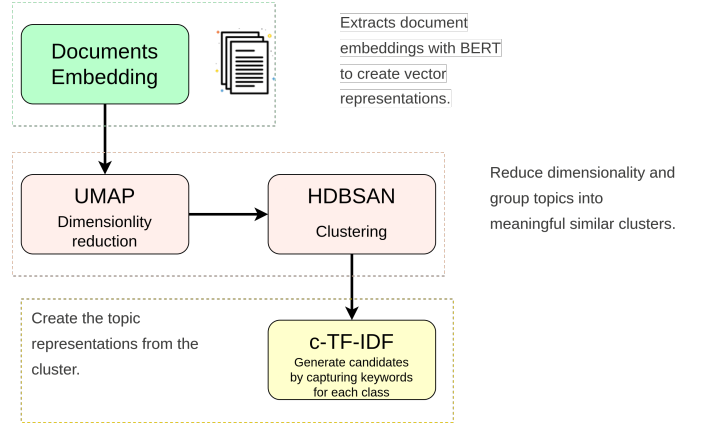


Fig. 2: The workflow for modeling topics using BERTopic.

plications with Noise (HDBSCAN) [54] for clustering, which identifies clusters of varying densities without requiring a preset number of clusters.

- **Topic representation**. After clustering, BERTopic [20] uses c-TF-IDF to extract the most representative words for each topic, highlighting words that are frequent within a topic but rare across others, capturing each unique cluster.

$W_{x,c}$  represents the weight assigned to the term  $x$  in the context of the cluster  $c$ .

$$W_{x,c} = \|tf_{x,c}\| \times \log\left(1 + \frac{A}{f_x}\right) \quad (1)$$

Where  $tf_{x,c}$  represents the frequency of word  $x$  within cluster  $c$ ,  $f_x$  indicates the frequency of word  $x$  across all clusters, and  $A$  is the average number of words per cluster.

2) *User as a Textual Sequence*: Based on generated topics of threads and replies for each user, we now have for each account: metadata, threads, thread topics, replies and reply topics. Inspired by [9], we created a unified representation by encoding user data as structured text sequences compatible with LLM inputs. Specifically, for users' metadata, *EUREKHA* extracts the corresponding information: username, thread count, post count, and reputation and organizes it in the following format:

[M]{username}{SEP}{thread\_count}{SEP}{post\_count} ...,

Where [M] is a special token [65] marking the beginning of the metadata section, while [SEP] separates the metadata attributes. The same encoding procedure is applied for thread topics, threads, replies, and reply topics. Fig. 3 shows an example of a user sequence representation.

To determine the optimal format for capturing a complete user profile, we explore different strategies to represent user sequences as detailed in Table III:

- **Full Representation (R1)**: Combines all user metadata, full threads, and full replies into one textual sequence representing the user, without any topic summarization.

TABLE III: User Sequence Representation formats.

Representation	Format
R1	[M] {metadata} [T] {thread} [R] {reply}
R2	[M] {metadata} [T] {thread topic} [R] {reply}
R3	[M] {metadata} [T] {thread} [R] {reply topic}
R4	[M] {metadata} [T] {thread topic} [R] {reply topic}

- Reduced Threads (R2): Replaces full threads with their summarized thread topics, while keeping metadata and full replies unchanged.
- Reduced Replies (R3): Replaces full replies with their summarized reply topics, retaining metadata and full threads.
- Reduced Threads and Replies (R4): Replaces both threads and replies with their corresponding topics, combining with metadata for a topic-centered representation.

Where [M], [T], and [R] are special tokens [65] added to the LLM's tokenizer.

### C. LLM Fine-tuning and Feature Extraction

We perform domain-adaptive fine-tuning of the LLM to improve the representation of users and increase the performance of our GNN model using user sequences and their corresponding labels. We use LLM to encode the textual sequence of the user, which can be formulated as follows:

$$z_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \text{LLM}(\mathbf{t}_i)_j, \quad (2)$$

Where  $\mathbf{t}_i$  is the textual sequence of the  $i$ -th user,  $\text{LLM}(\cdot)$  denotes the large language model adopted as feature extractor, and  $N_i$  is the number of tokens in  $\mathbf{t}_i$ . We use mean-pooling on the LLM output to generate 768-dimensional embeddings for representing users. To make predictions, we apply an  $L$ -layer Multilayer perceptron (MLP) [56] to reduce the dimension of  $z_i$ , project it into a binary classification space, and obtain the predicted logit, which is formulated as:

$$z_i^{(l)} = \text{LeakyReLU} \left( \mathbf{W}^{(l)} \cdot z_i^{(l-1)} + \mathbf{b}^{(l)} \right), \quad (3)$$

Where  $z_i^{(0)}$  is the output of the LLM in Equation (2). We then apply SoftMax to the output logit  $z_i^{(L)}$  to obtain the prediction.

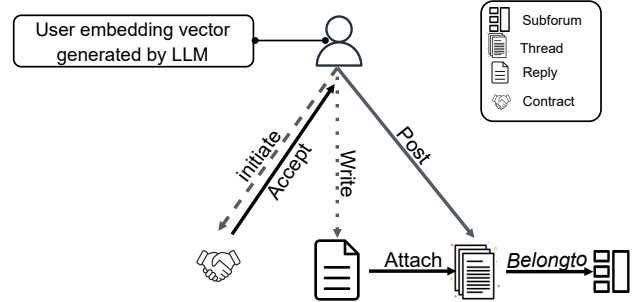
### D. GNN Training on Enriched Features

For deep user representation in the underground forum, the embeddings generated by the pretrained LLM are refined by a GNN layer to model complex interactions. *EUREKHA* uses GNNs to encode graph knowledge, which is expressed as:

$$\begin{aligned} \mathbf{a}_u^{(l)} &= \text{AGGREGATE}_{v \in \mathcal{N}(u)} \left[ \text{MSG}^{(l)} \left( \mathbf{h}_u^{(l-1)}, \mathbf{h}_v^{(l-1)} \right) \right] \\ \mathbf{h}_u^{(l)} &= \text{UPDATE} \left( \mathbf{h}_u^{(l-1)}, \mathbf{a}_u^{(l)} \right) \end{aligned} \quad (4)$$

**METADATA** Sat Oct 02 183401 2010 </s> [REDACTED] </s> 128 </s> 1276 </s> 404 </s> 793 **THREAD:** Encryption Decryption Tutorial </s> How To Hack An Email </s> Techniques and vulnerabilities in email hacking </s> Tools used for exploiting email systems </s> Best MD5 Cracker </s> Discussion on the most effective tools for cracking MD5 hashes </s> Pros and cons of various methods </s> Net Bios </s> Overview of NetBIOS vulnerabilities and exploitation techniques </s> How was HTML Born </s> History and evolution of HTML as a markup language </s> Hacking With Telnet </s> Using Telnet for network hacking and exploitation </s> Remote Penetration v20 Beta2 Public </s> Release discussion and features of the new penetration testing tool ..... **REPLY:** Thanks for the information where did you find this </s> If anyone has a working RAT Remote Access Tool please share </s> CMD Shell utilities like Zenmap Nmap are useful for network scanning Download here URL </s> Thank you for this tutorial </s> Batch files make tasks in CMD easier to automate </s> If you know the owner of the FTP server can you figure out the password .....

Fig. 3: User sequence representation example.

Fig. 4: Network schema of *Hack-Forums*.

Where  $\text{MSG}(\cdot, \cdot, \cdot)$  represents the process of passing the message from node  $u$  to its neighbor  $v$ . The  $\text{AGGREGATE}(\cdot)$  function collects and combines messages from all adjacent nodes of  $u$ , resulting in the aggregated message  $\mathbf{a}_u^{(l)}$ . The  $\text{UPDATE}(\cdot, \cdot)$  function then refines the node representation  $\mathbf{h}_u^{(l-1)}$  using  $\mathbf{a}_u^{(l)}$ , producing  $\mathbf{h}_u^{(l)}$ .

After processing through  $L$  GNN layers, we derive the final node representations and utilize a linear transformation to compute the prediction logits for each node:

$$\mathbf{h}_i^o = \mathbf{W}_o \cdot \text{LeakyReLU} \left( \mathbf{h}_i^{(L)} \right) + \mathbf{b}_o \quad (5)$$

Fig. 4 presents the user network schema, which allows a user to be expressively represented. Three types of relationships are captured:

- **Quoted reply relationships:** This represents direct communication where one user quotes another's reply. For instance, a `user1-quoted_reply-user2` relationship indicates user1 quoted a reply from user2.
- **Thread relationships:** A `user1-thread-user2` relationship means user1 started a thread, and user2 later replied.
- **Contract relationships:** This denotes formal agreements between users. A `user-contract-user2` relationship indicates that user1 initiated a contract with user2, potentially involving illicit transactions.

## V. EXPERIMENTS

In this section, we evaluate *EUREKHA* using the *Hack-Forums* dataset obtained from CrimeBB. We first outline the

dataset’s annotation process, followed by an assessment of the user sequence representations discussed earlier. Next, we describe the fine-tuning process and evaluate various LLM and GNN models.

#### A. Experiment Settings

1) *Datasets*: To evaluate our system, we use *Hack-Forums*, one of the largest online underground forums. Our study focuses on two active categories, “Market” and “Hacking” which cover various cybercrime topics such as Remote Access Trojans (RATs), keyloggers, web hacking, hacking tools, malware tools, security breaches, and monetization. We analyzed data collected from September 2007 to June 2023. Table II summarizes the data for these categories.

2) *Annotations*: To establish the ground truth for our study, we select a portion of users (i.e., 5,500 users) and perform a two-step labeling process to identify key hackers within *Hack-Forums*.

**Step 1: Automated Identification**: Based on the findings of [40], we used an automated approach to identify potential key hackers by analyzing metrics such as user activity, interests, reputation, and keywords in their threads and replies. This helped us generate a preliminary list of potential candidates by detecting patterns and anomalies associated with known key hackers.

**Public Records Analysis**: Using the Google search engine, we searched for *Hack-Forums* users arrested for cybercrimes with terms like “cybercrime” “Hack-Forums” “arrested” and “hackers”. We identified some matching usernames but later discovered that some users may have changed their names or were not involved in cybercrime.

**Behavioral Analysis**: We identify users by features such as activity levels (at least 400 replies and 20 threads), engagement in selling and monetizing (examined through Market subforums), and reputation (greater than 100). Reputation reflects the peer evaluations of users.

**Keyword Analysis**: We searched for threads advertising the top 300 Remote Access Trojans (RATs) from [58] and included keywords such as: ‘bypass’, ‘hacking’, ‘hacker’, ‘hack’, ‘shell’, ‘bomber’, ‘virus’, ‘bot’, ‘botnet’, ‘DDoS’, and ‘crypter’. Users who frequently use these terms are more likely to be involved in illicit activities.

Based on these criteria, we have identified 1,600 potential key hackers.

**Step 2: Manual Verification**: To further refine the automated method findings, we conducted a detailed two-month manual review of potential key hackers based on their posted content. This process resulted in the final classification of 794 users as key hackers, while the remaining 4,706 users were classified as non-key hackers.

3) *Evaluation Metrics*: We use accuracy and F1-score as evaluation metrics. **Accuracy** measures the overall correctness of the predictions, assessing how well the model classifies both key hackers and non-hackers. It calculates the proportion of correct predictions, including true positives (TP) and true negatives (TN), relative to the total number of cases evaluated.

**F1-score**, on the other hand, emphasizes the model’s performance in correctly predicting key hackers. It balances precision (the proportion of correctly identified key hackers out of all users flagged as hackers) and recall (the proportion of actual key hackers correctly identified). This metric is crucial for evaluating the effectiveness of *EUREKHA* in accurately identifying key hackers while managing false positives and negatives.

4) *Configurations*: We conducted our experiments using a Scholar High-Performance Computing system. The resources used in the experiments are detailed as follows: The server was running Linux (RedHat version) with 16 GB of RAM and a Tesla VOLTA 100 GPU. We use Python version 3.9.15.

#### B. Baselines

To evaluate the performance of *EUREKHA*, we compare it against several state-of-the-art baselines. The **DA** method [1] characterizes users through content-based and structural features and employs the X-means algorithm to identify expert hackers and their areas of expertise. The **RRI** approach [4] measures user radicalness and associations, ranking users by influence using a customized PageRank algorithm. **KADetector** [70] utilizes a heterogeneous information network (HIN) to identify key hackers based on relationships among users, posts, and replies, without relying on node attributes. Finally, **iDetective** [69] employs an attributed heterogeneous information network (AHIN) and meta-paths to model user interactions, enhancing the identification of key hackers.

#### C. Evaluation of User Sequence Representation

After generating user sequence representations, we noticed significant variations in sequence length, leading us to address the challenge of managing long sequences. LLMs such as BERT are limited to 512 tokens, and 100% of the sequences in R1 and R2 exceeded this limit. In R3, 25% of the sequences still surpassed the limit. With R4, where BERTopic is applied to both replies and threads, the sequences fit within the limited number of tokens. To manage these long user sequences, we adopt the common strategies outlined in [55]:

**Truncation methods**: Each segment of the user sequence (metadata, threads, and replies) is assigned a specific token limit: metadata (34 tokens), threads (239 tokens), and replies (239 tokens). Allocating 26 tokens to user metadata covers all essential details about the user. The remaining 484 tokens are evenly split between user threads and replies. This approach ensures a balanced representation of the user’s content within the token limit.

**Hierarchical methods**: The lengthy user sequence representations are divided into segments of fewer than 512 tokens to comply with the LLM’s token limit. Each segment is then input into the LLM to obtain its representation. The representation of each segment corresponds to the hidden state of the [CLS] token from the last layer. We then combine the representations of all segments using hierarchical mean pooling (Hierar. Mean), max pooling (Hierar. Max), and self-attention (Hier. Self-a.) techniques.



TABLE IV: Performance evaluation of various user sequence representations (User Rep.) using BERT models with different techniques for handling long sequences.

User Rep.	F1-Score			
	Trunca.	Hier. Mean	Hier. Max	Hier. Self
R1	74.78	61.12	65.78	73.56
R2	67.89	<u>53.34</u>	57.78	62.45
R3	<b>84.39</b>	69.56	74.34	76.89
R4	72.45	61.78	67.23	70.78

Table IV shows the performance of different user sequence representations based on the F1-score. The best result is in bold, and the worst one is underlined. User sequence R3 achieved the highest F1-score across all handling methods, indicating its effectiveness for representation and classification. In contrast, sequence R2 consistently recorded the lowest F1-score, especially with the Hierarchical Mean method. This suggests that R2 may contain more noise or fewer distinguishing features, making it difficult for representation models to capture meaningful patterns, which highlights the importance of preserving the thread’s integrity.

The “Truncation” method is more effective than the “Hierarchical Self-Attention” method, though the latter still yields competitive results. Conversely, the “Hierarchical Mean” method performs the weakest, suggesting that averaging may dilute important information.

R1 relies on full content without BERTopic likely which leads to the inclusion of excessive, less relevant information, which affects its performance, particularly with more complex handling methods such as Hierarchical Mean.

With the R3 user sequence representation, the truncation method achieves the best performance. Therefore, we used this combination in the following experiments.

#### D. Effect of LLM fine-tuning

We conducted a study to assess the impact of fine-tuning LLM, with the primary goal of determining whether fine-tuning is crucial for achieving optimal performance.

Table V shows that fine-tuning led to achieving the highest accuracy and F1-score, emphasizing the importance of fine-tuning LLMs for better performance. We enhance this performance further with hyperparameter tuning as described in the following subsection.

#### E. LLMs Hyperparameter Tuning

In [13], it has been observed that when models are trained on smaller datasets, they become more sensitive to hyperparameter choices, which increases the risk of overfitting. To address this issue, we conducted a grid search over a set of pre-defined hyperparameters recommended by the BERT authors, including **batch size** [16, 24], **learning rate** [1e-5, 5e-5], and **epochs** [1, 5], to find the best settings. Each experiment was conducted five times with shuffled data, and the results were averaged for consistency. We evaluated four

TABLE V: Impact of Fine-Tuning of LLM using user sequence representation R3.

Strategy	Model	Accuracy	F1-score
Without Fine-Tuning (BERT)	GAT	86.45	38.68
	GATv2	86.18	38.21
	GCN	84.81	35.01
	RGCN	86.09	40.92
Fine-Tuning (BERT-random hyperparameter)	GAT	88.36	53.28
	GATv2	91.81	70.58
	GCN	87.90	49.43
	RGCN	92.09	71.66
Fine-Tuning (BERT-tuned hyperparameter)	GAT	95.81	82.45
	GATv2	95.73	86.11
	GCN	90.18	58.14
	RGCN	95.54	87.07

optimizers: Adam, AdamW, Adadelta, and RAdam, finding that AdamW yielded the best performance. The configuration parameters included a weight decay of 0.01, a learning rate warmup of 0.6, and a linear decay of 0.1, alongside a two-layer LLM classifier. Additionally, we applied a dropout rate of 0.1 and used leaky ReLU activation. We fine-tuned BERT [29], RoBERTa [32], ALBERT [22], and XLNet [66] using the R3 user sequence representation, which was divided into training (60%), validation (20%), and testing (20%) sets.

Fig. 5 illustrates the optimal hyperparameter combinations for each LLM model, evaluated based on the F1-score, which was prioritized due to the imbalanced nature of the dataset. Moreover, Table V illustrates that LLM hyperparameter tuning achieved the best performance results, outperforming the current state-of-the-art methods, as shown in Fig. 6.

Furthermore, Table VI presents a detailed comparison of these LLMs. Fine-tuned LLMs consistently outperform state-of-the-art GNN-based methods. BERT, in particular, leads the group with an accuracy of 95%

XLNet required the longest training time (170 minutes), significantly more than RoBERTa and BERT, completed in 29–30 minutes. This extended training time is primarily due to the number of epochs needed for XLNet to converge effectively. XLNet, despite its advanced capabilities and strong performance in text-rich tasks, relies on a complex permutation-based architecture.

The strong performance of these models in text-heavy tasks, without relying on graph structures, underscores their efficiency in identifying key hackers in underground forums.

#### F. Combination of Different LLMs and GNNs

LLMs have varying levels of pre-existing knowledge and inductive biases. Moreover, diverse GNNs can be applied with distinct focus areas. Hence, we study the effects of combining different GNNs with LLMs to evaluate the robustness of our framework. Specifically, we explored combinations of four LLMs and four GNNs, resulting in 16 possible configurations. The LLMs selected were BERT, RoBERTa, ALBERT, and



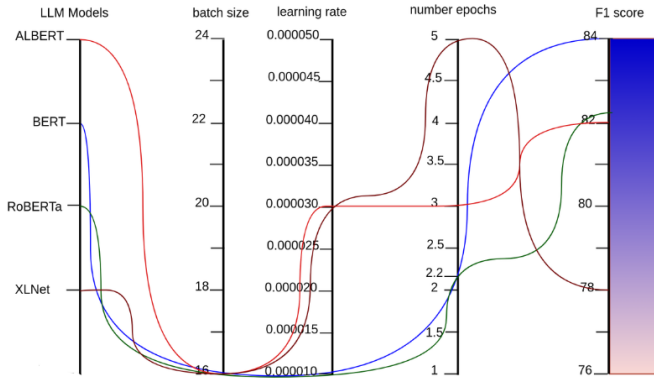


Fig. 5: Best hyperparameter configurations for ALBERT, BERT, RoBERTa, and XLNet on the R3 user sequence representation.

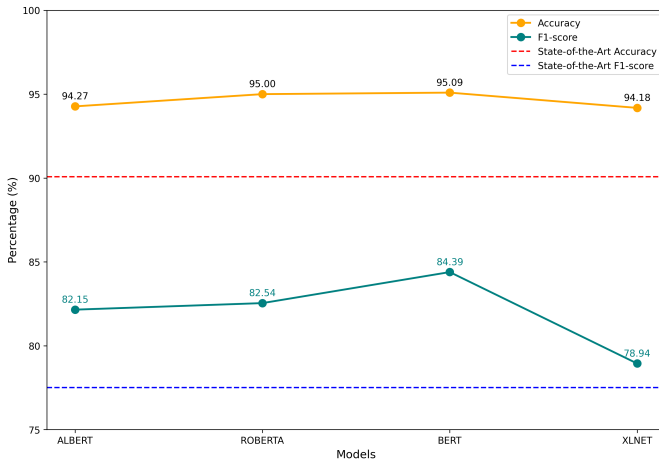


Fig. 6: Performance of fine-tuned LLMs.

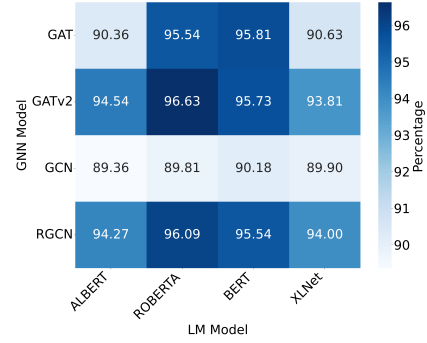
XLNet while the chosen GNNs were GCN [28], RGCN [52], GAT [59], and GATv2 [7]. The experimental results on the R3 representation are shown in Fig. 7.

The differences in the GNN architectures influence significantly the performance of *EUREKHA*. GATv2 and RGCN are the most effective GNNs, with the RoBERTa+GATv2 and BERT+RGCN combinations achieving the highest accuracy (96.63) and F1-score (87.07), respectively. This is likely due to their advanced attention mechanisms and relational reasoning capabilities. In contrast, the GCN underperforms, largely due to its lack of attention to node-specific relationships.

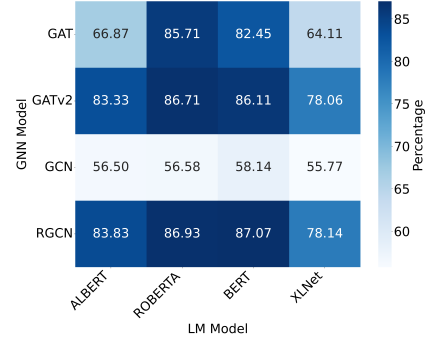
The combination of a strong LLM (e.g., RoBERTa, BERT) with a powerful GNN (e.g., GATv2, RGCN) consistently delivers the best results. This highlights the importance of selecting the right combination of the GNN and LLM to achieve optimal performance in tasks involving graph-structured data.

The choice of LLMs also plays a crucial role in overall performance. RoBERTa and BERT have emerged as the most adaptable and effective models across different GNN architectures. The most obvious improvement can be seen in GATv2, which achieves an F1-score of 87.07 when combined with

BERT, compared to only 78.06 when it is combined with XLNet.



(a) Accuracy



(b) F1-score

Fig. 7: Performance comparison under different combinations of LLMs and GNNs

### G. Comparisons with Alternative Approaches

We evaluate the proposed method, *EUREKHA*, against the baselines outlined in Section V-B, employing the same number of users to ensure a consistent evaluation. The experimental results are presented in Table VII.

*EUREKHA* consistently outperforms all baseline methods. By using advanced LLMs like BERT and RoBERTa, *EUREKHA* effectively captures detailed semantic information and combined with the GNNs such as GATv2, which model complex user relationships the performance improves further. *EUREKHA* (RoBERTa+GATv2) achieves the highest accuracy (0.966), while *EUREKHA* (BERT+GATv2) achieves the best F1-score (0.870). The strength of *EUREKHA* lies in the ability of LLMs to provide GNNs with enriched features, fully utilizing user-attributed features to construct higher-level semantic and structural connections between users. This deeper representation enables *EUREKHA* superior performance.

The approach iDetective [69], which also leverages GNNs, delivers strong results by surpassing KADetector by leveraging user-attributed features. Our approach outperforms iDetective due to its reliance on shallow text embeddings, which limits its performance.

In terms of time efficiency, our approach demonstrates superior performance, completing tasks in 29 to 32 minutes,

TABLE VI: Detailed Performance Comparison of LLM Models.

Models	Training			Validation		Test		Parameters	Memory	Total time(min.)
	Accuracy	F1-score	Loss	Accuracy	F1-score	Accuracy	F1-score			
ALBERT	94.27	82.15	0.04263	96.00	86.66	94.27	82.15	11783682	11800MiB	41
BERT	95.09	84.39	0.05083	97.09	87.00	95.09	84.39	109588098	14718MiB	30
RoBERTa	95.00	82.54	0.14916	96.90	86.59	95.00	82.54	124750722	14954MiB	29
XLNet	94.18	78.94	0.05456	95.36	81.85	94.18	78.94	116823426	12680MiB	170

TABLE VII: Average accuracy and F1-score over five runs on the R3 user sequence representation of *EUREKHA*, compared to other methods. The best results are highlighted in bold.

Method	Accuracy	F1-score	Time (min.)
RRI [4]	0.722	0.479	-
DA [1]	0.772	0.545	-
KADetector [70]	0.884	0.729	-
iDetective [69]	0.907	0.775	50
RoBERTa-finetune	0.950	0.825	29
BERT-finetune	0.950	0.843	30
<i>EUREKHA</i> (RoBERTa+GATv2)	0.966	<b>0.867</b>	31
<i>EUREKHA</i> (BERT+RGCN)	<b>0.955</b>	0.870	32

while iDetective takes 50 minutes. In [4], [1], and [70], the authors did not assess time evaluations. Despite the increased complexity of GNNs, our models maintain reasonable runtimes of 31 minutes for RGCN and 32 minutes for GATv2.

#### H. Implementation Details

We implemented our proposed *EUREKHA* using PyTorch [42], scikit-learn [43], PyTorch Geometric [19], and the Transformers [63] library. To ensure reproducibility, we configured the hyperparameters for the GNN in *EUREKHA* as follows: the AdamW optimizer, a learning rate of  $5 \times 10^{-4}$ , a dropout rate of 0.4, a total of 2 layers, a hidden size of 128, and 200 fine-tuning epochs. The code is publicly available at: <https://github.com/jumbo110/EUREKHA>.

#### I. Case Studies

For illustration purposes, we present four key hackers on *Hack-Forums*. We retrieved and analyzed all threads created by these users, uncovering several findings: (1) Many users initially focused on building their reputation by providing free tutorials and tools, which they later leveraged for larger transactions. (2) As shown in Table VIII, most identified hackers specialize in specific products or services, indicating potential value for CTIs and security firms in monitoring them. The term 'rat' (Remote Access Trojan) frequently appears, highlighting key hackers' strong association with RAT coding. Other offensive tool-related terms include 'bot', 'booter', 'crypter', and 'fud' (fully undetectable). Commerce-related terms such as 'paypal', 'btc' (Bitcoin), 'free', and 'cheap' are also common. Additionally, high occurrences of 'help', 'need', and 'question' suggest users frequently seek assistance.

Studying these key hackers identified by *EUREKHA* shows that mining data from underground forums can improve our understanding of the cybercrime ecosystem, helping to develop effective interventions.

#### VI. ETHICAL CONSIDERATIONS

Ethical considerations are central to our research. We have signed a Non-Disclosure Agreement (NDA) with Cambridge University to access the CrimeBB dataset, which consists of publicly available content. Our use of the dataset is strictly limited to analyzing collective behaviors, without targeting any individual users. To minimize risks, we have taken additional precautions, such as omitting specific user details and the exact structure of the dataset.

#### VII. LIMITATIONS AND FUTURE WORK

While *EUREKHA* has significantly advanced the identification of influential hackers within *Hack-Forums*, several limitations persist in our framework. Firstly, our approach is currently limited to a single forum. Future research will expand this scope by analyzing hacker activity across multiple forums. Secondly, the ground truth used to identify key hackers is subjective and may be influenced by annotator bias. To address this, future work will explore content similarity [14] to analyze a wider set of heterogeneous characteristics, including shared textual content, social connections, and temporal patterns. According to [40], these key hackers exhibit similar features that are crucial for their activities, such as profiles, interests, and social interactions. They frequently engage in discussions focused on hacking-related topics, emphasizing their shared context and expertise [36]. It could also be interesting to explore other LLM models like T5 and GPT.

#### VIII. CONCLUSION

In this paper, we propose *EUREKHA*, an advanced system designed for key hacker identification. Each user is initially represented as a textual sequence. We employ BERTopic to condense extensive posting histories into coherent topics and generate multiple representations of user sequences. LLM processes these representations, performing domain adaptation and feature extraction to select the best one. The output of the LLM is then used by the GNN as features and further establishes relationships among users, enhancing performance. Our study demonstrates that fine-tuned LLMs outperform state-of-the-art methods in identifying key hackers, with further improvements when combined with GNNs. Evaluated on the *Hack-Forums* dataset, *EUREKHA* achieves around 6% and

TABLE VIII: Overview of 4 Key Hackers on *Hack-Forums*

User ID	Activity	Threads	Posts	Popularity
63671	Provides hacking tutorials, technical discussions and first member of Ub3r	728	10,000+	1,000+
71764	Software cracking, private data transactions (142)	44	7,264	793
1462	Cryptography, Encryption, Decryption discussions (BTC, ETH, LTC)	265	6,319	640
63398	Program cracking, trades activation keys, release illegal keys	692	8,907	1,246

**Note:** Ub3r: A high-ranking member on *Hack-Forums*.

10% increases in accuracy and F1-score, respectively over state-of-the-art approaches, underscoring its effectiveness in key hacker identification in online underground forums.

## IX. ACKNOWLEDGEMENTS

The authors would like to express their gratitude to the Cambridge Cyber Crime Center, UK, for providing access to the CrimeBB dataset, which was used to conduct this research.

## REFERENCES

- [1] Ahmed Abbasi, Weifeng Li, Victor Benjamin, and Shiyu Hu. Descriptive analytics: Examining expert hackers in web forums. pages 56–63, 09 2014.
- [2] Ugur Akyazi, MJG van Eeten, and C Hernandez Ganan. Measuring cybercrime as a service (caas) offerings in a cybercrime forum. In *Workshop on the Economics of Information Security*, 2021.
- [3] Dimo Angelov. Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*, 2020.
- [4] Tarique Anwar and Muhammad Abulaish. Ranking radically influential web forum users. *Information Forensics and Security, IEEE Transactions on*, 10:1289–1298, 06 2015.
- [5] Nolan Arnold, Mohammadreza Ebrahimi, Ning Zhang, Ben Lazarine, Mark Patton, Hsinchun Chen, and Sagar Samtani. Dark-net ecosystem cyber-threat intelligence (cti) tool. In *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 92–97, 2019.
- [6] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Scene classification via pls. In *Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006, Proceedings, Part IV 9*, pages 517–530. Springer, 2006.
- [7] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.
- [8] José Cabrero-Holgueras and Sergio Pastrana. A methodology for large-scale identification of related accounts in underground forums. *Computers & Security*, 111:102489, 2021.
- [9] Zijian Cai, Zhaoxuan Tan, Zhenyu Lei, Zifeng Zhu, Hongrui Wang, Qinghua Zheng, and Minnan Luo. Lmbot: Distilling graph knowledge into language model for graph-less deployment in twitter bot detection. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM '24*, page 57–66, New York, NY, USA, 2024.
- [10] Giuseppe Cascavilla, Damian A. Tamburri, and Willem-Jan Van Den Heuvel. Cybercrime threat intelligence: A systematic multi-vocal literature review. *Computers & Security*, 105:102258, 2021.
- [11] Chao Chen, Claudia Peersman, Matthew Edwards, Ziauddin Ursani, and Awais Rashid. Amoc: A multifaceted machine learning-based toolkit for analysing cybercriminal communities on the darknet. In *IEEE International Conference on Big Data (Big Data)*, pages 2516–2524, 2021.
- [12] Othmane Cherqi, Youness Moukafih, Mounir Ghogho, and Houda Benbrahim. Enhancing cyber threat identification in open-source intelligence feeds through an improved semi-supervised generative adversarial learning approach with contrastive learning. *IEEE Access*, 11:84440–84452, 2023.
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.
- [14] Marco Di Giovanni and Marco Brambilla. Hierarchical transformers for user semantic similarity. In *International Conference on Web Engineering*, pages 143–157. Springer, 2023.
- [15] Po-Yi Du, Mohammadreza Ebrahimi, Ning Zhang, Hsinchun Chen, Randall A Brown, and Sagar Samtani. Identifying high-impact opioid products and key sellers in dark net marketplaces: An interpretable text analytics approach. In *IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 110–115. IEEE, 2019.
- [16] Yutong Du, Cheng Huang, Genpei Liang, Zhihao Fu, Dunhan Li, and Yong Ding. Expseeker: extract public exploit code information from social media. *Applied Intelligence*, 53(12):15772–15786, Jun 2023.
- [17] Muhammad Shahzad Faisal, Ali Daud, Abubakr Usman Akram, Rabeeh Ayaz Abbasi, Naif Radi Aljohani, and Irfan Mehmood. Expert ranking techniques for online rated forums. *Computers in Human Behavior*, 100:168–176, 2019.
- [18] Zhen Fang, Xinyi Zhao, Qiang Wei, Guoqing Chen, Yong Zhang, Chunxiao Xing, Weifeng Li, and Hsinchun Chen. Exploring key hackers and cybersecurity threats in chinese hacker communities. In *IEEE Conference on Intelligence and Security Informatics (ISI)*, pages 13–18, 2016.
- [19] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.
- [20] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.
- [21] Zellig S. Harris. Distributional structure. *WORD*, 10(2-3):146–162, 1954.
- [22] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.
- [23] Gloria Hristova and Nikolay Netov. Media coverage and public perception of distance learning during the covid-19 pandemic: a topic modeling approach based on bertopic. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 2259–2264. IEEE, 2022.
- [24] Cheng Huang, Yongyan Guo, Wenbo Guo, and Ying Li. Hackerrank: Identifying key hackers in underground forums. *International Journal of Distributed Sensor Networks*, 17(5):15501477211015145, 2021.
- [25] Chenji Huang, Yixiang Fang, Xuemin Lin, Xin Cao, and Wenjie Zhang. Able: Meta-path prediction in heterogeneous information networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(4):1–21, 2022.
- [26] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia tools and applications*, 78:15169–15211, 2019.
- [27] S Joshua Johnson, M Ramakrishna Murty, and I Navakanth. A detailed review on word embedding techniques with emphasis on word2vec. *Multimedia Tools and Applications*, 83(13):37979–38007, 2024.
- [28] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [29] Mikhail V Korotkev. Bert: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943*, 2021.
- [30] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016.
- [31] Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.
- [32] Y Liu, M Ott, N Goyal, J Du, M Joshi, D Chen, O Levy, M Lewis, L Zettlemoyer, and V Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arxiv [preprint](2019)*. *arXiv preprint arXiv:1907.11692*, 1907.
- [33] Ericsson Marin, Jana Shakarian, and Paulo Shakarian. Mining key-hackers on darkweb forums. In *2018 1st International Conference on Data Intelligence and Security (ICDIS)*, pages 73–80, 2018.

- [34] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
- [35] Tomas Mikolov. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [36] Eric Nunes, Ahmad Diab, Andrew Gunn, Ericsson Marin, Vineet Mishra, Vivin Paliath, John Robertson, Jana Shakarian, Amanda Thart, and Paulo Shakarian. Darknet and deepnet mining for proactive cybersecurity threat intelligence. In *IEEE Conference on Intelligence and Security Informatics (ISI)*, pages 7–12. IEEE, 2016.
- [37] 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 '24*, page 368–383, New York, NY, USA, 2024.
- [38] Mandeep Pannu, Iain Kay, and Daniel Harris. Using dark web crawler to uncover suspicious and malicious websites. In Tareq Z. Ahram and Denise Nicholson, editors, *Advances in Human Factors in Cybersecurity*, pages 108–115, Cham, 2019. Springer International Publishing.
- [39] Anum Atique Paracha, Junaid Arshad, and Muhammad Mubashir Khan. Sus you're sus!—identifying influencer hackers on dark web social networks. *Computers and Electrical Engineering*, 107:108627, 2023.
- [40] Sergio Pastrana, Alice Hutchings, Andrew Caines, and Paula Buttery. Characterizing eve: Analysing cybercrime actors in a large underground forum. In *Research in Attacks, Intrusions, and Defenses: 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings 21*, pages 207–227. Springer, 2018.
- [41] 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*, pages 1845–1854, 2018.
- [42] 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. *Advances in neural information processing systems*, 32, 2019.
- [43] 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 re-search, 12. 2011.
- [44] Claudia Peersman, Denny Pencheva, and Awais Rashid. Tokyo, denver, helsinki, lisbon or the professor? a framework for understanding cybercriminal roles in darknet markets. In *2021 APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–12, 2021.
- [45] Ildiko Pete, Jack Hughes, Andrew Caines, Anh V Vu, Harshad Gupta, Alice Hutchings, Ross Anderson, and Paula Buttery. Postcog: A tool for interdisciplinary research into underground forums at scale. In *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 93–104. IEEE, 2022.
- [46] Shahzad Qaiser and Ramsha Ali. Text mining: use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, 181(1):25–29, 2018.
- [47] Priyanka Ranade, Aritran Piplai, Sudip Mittal, Anupam Joshi, and Tim Finin. Generating fake cyber threat intelligence using transformer-based models. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2021.
- [48] Barbara Rosario. Latent semantic indexing: An overview. *Techn. rep. INFOSYS*, 240:1–16, 2000.
- [49] Sagar Samtani and Hsinchun Chen. Using social network analysis to identify key hackers for keylogging tools in hacker forums. In *2016 IEEE conference on intelligence and security informatics (ISI)*, pages 319–321. IEEE, 2016.
- [50] Sagar Samtani, Ryan Chinn, and Hsinchun Chen. Exploring hacker assets in underground forums. In *IEEE international conference on intelligence and security informatics (ISI)*, pages 31–36. IEEE, 2015.
- [51] Sagar Samtani, Ryan Chinn, Hsinchun Chen, and Jay F Nunamaker Jr. Exploring emerging hacker assets and key hackers for proactive cyber threat intelligence. *Journal of Management Information Systems*, 34(4):1023–1053, 2017.
- [52] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC, Heraklion, Crete, Greece, June 3–7, proceedings 15*, pages 593–607. Springer, 2018.
- [53] Matthias Schäfer, Markus Fuchs, Martin Strohmeier, Markus Engel, Marc Liechti, and Vincent Lenders. Blackwidow: Monitoring the dark web for cyber security information. In *11th International Conference on Cyber Conflict (CyCon)*, volume 900, pages 1–21, 2019.
- [54] Geoffrey Stewart and Mahmood Al-Khassawneh. An implementation of the hdbscan\* clustering algorithm. *Applied Sciences*, 12(5):2405, 2022.
- [55] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *Chinese computational linguistics: 18th China national conference, CCL, Kunming, China, October 18–20, proceedings 18*, pages 194–206. Springer, 2019.
- [56] Hind Taud and Jean-Francois Mas. Multilayer perceptron (mlp). *Geomatic approaches for modeling land change scenarios*, pages 451–455, 2018.
- [57] Marie Uncovska, Bettina Freitag, Sven Meister, and Leonard Fehring. Rating analysis and bertopic modeling of consumer versus regulated mhealth app reviews in germany. *NPJ Digital Medicine*, 6(1):115, 2023.
- [58] V. Valeros. A study of rats: Third timeline iteration, 2018.
- [59] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [60] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032, 2019.
- [61] Zhongyi Wang, Jing Chen, Jiangping Chen, and Haihua Chen. Identifying interdisciplinary topics and their evolution based on bertopic. *Scientometrics*, pages 1–26, 2023.
- [62] Lydia Wilson, Viet Anh Vu, Ildikó Pete, and Yi Ting Chua. Identifying and collecting public domain data for tracking cybercrime and online extremism. In *Open Source Investigations in the Age of Google*, pages 281–301. World Scientific, 2024.
- [63] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.
- [64] Yijia Xu, Yong Fang, Cheng Huang, and Zhonglin Liu. Hghan: Hacker group identification based on heterogeneous graph attention network. *Information Sciences*, 612:848–863, 2022.
- [65] Xiacong Yang, James Y Huang, Wenxuan Zhou, and Muhao Chen. Parameter-efficient tuning with special token adaptation. *arXiv preprint arXiv:2210.04382*, 2022.
- [66] Zhilin Yang. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.
- [67] Xiong Zhang, Alex Tsang, Wei Yue, and Michael Chau. The classification of hackers by knowledge exchange behaviors. *Information Systems Frontiers*, 17:1239–1251, 12 2015.
- [68] Yiming Zhang, Yujie Fan, Shifu Hou, Jian Liu, Yanfang Ye, and Thirimachos Bourlai. idetector: Automate underground forum analysis based on heterogeneous information network. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1071–1078. IEEE, 2018.
- [69] Yiming Zhang, Yujie Fan, Yanfang Ye, Liang Zhao, and Chuan Shi. Key player identification in underground forums over attributed heterogeneous information network embedding framework. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 549–558, 2019.
- [70] Yiming Zhang, Yujie Fan, Yanfang Ye, Liang Zhao, Jiabin Wang, Qi Xiong, and Fudong Shao. Kadetector: Automatic identification of key actors in online hack forums based on structured heterogeneous information network. In *2018 IEEE international conference on big knowledge (ICBK)*, pages 154–161. IEEE, 2018.
- [71] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.