Exploring the Evolution of Exploit-Sharing Hackers: An Unsupervised Graph Embedding Approach

Kaeli Otto
Department of Management Information Systems
University of Arizona
Tucson, AZ, United States
kaeliotto@email.arizona.edu

Benjamin Ampel
Department of Management Information Systems
University of Arizona
Tucson, AZ, United States
bampel@email.arizona.edu

Sagar Samtani
Department of Operations and Decision Technologies
Indiana University
Bloomington, IN, United States
ssamtani@iu.edu

Hongyi Zhu
Department of Information Systems and Cyber Security
UTSA
San Antonio, TX, United States
hongyi.zhu@utsa.edu

Hsinchun Chen
Department of Management Information Systems
University of Arizona
Tucson, AZ, United States
hsinchun@arizona.edu

***Abstract* — Cybercrime was estimated to cost the global economy $945 billion in 2020. Increasingly, law enforcement agencies are using social network analysis (SNA) to identify key hackers from Dark Web hacker forums for targeted investigations. However, past approaches have primarily focused on analyzing key hackers at a single point in time and use a hacker's structural features only. In this study, we propose a novel Hacker Evolution Identification Framework to identify how hackers evolve within hacker forums. The proposed framework has two novelties in its design. First, the framework captures features such as user statistics, node-level metrics, lexical measures, and post style, when representing each hacker with unsupervised graph embedding methods. Second, the framework incorporates mechanisms to align embedding spaces across multiple time-spells of data to facilitate analysis of how hackers evolve over time. Two experiments were conducted to assess the performance of prevailing graph embedding algorithms and nodal feature variations in the task of graph reconstruction in five time-spells. Results of our experiments indicate that Text-Associated Deep-Walk (TADW) with all of the proposed nodal features outperforms methods without nodal features in terms of Mean Average Precision in each time-spell. We illustrate the potential practical utility of the proposed framework with a case study on an English forum with 51,612 posts. The results produced by the framework in this case study identified key hackers posting piracy assets.**

Keywords – Hacker forums, social network analysis, unsupervised graph embedding, hacker evolution

I. Introduction

The incidence of cybercrime has increased substantially from a $600 billion annual cost to the global economy in 2018 to an estimated $945 billion annual loss in 2020 [1]. Dark Web hacker forums have emerged as a valuable data source for law enforcement and researchers to monitor hackers engaging in cyber-criminal activities. Dark Web hacker forums provide hackers and cyber-criminals with mechanisms to share malicious tools, exploits, content, and knowledge. Overall, hundreds of hacker forums exist that contain millions of members who post tens of thousands of malicious tools and source code exploits [2]. We present an example of a hacker forum post containing a remote exploit in Figure 1.

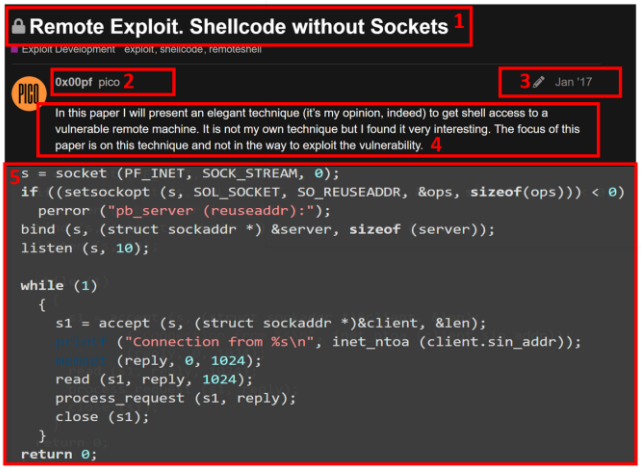

Fig. 1. Example of a Hacker Forum Post. Each forum post has a (1) thread title, (2) user information such as username, member type, join date, (3) a post date, (4) post content, and (5) source code.

Increasingly, law enforcement agencies are using social network analysis (SNA) to identify key hackers within hacker forums for targeted investigations. However, past approaches have primarily focused on analyzing key hackers at a single point in time [2, 3, 4, 5]. How hackers evolve (e.g., transition from novice to expert) in hacker forums remains unknown. Additionally, past studies have analyzed hackers using their structural features (i.e., relationships to other hackers) only. Including features for each hacker (i.e., nodal features) can more comprehensively represent a hacker and their behaviors than structural features alone. Graph embedding techniques can produce fixed length vectors (i.e., embeddings) for each hacker based on their structural and nodal features [6]. However, how to capture the shift of hacker embeddings that encompass nodal and structural features across time-spells requires further investigation.

In this study, we propose the Hacker Evolution Identification Framework to identify how hackers evolve within Dark Web hacker forums. The proposed framework has two novelties in its design. First, the framework captures a rich set of nodal features (e.g., user statistics, lexical measures, and post style) when representing each hacker with unsupervised graph embedding methods. Second, the framework incorporates mechanisms to align

embedding spaces across multiple time-spells of data to facilitate analysis of how hackers evolve over time.

The remainder of this paper is organized as follows. First, we review literature on modeling hackers in hacker forums, unsupervised graph embedding methods, and embedding alignment mechanisms. Second, we present our research gaps and questions. Third, we propose our research framework. Fourth, we present our results and discussion. Finally, we conclude this research and point to promising directions for future research.

II. Literature Review

We reviewed three areas of literature to guide this research. First, we review prevailing methods for modeling hackers in hacker forums to understand methods for analyzing hackers sharing exploit content. Second, we review unsupervised graph embedding methods to identify what algorithms can be leveraged to represent hackers as embeddings in low dimensional space. Third, we review embedding alignment methods to identify mechanisms that can map hacker embeddings across multiple time-spells to facilitate an analysis of how they evolve.

A. Modeling Hackers in Hacker Forums

SNA is the prevailing approach to model hackers in social media [2, 3, 4, 5]. Graphs are commonly denoted as $G = (N, E)$ where G is the entire graph, N is the node set that represents forum users, and E is the edge set that denotes the interactions between forum users. SNA allows scholars to calculate metrics to understand overall graph properties (e.g., density) and centrality measures (e.g., degree, betweenness, eigenvector) to identify key members. To date, SNA has been leveraged to identify key hackers sharing keylogging tools [2], malware [3-4], communication networks [5, 7], and crypters [8]. These studies have revealed several valuable insights for law enforcement, such as discovering that key hackers are often senior administrators or longer-tenured users.

Extant literature has two shortcomings as it pertains to the objective of this study. First, nodal features are often omitted by prior studies when modeling hacker networks. However, these features are essential for understanding a hacker's expertise. Second, most studies do not analyze how hackers evolve. Analyzing how hackers evolve requires representing them based on their relationships and features (e.g., post content, source code). Graph embedding methods can convert nodes in social networks into fixed-length vector representations (i.e., embeddings) that can be used for downstream tasks, such as identifying evolving hackers. Therefore, we review graph embedding methods next.

B. Unsupervised Graph Embedding Methods

Graph embedding methods aim to convert a network and/or its components (e.g., nodes) into a low-dimensional vector (i.e., embedding) with a series of mathematical computations. Unsupervised graph embedding methods are commonly used for cybersecurity tasks where there is little prior knowledge about the graph (e.g., labels, relationships) [9]. Since there is limited apriori knowledge about each hacker in a forum, we restrict our review to unsupervised graph embedding methods that produce an embedding for each node in a network. Unsupervised graph embedding methods can be grouped into three major categories:

- **Matrix factorization methods** such as Text-Associated Deep-Walk (TADW) apply a series of low-rank matrix factorization techniques on a graph's adjacency, degree, and nodal feature matrices to produce an embedding [10].
- **Random walk-based methods** such as Node2Vec use random walks, biased random walks, and skip-gram to learn a node's embedding [11].
- **Deep neural network-based methods** such as Graph Convolutional Autoencoder (GCAE) and Graph Attention Autoencoder (GATE) leverage neural network-based autoencoders to reconstruct a graph's adjacency matrix using graph convolution and graph attention layers, respectively [12, 13].

The prevailing unsupervised graph embedding methods that can account for nodal features are the matrix factorization-based TADW and the deep neural network-based GCAE and GATE. These methods have potential utility in comprehensively modeling a hacker based on their relationships and features. However, identifying how a hacker evolves requires generating embeddings for a hacker in multiple time-spells and computing their shifts. While embeddings within a time-spell can be compared, embeddings across time-spells need to be aligned in order to model how a hacker shifts. Therefore, we review embedding alignment methods next.

C. Embedding Alignment Methods

The prevailing approach to align embedding space first constructs a matrix of embeddings,

$$\boldsymbol{W}^{(t)} \in \mathbb{R}^{\mathrm{d}\, x\, |V|}, \quad (1)$$

where $\boldsymbol{W}^{(t)}$ is the embedding matrix and t is the time-spell [14]. Embedding spaces across time-spells can then be aligned by optimizing,

$$\mathrm{R}^{(t)} = argmin_{Q^T Q = I} \left\| \boldsymbol{W}^{(t)} \boldsymbol{Q} - \boldsymbol{W}^{(t+1)} \right\|_{\mathrm{F}}, \quad (2)$$

where $\|\cdot\|_F$ denotes the Frobenius norm representing the orthogonal alignment of embedding spaces across time-spells. Aligning embedding spaces in this fashion enables the calculation of how a hacker has shifted (i.e., evolved) with cosine distance $\text{cosine-dist}(e_t, e_{t+\Delta})$, where e_t is the location of an embedding within the embedding space at time t, and $e_{t+\Delta}$ is the location of the same embedding in the embedding space after time t. Although having potential utility for mapping hackers, these computations have only been conducted for text graphs without nodal features but not for identifying evolving hackers [14].

III. Research Gaps and Questions

We identified three research gaps from our literature review. First, despite their potential to capture a hacker's behavior, nodal features (e.g., a hacker's post content) are often omitted when representing and modeling hackers with SNA. Second, although unsupervised graph embedding algorithms such as TADW, GCAE, and GATE can be leveraged to represent each hacker in a social network based on their relationships and nodal features, most studies do not conduct such an analysis. Third,

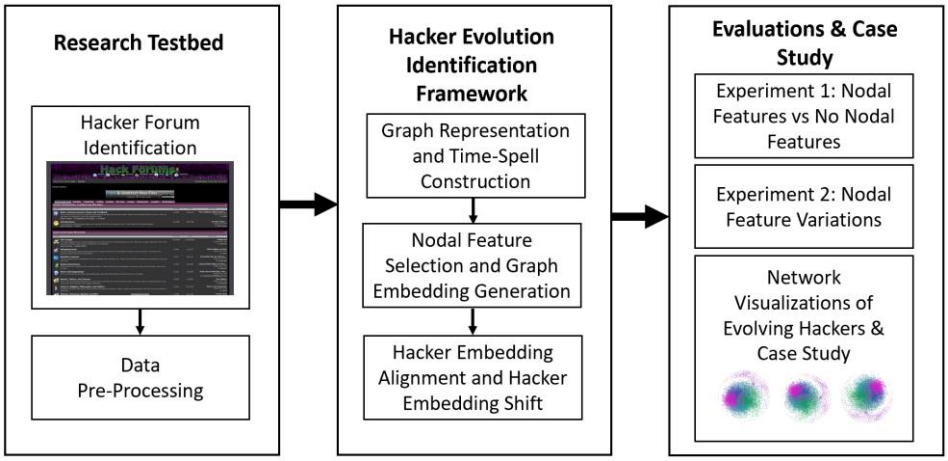

Fig. 2. Proposed Research Framework

prevailing embedding alignment mechanisms have been used to operate on text graphs; how to operate on social networks requires additional examination. Based on these research gaps, we pose the following questions for study:

1. How can we extract and include a hacker's features when generating an embedding for each hacker in online hacker forums?
2. How can we develop a framework that incorporates embedding alignment mechanisms to operate on hacker embeddings and identify how hackers evolve in online hacker forums?

IV. Research Design

To answer the posed research questions, we developed a research framework (Fig. 2): with three components: (1) Research Testbed, (2) Hacker Evolution Identification Framework, and (3) Evaluations & Case Study. We discuss each component in the following sub-sections.

A. *Research Testbed: Hacker Forum Identification and Data Pre-Processing*

We identified three English hacker forums that are well-known in the online hacker community for containing large quantities of malicious source code [15]. A web crawler routed through the TOR network collected the content and metadata (e.g., hacker name, post dates) for each post from each forum. We summarize the testbed in Table I.

TABLE I. Hacker Forum Research Testbed

Name	Posts	Date Range	Authors	Threads	Source Code
Cipher	51,612	5/1/2015-12/31/2020	3,551	5,198	2,207
go4expert	62,103	12/25/2004-12/31/2020	15,213	19,158	5,800
AntiOnline	291,914	4/10/2002-12/31/2020	13,017	55,086	2,063
Total:	405,629	4/10/2002-12/31/2020	31,781	79,442	10,070

The three hacker forums contained 405,629 posts, 31,781 unique authors, 79,442 threads, and 10,070 source code snippets ranging in post-date from 2002 to 2020. To prepare hacker forums for SNA, the forums were queried to only include threads that had at least one post containing source code. This ensures that the results capture malicious hackers who are sharing source code exploits and excludes unrelated threads and users [14].

B. *Hacker Evolution Identification Framework*

Given the limitations of previous literature as it pertains to identifying evolving hackers, we propose a novel Hacker Evolution Identification Framework. The proposed framework (Fig. 3) operates in five steps: (1) Graph

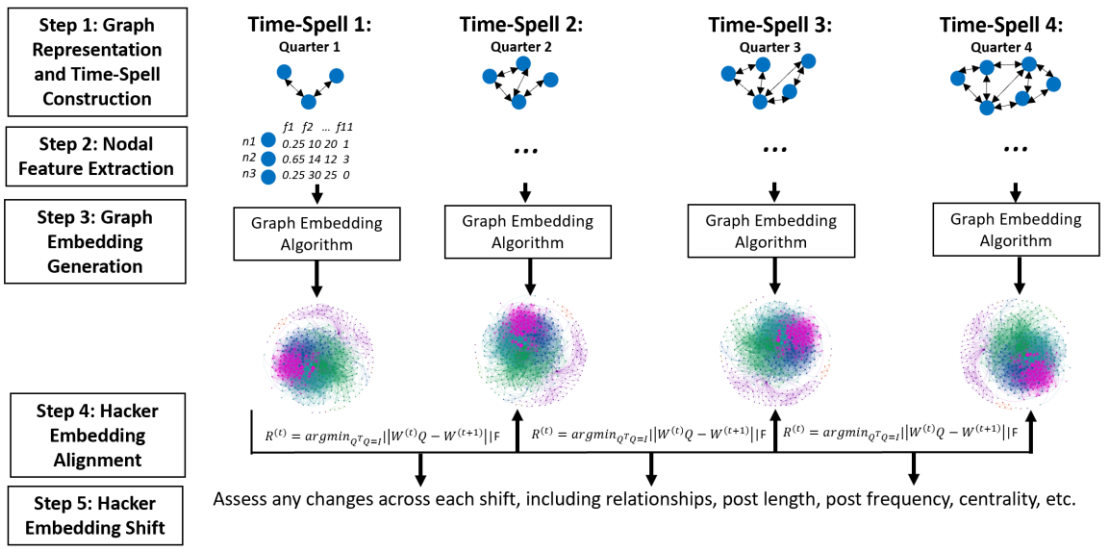

Fig. 3. Hacker Evolution Identification Framework

Representation and Time-Spell Construction, (2) Nodal Feature Extraction, (3) Graph Embedding Generation, (4) Hacker Embedding Alignment, and (5) Hacker Embedding Shift. We present each step in the following sub-sections.

a) Graph Representation & Time-Spell Construction

Each hacker forum network is denoted as $G = (N, E, F)$ where G is an undirected graph, N is the node set of forum users, E is the undirected edge set of users who posted in the same source code containing thread, and F is the feature matrix for each user. To understand the overall dynamics of each collected hacker forum, we present summary statistics about each graph in Table II.

TABLE II. HACKER FORUM NETWORK STATISTICS

Hacker Forum	Cipher	Go4Expert	AntiOnline
Number of Nodes	2,010	3,365	1,046
Number of Edges	109,176	24,273	4,938
Max Degree	1,023	419	131
Average Degree	108.63	14.43	9.44
Average Betweenness	1,096.39	3,682.38	1,195.77
Graph Density	0.05	0.004	0.009

Go4Expert had the highest quantity of nodes with 3,365. However, Cipher possessed the largest number of edges with 109,176. As a result, Cipher has the highest average degree. This indicates that forum users in Cipher can more readily distribute and access exploit source code than the users in Go4Expert and AntiOnline. Cipher also has the highest graph density indicating that malicious code can spread more quickly through Cipher than the other forums. Consistent with past hacker forum literature, our datasets were split into three-month timeframes from 2015-2019 [14].

b) Nodal Feature Selection

Since past studies often omit nodal features from their analysis, we propose a rich set of features for each hacker in each time-spell based on past hacker forum literature and recommendations from FBI analysts who investigate darknets. We propose two categories of features: user and post content. We present the proposed features in Table III.

TABLE III. PROPOSED FEATURE SETS *NOTE: FS = FEATURE SETS

Category	Features*		Description
User	User Statistics *(FS1)*	Post Frequency	# of posts users makes
		Account Age	Post date minus user join date converted into days
	Node-Level Metrics *(FS2)*	Degree	Measure of user's immediate influence
		Betweenness	Measures importance of user's social position
		Closeness	Measures how quickly a user can reach another user
		Eigenvector	Measures strength of forum user's connections
Post Content	Lexical Measures *(FS3)*	Average Sentence Length	Average number of words in a hacker's sentences
		Average Words per Post	Average number of words in a post
	Post Style *(FS4)*	Punctuation	Count of punctuation
		Source Code	Whether or not user included source code
		Exploit Terms	Exploit terms occurrence

The user features comprise of user statistics (FS1) and node-level metrics (FS2). User statistics include features provided directly from each forum, including post frequency and the age of the user's account. Node-level metrics refer to each hacker's centrality measures (e.g., degree, betweenness, closeness, eigenvector). Past literature studying hacker communities has indicated that including user statistics and node-level metrics into an analytical method captures a hacker's relationships and structural characteristics [7, 16].

The post content category of features comprise of lexical measures (FS3) and post style (FS4). Lexical features include average sentence length and average words per post. Post style refers to how hackers compose their posts. Post style features included punctuation, source code, and exploit terms. Both FS3 and FS4 were included since past literature has indicated that advanced hackers will typically have longer sentences with more exploit terminology compared to novice hackers [8, 15, 16].

c) Graph Embedding Generation

Since our objective is to identify how hackers evolve based on their relationships and features, we leverage a prevailing unsupervised graph embedding algorithm TADW to generate an embedding for each node. TADW operates on each time-spell's graph. TADW was selected as it is designed for high-dimensional nodal features (e.g., nodes with text features) [10]. TADW has also shown strong performances in cybersecurity tasks [9].

d) Hacker Embedding Alignment

Identifying how a hacker shifts across time-spells requires first aligning embedding spaces across time-spells. To achieve this task, embeddings generated from TADW are aligned with the optimization function,

$$\mathrm{R}^{(t)} = argmin_{Q^T Q = I} \left\| \boldsymbol{H}^{(t)} \boldsymbol{Q} - \boldsymbol{H}^{(t+1)} \right\|_{\mathrm{F}}. \quad (3)$$

where $\boldsymbol{H}^{(t)}$ is the matrix of hacker embeddings at time-spell t and $\|\cdot\|_F$ denotes the Frobenius norm representing the orthogonal alignment of embedding spaces across time-spells. Aligning spaces using the optimization function facilitates the computation of a hacker's profile shift across time-spells.

e) Hacker Embedding Shift

The final component of our framework aimed to identify how much hackers shift (i.e., evolve) across the aligned embedding spaces. To execute this, we computed the hacker embedding shifts with cosine distance,

$$\text{cosine-dist}(h_t, h_{t+\Delta}), \quad (4)$$

where h_t is the location of the hacker within the embedding space at time t, and $h_{t+\Delta}$ is the location of the same hacker in the embedding space after time t. This computation helps to identify hackers who have evolved the most by looking at changes such as the formation of new relationships or the increase or decrease in post length or frequency.

C. Evaluations & Case Study

Evaluating unsupervised graph embedding methods is often conducted by comparing the quality of multiple algorithms in a downstream task [9, 14]. We conducted two experiments for this research. In Experiment 1, we evaluated the performance of the three prevailing unsupervised graph embedding methods: TADW, GCAE, and GATE when using all of the nodal features versus no nodal features [12, 13]. In Experiment 2, we evaluated all combinations of feature sets (15 total) for each algorithm.

Following best practice in cybersecurity literature, we evaluated the quality of the embeddings generated from each method in the downstream task of graph reconstruction [9]. We used the Mean Average Precision (MAP) metric to measure the performance of each algorithm. MAP evaluates the quality of an embedding based on how well it constructs the original graph by calculating the average precision of each node. MAP returns a scalar value between zero and one, where scores of 0.70 or higher are considered to be high quality [17].

To demonstrate the practical utility of our framework, we executed a case study that applied the proposed Hacker Evolution Identification Framework to identify evolving hackers. We executed each experiment and the case study on the Cipher forum as it has the highest average degree and graph density compared to other forums in our testbed.

V. Results & Discussion

A. Experiment 1: Nodal Features vs No Nodal Features

In Experiment 1, we aimed to identify how TADW, GCAE, and GATE performed with and without nodal. Evaluation results are summarized in Table IV. For space considerations, we listed the MAP performances for Cipher for each year. The top performances appear in bold-face.

TABLE IV. Experiment 1 Results: Nodal Features vs No Nodal Features (*Note: n = number of nodes)

Feature Variation	Model	2015* (n=228)	2016 (n=1,136)	2017 (n=1,580)	2018 (n=1,845)	2019 (n=1,989)
No Nodal Features	TADW	0.487	0.189	0.144	0.140	0.132
	GCAE	0.468	0.157	0.105	0.098	0.116
	GATE	0.637	0.244	0.194	0.170	0.160
All Features	TADW	**0.759**	**0.489**	**0.449**	**0.410**	**0.380**
	GCAE	0.427	0.083	0.075	0.062	0.075
	GATE	0.448	0.109	0.077	0.054	0.000

Overall, TADW attained the second best MAP score (after GATE) every year when no nodal features were included. When all nodal features are included, TADW's performance improved by at least 0.25 in each time-spell (e.g., from 0.487 to 0.759 in 2015). However including nodal features into GCAE and GATE caused their performances to decline substantially. For example, GATE's performance dropped from 0.637 to 0.448 (0.311 lower than TADW) in 2015 when nodal features were included. TADW's performance improvement is likely attributable to its use of random walks to represent structural features and a low-rank matrix factorization approach to producing embeddings from nodal and structural features. These approaches help preserve nodal features in the generated embedding [10]. In contrast, both GCAE and GATE leverage autoencoder architectures that aim to produce embeddings through feed-forward, back-propagation, and error correction computations. These methods can often emphasize structural features over nodal features when producing embeddings [12, 13].

B. Experiment 2 Results: Nodal Feature Variations

In Experiment 2, we aimed to identify which combination of features generates the highest quality embedding. For space considerations, we only present the results of the feature set variations for TADW (the top-performing algorithm from Experiment 1). The top-performing feature variations appear in bold-face.

TABLE V. Experiment 2: Nodal Feature Variations (*Note: n = number of nodes)

Nodal Feature Variation	2015* (n=228)	2016 (n=1,136)	2017 (n=1,580)	2018 (n=1,845)	2019 (n=1,989)
FS1	0.663	0.314	0.280	0.233	0.214
FS2	0.680	0.428	0.413	0.421	0.435
FS3	0.531	0.181	0.125	0.111	0.113
FS4	0.653	0.316	0.274	0.257	0.242
FS1 & FS2	0.695	0.375	0.317	0.281	0.263
FS1 & FS3	0.772	0.402	0.327	0.306	0.303
FS1 & FS4	0.707	0.396	0.371	0.386	0.364
FS2 & FS3	0.749	0.476	0.427	0.402	0.409
FS2 & FS4	0.677	0.363	0.315	0.308	0.281
FS3 & FS4	0.689	0.428	0.423	0.412	0.336
FS1, FS2, FS3	0.743	0.451	0.377	0.338	0.321
FS1, FS2, FS4	0.713	0.423	0.397	0.399	0.352
FS1, FS3, FS4	**0.778**	**0.505**	**0.467**	**0.445**	0.433
FS2, FS3, FS4	0.762	0.439	0.420	0.436	**0.454**
All Features	0.759	0.489	0.449	0.410	0.380

TADW with FS1 (user statistics), FS3 (lexical measures), and FS4 (post style) attained the highest MAP from 2015-2018, and the second highest MAP for 2019. The results also indicate that MAP scores improve as additional feature sets are included. However, TADW using all nodal features attained a lower performance than the FS1, FS3, and FS4 variation in every year. This indicates that including FS2 along with other feature sets results in a degradation of performance. A possible explanation for these results may be because TADW may

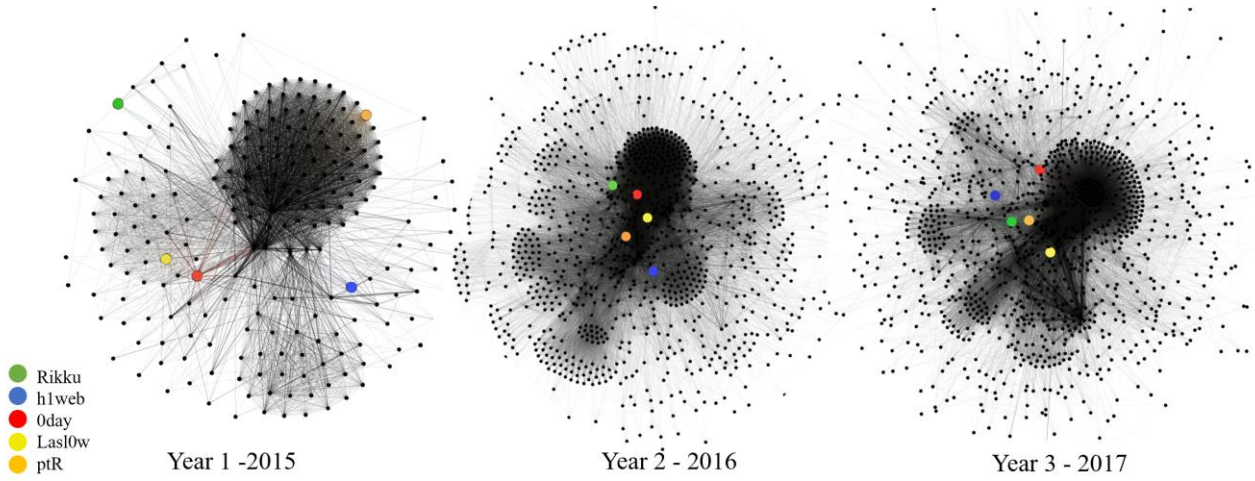

Fig. 4. Network Visualization of Top Five Forum Users With Large Profile Shifts Over 3-Year Span

represent centrality scores when capturing the structural features for each node [10]. Including FS2 would therefore duplicate this information and may decrease MAP scores.

C. Network Visualizations of Evolving Hackers and Case Study Results

Based on MAP results, we used TADW with nodal features (FS1, FS3, and FS4) to produce an embedding for each hacker in each time-spell. We employed the graph embedding alignment and shift calculations for each hacker's shifts. The top five users with large profiles shifts are: Rikku – 1.035, h1web – 0.909, 0day – 0.822, Lasl0w – 0.773, and prR – 0.738. The five users are color-coded and presented in network visualizations (2015-2017) in Fig. 4.

The five hackers with the largest average profile shifts became more central in their networks over time. The hacker Rikku (green node; largest average shift) moved from the edge of the network in 2015 to the center in 2017. We visualize Rikku's evolution in terms of post length, count, and content over time in Fig. 5.

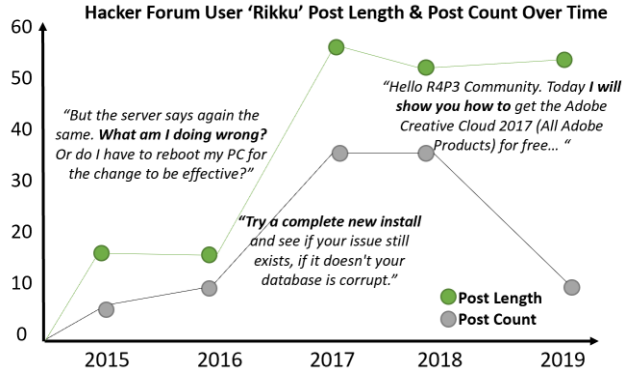

Fig. 5. Hacker Forum User Rikku's Evolution Between 2015-2019

In 2015, Rikku requested help from other hackers with phrases such as "What am I doing wrong?" and "Do I have to reboot my PC for the change to be effective?" In 2016, Rikku started helping others by answering questions (e.g., "Try a complete new install and see if your issue still exists…"). Finally, Rikku posted piracy specific assets in 2017, including a tutorial on how to bypass paying for Adobe products. Sharing cyber-criminal tools, code, and tutorials are an indicator of expertise [16]. These insights are not attainable with extant approaches for hacker SNA.

VI. Conclusions & Future Directions

Cybercrime has become a significant societal issue. In this study, we proposed a novel Hacker Evolution Identification Framework to help identify how hackers evolve in hacker forums. The proposed framework captured a rich set of nodal features when representing each hacker with unsupervised graph embedding methods. Additionally, the framework incorporated an optimization mechanism to align embedding spaces across multiple time-spells of data to facilitate an analysis of how hackers evolve over time.

There are two promising future directions for research. First, the framework could be extended to predict how hackers will shift in their behavior to support proactive investigations. Second, future work can examine how hacker communities grow, shift, and dissolve over time. Each direction could provide valuable capabilities for law enforcement to combat cybercrime.

VII. Acknowledgment

This work is supported in part by the National Science Foundation under Grant Numbers DGE-1921485 (SFS), OAC-1917117 (CICI), and CNS-1850362 (SaTC CRII).

VIII. References

[1] Barclay, B., "Cybercrime Apparently Cost the World over $1 Trillion in 2020," *TechRadar,* 2021. Accessed on: Aug. 23, 2021. [Online]. Available: https://www.techradar.com/news/cybercrime-cost-the-world-over-dollar1-trillion-in-2020.

[2] Samtani, S. & Chen, H., "Using Social Network Analysis to Identify Key Hackers for Keylogging Tools in Hacker Forums," *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, Tucson, Arizona, pp. 319-321, 2016.

[3] MacDonald, M. & Frank, R., "The Network Structure of Malware Development, Deployment and Distribution," *Global Crime,* vol. 18, no. 1, pp. 49-69, 2017.

[4] Grisham, J., Samtani, S., Patton, M., & Chen, H., "Identifying Mobile Malware and Key Threat Actors in Online Hacker Forums for Proactive Cyber Threat Intelligence," *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Beijing, China, pp. 13-18, 2017.

[5] Johnsen, J. W. & Franke, K., "Feasibility Study of Social Network Analysis on Loosely Structured Communication Networks," *2017 International Conference on Computational Science (ICCS),* Zurich, Switzerland, pp. 2388-2392, 2017.

[6] Cai, H., Zheng, V. W., & Chang, K. C. C., "A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1616-1637, 2018.

[7] Kwon, K. H., Yu, W., Kilar, S., Shao, C., Broussard, K., & Lutes, T., "Knowledge Sharing Network in a Community of Illicit Practice: A Cybermarket Subreddit Case," *Proceedings of the 53rd Hawaii International Conference on System Sciences,* Manoa, Hawaii, pp. 1-10, 2020.

[8] Samtani, S., Chinn, R., Chen, H., & Nunamaker, J. F., "Exploring Emerging Hacker Assets and Key Hackers for Proactive Cyber Threat Intelligence," *Journal of Management Information Systems,* vol. 34, no. 4, pp. 1023–1053, 2017.

[9] Lazarine, B., Samtani, S., Patton, M., Zhu, H., Ullman, S., Ampel, B., & Chen, H., "Identifying Vulnerable GitHub Repositories and Users in Scientific Cyberinfrastructure: An Unsupervised Graph Embedding Approach," *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Washington, DC, pp. 1-6, 2020.

[10] Yang, C., Liu, Z., Zhao, D., Sun, M., & Chang, E., "Network Representation Learning with Rich Text Information," *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, pp. 1-7, 2015.

[11] Grover, A. & Leskovec, J., "node2vec: Scalable Feature Learning for Networks," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* San Francisco, California, pp. 855-864, 2016.

[12] Kipf, T. & Welling, M., "Semi-Supervised Classification with Graph Convolutional Networks," *5th International Conference on Learning Representations*, Toulon, France, pp. 1-14, 2017.

[13] Salehi, A. & Davulcu, H. "Graph Attention Auto-Encoders," *arXiv:1905.10715 [cs.LG],* pp. 1-10, 2019.

[14] Samtani, S., Zhu, H., & Chen, H., "Proactively Identifying Hacker Threats from the Dark Web: A Diachronic Graph Embedding Framework (D-GEF)," *ACM Transactions of Privacy and Security,* vol. 23, no. 4, pp. 1-33, 2020.

[15] Ampel, B., Samtani, S., Zhu, H., Ullman, S., & Chen, H., "Labeling Hacker Exploits for Proactive Cyber Threat Intelligence: A Deep Transfer Learning Approach," *2020 IEEE International Conference on Intelligence and Security Informatics (ISI),* Washington, DC, pp. 1-6, 2020.

[16] Benjamin, V., Valacich, J., & Chen, H., "DICE-E: A Framework for Conducting Darknet Identification, Collection, Evaluation with Ethics," *MIS Quarterly*, vol. 43, no. 1, pp.1-22, 2019.

[17] Ciu, P., Wang, X., Pei, J., & Zhu, W., "A Survey on Network Embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 833-852, 2019.