SmishViz: Towards A Graph-based Visualization System for Monitoring and Characterizing Ongoing Smishing Threats

Seyed Mohammad Sanjari
Department of Computer Science
Tennessee Tech University
Cookeville, Tennessee, USA
ssanjarip42@tntech.edu

Ashfak Md Shibli
Department of Computer Science
Tennessee Tech University
Cookeville, Tennessee, USA
ashibli42@tntech.edu

Maraz Mia
Department of Computer Science
Tennessee Tech University
Cookeville, Tennessee, USA
mmia43@tntech.edu

Maanak Gupta
Department of Computer Science
Tennessee Tech University
Cookeville, Tennessee, USA
mgupta@tntech.edu

Mir Mehedi Ahsan Pritom
Department of Computer Science
Tennessee Tech University
Cookeville, Tennessee, USA
mpritom@tntech.edu

Abstract

SMS phishing (aka 'smishing') threats have grown to be a serious concern for mobile users around the globe. In cases of successful smishing, attackers take advantage of users' trust through deceptive text messages to trick them into downloading malicious content, disclosing private information, or becoming victims of fraud. Current studies on smishing mostly focus on the classification of smishing (or spam) messages from benign ones as a means of defense. However, there is no systematic study to characterizing smishing threats and their landscapes by which we can monitor the ongoing campaigns from a bird's-eye perspective to apply effective defense. In this paper, we propose *SmishViz*, a graph-based visualization system that can aid defenders (i.e., analysts) to characterize ongoing smishing threats in the wild and allow them to monitor the connected campaigns and campaign-operations through effective graph visualization approach integrated with state-of-the-art open-source visualization tool. This paper also provides case study with real-world smishing dataset to showcase the efficacy of *SmishViz* system in practical use-case scenarios. Our case study results reveal that the proposed system can certainly help defenders to track and monitor ongoing smishing campaigns, understand attackers' tactics to formulate strategic defense and uproot the attack operations.

CCS Concepts

• **Security and privacy** → **Mobile and wireless security**; • **Human-centered computing** → *Visualization toolkits*; **Graph drawings**.

Keywords

SMS phishing, smishing, visualization, mobile threats

CODASPY '25, June 4–6, 2025, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-1476-4/2025/06
https://doi.org/10.1145/3714393.3726499

ACM Reference Format:
Seyed Mohammad Sanjari, Ashfak Md Shibli, Maraz Mia, Maanak Gupta, and Mir Mehedi Ahsan Pritom. 2025. SmishViz: Towards A Graph-based Visualization System for Monitoring and Characterizing Ongoing Smishing Threats. In *Proceedings of the Fifteenth ACM Conference on Data and Application Security and Privacy (CODASPY '25), June 4–6, 2025, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3714393.3726499

1 Introduction

In today's digital age, smartphones have become an essential part of our daily lives. We rely on them for communication, banking, shopping, and many other online browsing activities. Unfortunately, this reliance has also made mobile device users a prime target for cyber-criminals. SMS Phishing attacks also known as 'Smishing' have emerged as a significant threat, taking advantage of users' trust through text messages to trick them into revealing sensitive personal information or installing malicious software [1, 26].

Recent studies have also revealed that users do fall victim to these smishing attacks and interact with smishing messages more than often [20, 25].

Smishing attacks can have serious consequences for both individuals and organizations. A successful smishing attack can lead to identity theft, financial losses, data breaches, and reputation damage through a mobile device [4]. We see reports showing sharp 1245% increase of smishing attacks in Q1 2023 from the previous quarter [30]. According to the fraud reports by the Federal Trade Commission (FTC), only during Q1 of 2024, a total loss of $645.7 million has been reported in United States alone through frauds phone calls and text messages [6]. Attackers are constantly coming up with new stealthy strategies to penetrate users' trust making the job even more challenging for defenders [2].

Traditional methods for smishing detection, such as blacklists based on contents and URLs within texts are becoming less effective as attackers continuously adapt their techniques and tactics [13]. To keep up with the ever-evolving threat landscape, it is essential to develop more advanced and robust defense mechanisms, which requires a deep understanding of the characteristics and infrastructures of smishing attacks. We believe we can understand smishing ecosystem better through a comprehensive analysis and continuous monitoring of real-world smishing attacks. Usually, within

the real-world smishing dataset, we observe that various themes of smishing messages can be identified based on textual (string-based) similarity metrics [13, 31]. However, there might be connection between completely dissimilar text message groups with different topic themes based on the web-entity infrastructures (e.g., website URLs and domain names associated with the SMS). These characteristics might unveil larger attack campaign operation owners who might be running multiple different thematic topic based smishing campaigns targeting various brands. We hypothesize that building these web infrastructural-level connections between various message groups and visualizing them through a connected graph may aid defenders uncover campaign operations and dismantle them effectively. Hence, in this paper, we propose *SmishViz*, a graph-based visualization system, to observe and capture smishing campaigns and campaign-operations that are operated by same bad actor (or actors) using common underlying web-entity infrastructures and/or using similar message templates. We also propose to continuously monitor smishing messages in real-time and extract campaign-operations to proactively take defensive actions against various ongoing smishing campaigns. In this paper, we make the following four major contributions.

- First, we propose to create similar topic-themed message clusters using sentence embeddings to characterize smishing messages into various meaningful groups and investigate them further for finding within-group campaigns (i.e., sharing high textual similarity).
- Second, we propose to group similar semantic topic-themed messages together as clusters using pre-trained *BERTopic* model, find highly similar text pattern (i.e., template-based) groups within each clusters using Ratcliff patter recognition algorithm to identify them as sub-clusters representing campaigns. These individual sub-clusters from various topic-themed clusters are then connected based on shared web-entity infrastructures to form bigger campaign-operations that are possibly coordinated by the same bad actor(s).
- Third, we propose to visually analyze the campaign-operations as connected tripartite graphs and get data-driven insights from each campaign-operations through the web-based visualization system integrated with *D3.js* state-of-the-art visualization tool.
- Fourth, we present case study with a recently published real-world smishing dataset to characterize smishing threats and showcase the practicality of the proposed visualization system in understanding smishing campaigns and operations that can aid defenders create further preventive measures.

Paper Outline: Section 2 discuss the current literature of SMS phishing attacks and defenses. Section 3 describes the research questions (RQs) and detail functionalities of SmishViz system modules. Section 4 presents the case study with real-world smishing dataset to answer the RQs and show the use-case scenarios. Section 5 discusses the limitations. Finally, section 6 concludes the paper.

2 Related Works

Smishing attacks are growing rapidly and that is why the field has received an enhanced attention from the security community in recent years. In recent notable studies, Nahapetayan et al. [18] has analyzed a large amount of public SMS gateways' data to understand SMS phishing tactics and characterize the underneath infrastructures. They have analyzed SMS phishing campaigns and operations, identifying patterns such as multiple attacks targeting the same receiver's cell phone numbers and the use of shortened URLs, which has immensely inspired our study. In the literature, we find some studies have proposed using machine learning (ML) to analyze words and links in text messages for smishing detection, but have scoped their approach around a limited set of hand-picked clues or manually selected features [17]. In another follow-up study, Jain et al. [13] closely analyzed the words and URLs in smishing messages to find smish indicators. Similarly, Goel and Jain [9] proposed an ML-based classifier framework, which identifies smishing messages based on the SMS text contents. Next, Mishra and Soni [17] have introduced a promising NLP and ML-based tool named *DSmishSMS* for identifying smishing messages, but the evaluation is not comprehensive due to dataset limitations. In another study, Yeboah-Boateng et al. [34] have assessed the threats of phishing, smishing, and vishing attacks against mobile devices, while Wu et al. [33] proposed defense schemes against phishing attacks on mobile platforms. Additionally, Hossain et al. [12] have proposed using deep learning (DL) models, such as CNN and LSTM, to detect spam and phishing SMS, which achieved good results but they only considered the text's word frequency feature such as Term Frequency–Inverse Document Frequency (TF–IDF) which did not take into account some core aspects of a smishing attack like the associated URL, brand impersonation, and sender information. Furthermore, researchers have introduced computer vision techniques to analyze the similarity of looks among the phishing [14], messages but visual similarities are not always reflect on the similarities in word or metadata levels as stated in phishing literature [19]. These studies highlight the sophistication of mobile-based attacks and our research builds upon this foundation by providing a graph-based data-driven visualization system to further investigate underneath smishing campaigns and bigger operations.

In addition, existing literature also discuss how generative AI can be abused by attackers for malicious purposes and particularly create newer and previously unseen smishing and phishing campaigns [11, 24, 28], which indicates that previous knowledge based content-driven defense solutions may not be enough for detecting new deceiving AI generated campaigns. Moreover, researchers also proposed to leverage LLM-empowered defense mechanism to detect smishing messages with natural language based reasoning [27]. Lastly, existing literature have also explored attack vectors, users susceptibility and users awareness when exposed to smishing attacks by conducting user studies to conclude that users do fall victim to these attacks at an alarming rate [8, 20].

3 Methodology and System Overview

In this section, we detail the driving research questions, problem formulation, and define the integrated modules of *SmishViz* and their functionalities.

Research Questions (RQs): We address five major research questions (RQs) in this paper.

- **RQ1:**Can we identify clusters of messages with high similarity in meanings and themes (e.g., topics) and automatically

label them with corresponding themed topics using data-driven text analysis?
- **RQ2:** Can we find groups of messages within a cluster that share templates or patterns to be identified as a same-origin (i.e., created by same bad actor) campaign?
- **RQ3:** Are there evidence that these same-origin campaigns may connect multiple sub-groups of messages from multiple clusters to form campaign-operations based on common infrastructures or underneath connections?
- **RQ4:** How can we effectively visualize the campaigns and campaign-operations that can aid any cyber defenders (i.e., analysts)?

3.1 Problem Formalization and Notations

The ideation of *SmishViz* system is to integrate with a live SMS dataset that collects data continuously from the wild and then investigate the data to monitor ongoing smishing campaigns. In order to formally define the system, let's first define some notations and terminologies that are used in the rest of the paper–

- Each incoming SMS into *SmishViz* is represented as a tuple $M_i = (m_i, s_i, w_i)$, where, m_i denotes the text content of the i-th SMS after removing URLs or links; s_i denotes the sender information (email, phone numbers); and w_i presents the web-entity present in the i-th SMS that can be collected from to a corresponding domain name d_i or URL string u_i.
- $C_J = \{\{m_i, m_j, m_k, \cdots\}, Th_J\}$represents the J-th SMS cluster, generated using BERTopic, a topic modeling technique that leverages BERT embeddings and density-based clustering. Each cluster contains messages with semantically similar content, grouped together based on their contextual relationships and assign a corresponding theme Th_J for the whole cluster.
- $S_K \in C_J$ represents the K-th sub-cluster of messages within cluster C_J that can be grouped together because of high pattern similarity (i.e., using a common template). The cluster C_J can also be presented as a set of sub-clusters, $C_J = \{S_1, S_2, \cdots, S_K\}$ where the sub-clusters may vary in text patterns but they have same or similar topics. Moreover, each of these sub-clusters also represent individual campaigns that can be safely assumed to be originated by the same bad actor.
- CO_L represents the L-th campaign-operation that connects multiple sub-clusters from same or different clusters based on common web entity infrastructures between them. This campaign operation will help to identify sub-clusters that are of different themes and topics but somehow connected based on the underneath infrastructure which indicates a common source of origin. Formally, $CO_L = \{S_X \in C_I \cup S_Y \in C_J\}$ when there is common web entity between messages in sub-cluster $S_X \in C_I$ and sub-cluster $S_Y \in C_J$.

Figure 1 presents the system overview of *SmishViz* and its modules. First, it takes the raw bulk SMS data as input from any public smishing repository D_{sms}. Next, it curates each SMS $M_i \in D_{sms}$ by extracting (m_i, s_i, w_i) tuples. Then, it applies *BERTopic* [10]embedding to find semantic similarity among these messages and group them together into set of clusters $\{C_1, C_2, \cdots, C_J\}$, where each cluster will ideally contain messages of unique topics or themes. Next, we apply pattern matching algorithms between messages within cluster boundary to find sub-clusters ($S_K \in C_J$) within a particular cluster C_J that exhibits very high similarity in patterns, meaning they are most likely using a common template. Additionally, we also connect these sub-clusters from different clusters based on the common web entity infrastructures to form any campaign-operations CO_L. Each campaign-operation, by our definition, can be presented as strongly connected graphs and visualized for further analysis using state-of-the-art visualization tools. Furthermore, any campaign-operation can be identified as connected smishing threats originated from the same source or bad actors. Here, each campaign-operation can be represented as a tripartite graph network where sub-clusters nodes are connected to some message nodes on one side and web-entity nodes on the other side as shown in Figure 2.

3.2 SMS Data Collection and Pre-Processing Module

In this module, we collect raw SMS data D_{sms} from any public facing smishing dataset repository. However, raw data usually have noises and inconsistencies that makes our analysis and clustering task more challenging. To deal with these issues, we use a data pipeline for each individual SMS M_i that helps us preparing the data. At first, we extract the tuple (m_i, s_i, w_i) from SMS M_i. Next, we need to further clean up the extracted message text part (i.e., m_i) to remove stop words, digits, and punctuations to get a clean version of text [35] without losing the meaningful context. Additionally, some more technical details on cleaning the URL data in raw SMS data those encounter additional space characters for the usage of optical character reading techniques are discussed in Appendix A.

3.3 Semantic Similarity-Based Clustering Module With BERTopic

Linguistic analysis of smishing texts can reveal indicating patterns and red flags associated with malicious characteristics. The main motivation behind the semantic similarity based clustering approach is to group the messages together that are covering same thematic topics (e.g., delivery theme, account theme, lottery theme) and potentially similar type of target brands (e.g., FedEx, USPS). Moreover, we hypothesize that smishing messages originating from the same bad actors would share common textual patterns (i.e., linguistic templates) and common web entity infrastructures. Thus, grouping the text messages based on semantic similarity would further help us to uncover smishing *campaigns* and later *campaign-operations* that are run and operated by the same bad actors. Since *BERTopic* [10] is performing well for creating topics for a set of documents, we propose to use it as a clustering approach where the same topic themed messages are grouped together in one cluster and eventually get various clusters each labeled with a different theme. Additionally, *BERTopic* provides a comprehensive approach to the clustering requirement, through sentence embeddings instead of word embeddings. This method uses UMAP [16] for dimension reduction that creates a noise cluster of messages separated from the established similar message topic clusters. We provide further

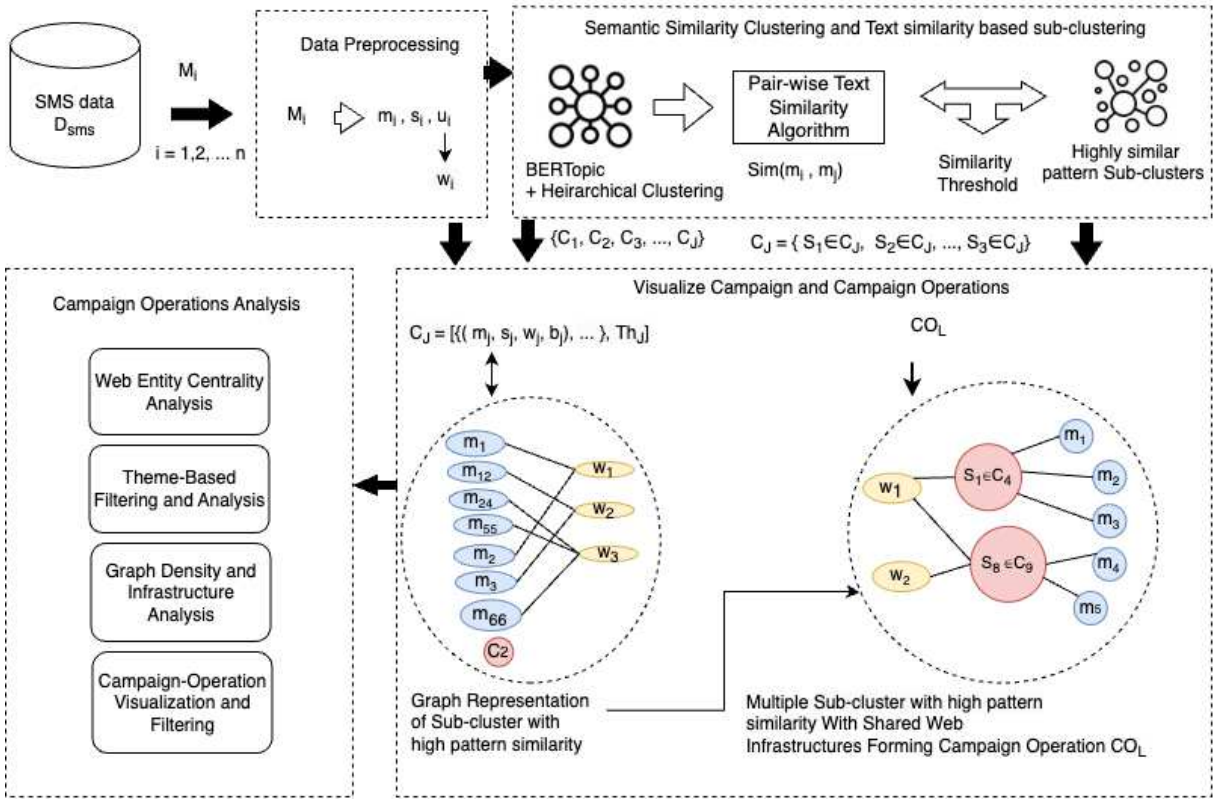

Figure 1: *SmishViz* system overview

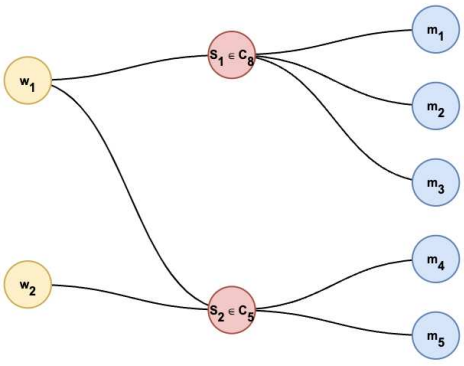

Figure 2: Tripartite graph representation illustrating the connections between sub-cluster nodes, message nodes, and web-entity nodes

additional details on how *BERTopic* approach works and why we have preferred it in the Appendix B.

3.3.1 Semantic Refinement of Clusters Through Hierarchical Clustering. To see the semantic relationships between clusters, we propose to leverage the BERTopic's *visualize_hierarchy()* function [10]. This hierarchical clustering approach can help us merge or split initial clusters for better interpretability to ensure that each final cluster represents a unique topic theme and there are no multiple clusters with same topic theme.

3.3.2 Sub-Cluster Formation Within Clusters. We also propose to group messages within a cluster into multiple sub-groups based on high pattern similarity (i.e., infers as these messages are using a common template). For this, we propose Ratcliff pattern recognition algorithm [21] to investigate messages from the same cluster to identify similar message patterns and create sub-clusters of messages. Appendix C details the functionality of Ratcliff algorithm.

3.4 Campaign-Operation Graph Generation and Storage Module

This module connects multiple cohesive sub-clusters together based on common or shared web entity infrastructures among these sub-clusters to form *campaign-operations* that are safely assumed to be owned and operated by the same bad actors. We envision that the campaign-operations would give defenders insights on coordinated big smish campaigns, their topic themes, target brands, templates, and underneath web infrastructures, which can then be then used to strategize defense against these threats. Here, we propose this as a graph problem, where each of the *campaign-operation* can be represented as a connected tripartite graph network, comprising the following nodes and edges–

- **Node Types:** *Sub-cluster* node [S], *Message* node [m], and *Web-entity* node [w].

- **Edge Types:** There are two edge types. (1) edge between node S_k and node m_i– representing messages within a particular sub-cluster; (2) edges between S_k and w_j– represents web-entities used as infrastructure withing a sub-cluster.

As a whole, these connected graph component would represent a *campaign-operation* run and operated by same bad actor. An illustrative example graph is already presented in figure 2. To achieve this, we collect all the web entities linked to messages residing within a sub-cluster and store this information. Each message can be assigned a corresponding sub-cluster ID with its parent cluster-ID and linked to its associated web entities. This ensures that campaign graphs are built with both cluster-level and sub-cluster-level precision. If a new message is entirely novel and uses a completely new domain infrastructure, it will not fit into any existing cluster or campaign graph. While such cases are more challenging to detect, they impose significant cost and effort on attackers to evade detection. Next, we generate a JSON file to encode and store the graph data structure where each web-entity acts as a root key, linking the sub-clusters in a hierarchical format. An example JSON file structure is highlighted in Appendix D. These JSON files are then imported to the visualization tool for visualizing the campaign-operation graphs swiftly.

3.5 Campaign-Operation Graph Visualization and Analysis Module

One of the primary goals of *SmishViz* system is to uncover bigger smishing campaigns and *campaign-operations* those are otherwise not exposed. To visually analyze the relationships between different message groups with their connected web infrastructures, we propose graph-based visualization with state-of-the art visualization tool, such as *D3.js* [5] to monitor and analyze campaign-operations. We find *D3.js* visualization tool is highly compatible for importing JSON format graph data and can be easily integrated with any web application and web services. Additionally, for gaining more insights on campaign-operations, defenders (i.e., analyst) can explore the following elements from the graph visualizations–

- Central *web-entities* will be determined as highly connected web-entities that are shared across multiple sub-clusters and/or clusters.
- The size of campaign-operation will be visible, such that it includes total number of sub-clusters and number of messages in each of the sub-clusters.

The visualizations tool we use should also provide zoom-in/zoom-out capabilities on specific sub-clusters or web entities to examine many connected nodes effectively if too many nodes are present in one campaign-operation graph. We can also filter the data by campaign topic themes to focus on specific thematic clusters as all our clusters are labeled with specific topic themes. Moreover, during the graph analysis with *SmishViz*, analysts can use metrics such as *degree centrality*, *graph density*, and *connected components* to aid their analysis. To illustrate, by identifying *degree centrality*, analyst can find web entities or sub-clusters that attackers rely on more, which can be critical for defensive strategies. Again, measuring *graph density* also helps to realize the overall cohesion of a campaign-operation, which show how strongly the components are connected and provides insight into the campaign's structure. Finally, *connected components* metric can detect separate campaign-operations which allow defenders to isolate and analyze distinct smishing campaigns. By analyzing these metrics, defenders can prioritize their investigation and mitigation efforts.

4 Experimental Case Study: SmishViz with SmishTank

4.1 SmishTank Data Collection and Pre-processing

For the case study, we have collected a snapshot of *Smishtank* SMS dataset [32] denoted as D_{sms} where $|D_{sms}| = 1,062$. This dataset provides us with a good resource for smishing messages which is ideal for analyzing text patterns, including impersonation of brands, topic themes, and usage of web entities. The smishtank dataset has revealed the following key characteristics–

- **URLs in messages:** 88.6% of these messages contained URLs, many exhibiting suspicious patterns, such as random characters, or unusual domain names.
- **Brand impersonation:** 65.91% of messages included identifiable brand names. A total of 182 unique brands were impersonated, containing industries such as financial services (e.g., Bank of America), e-commerce platforms (e.g., Amazon), delivery services (e.g., USPS), and streaming platforms (e.g., Netflix).
- **Web-entity patterns:** The dataset included frequent use of URL shortener services like *bit.ly* and *tinyurl.com*, which attackers are possibly using to obscure malicious domains.

Some high level basic statistics on the smishtank SMS dataset is presented in table 1. To prepare the data as input for our SmishViz system, we conduct data cleaning following the process described in section 3.2.

Table 1: Data Types & Distributions of SMS Contents in *Smishtank* Dataset

Data Types	Counts	Unique Counts	Avg Length	Median Length	Min Length	Max Length
Full-text	1067	1065	296	259	15	1093
Main-Text	1060	1058	182	142	16	946
URL	937	861	30	26	4	335
FQDN	928	772	16	16	4	43

4.2 Semantic Clustering of Smishtank Messages

We analyze smishtank dataset to create clusters based on semantic similarity in terms of their meanings and topic themes. These themes may include specific topics such as streaming service theme, account verification theme, delivery theme, promotional offer theme, and so on. At first, we apply BERTopic [10] to group messages based on their contextual semantic sentence embeddings, which resulted in 31 message groups with nuanced meanings and distinct thematic patterns. For further refinement of the clusters, we apply Hierarchical clustering using BERTopic's *visualize_hierarchy()* function to eventually end up with several distinct clusters and an outlier group containing noisy or irrelevant messages. Figure 3 highlights the hierarchical clustering with BERTopic and semantic distances between the clusters that are used for cluster refinement. The cluster

Table 2: Cluster-ID and Corresponding Cluster Size

Cluster ID	Size (number of messages within each cluster)	Unique Domain Count
$C_{outlier}$	168	117
C_1	88	50
C_2	223	137
C_3	88	23
C_4	71	66
C_5	73	69
C_6	62	59
C_7	44	36
C_8	44	23
C_9	30	24
C_{10}	171	128

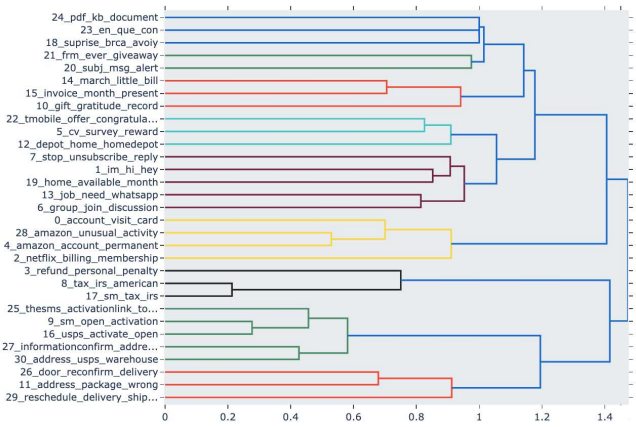

Figure 3: Hierarchical clustering results with BERTopic (where x-axis represents the semantic distance between topics, and y-axis represents 31 message groups resulting from BERTopic prior to cluster merging)

refinement process helps us to merge multiple groups of messages that has very similar topic themes to eventually find unique topic clusters. For example, after the generic BERTopic cluster we have two different message groups that revolving around USPS delivery scams), which then merged to form one cluster for USPS delivery theme. This approach revealed 11 distinct clusters with different sizes (ranging from 223 to 30 messages) as shown in table 2. Figure 5 shows the word distributions by each cluster (topic).

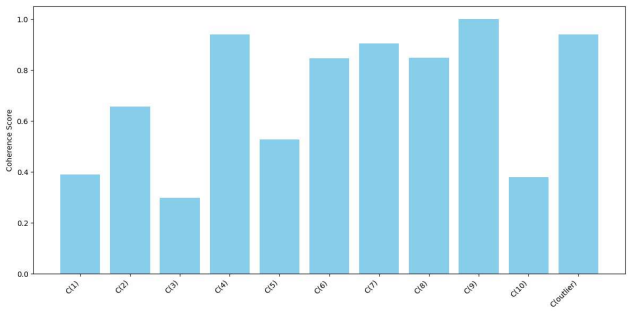

Figure 4: Coherence score of topics for each cluster

This clustering approach achieved strong thematic cohesion, with an overall topic coherence score of 0.702, as calculated using the *Gensim* Python library [22]. Figure 4 presents the topic-wise coherence score for each cluster. To label the clusters systematically with topic themes, we employed a multi-step process:

- **[S1]** Automated topic suggestions: BERTopic generated the top five representative topics for each cluster based on semantic analysis. For instance, Cluster C_5, focused on USPS delivery scams, suggested topics keyword like 'link', 'address', 'please', 'open, 'usps'.
- **[S2]** Word cloud generation: Word clouds provide visualization of main keywords within a cluster. For cluster C_5, terms like 'Delivered', 'arrived', 'USPS', ', 'link', and 'address' are prominently featured in the word cloud. Figure 6 shows an example world cloud for cluster C_5 to reflect its theme of USPS delivery scams.
- **[S3]** Consensus-Based label selection: Combining BERTopic's suggestions, word clouds, and representative messages, we have identified cohesive theme topics for each cluster to assign their theme label.

By employing semantic clustering methods and a systematic labeling approach, we effectively **answered RQ1** by identifying and grouping messages with high similarity together which share a common topic theme. The topic theme label for each of the clusters is presented in Table 3. Moreover, as per our sub-cluster formation process, we apply Ratcliff pattern recognition algorithm into each of the 11 clusters (including $C_{outlier}$). The distribution of sub-clusters within these 11 message clusters are also highlighted in table 3, which effectively **answers RQ2**.

Table 3: Corresponding Cluster Theme Labels and Distribution of Sub-clusters Within Each Cluster

Cluster ID	Unique Sub-Cluster Count	Cluster Topic Theme
C_1	21	Account Suspension and Billing Scams
C_2	121	Social Invitation and Message Reply Scams
C_3	16	IRS Tax Refund Scams
C_4	28	Survey-based Reward Scams
C_5	6	USPS Delivery Scams
C_6	12	Gift Rewards and Fake Billing Scams
C_7	15	User Data Collection Scams
C_8	8	Malicious Document and PDF Scams
C_9	12	Fake Giveaways Scams
C_{10}	145	Account Verification and Scams
$C_{outlier}$	135	Outlier

4.3 Campaign-Operation Graph Generation for Smishtank

We analyze Smishtank dataset to uncover various *campaign-operations* by generating campaign-operation graphs. We use sub-clusters as nodes in our tripartite graph to reveal campaign operations by examining shared web entities among the sub-clusters. To generate these graphs, we first extract all URLs from the messages within each sub-cluster and identify the corresponding web entities (e.g., domain names, full URL, or hostname). These web entities serve as the root nodes of our tripartite graph structure.

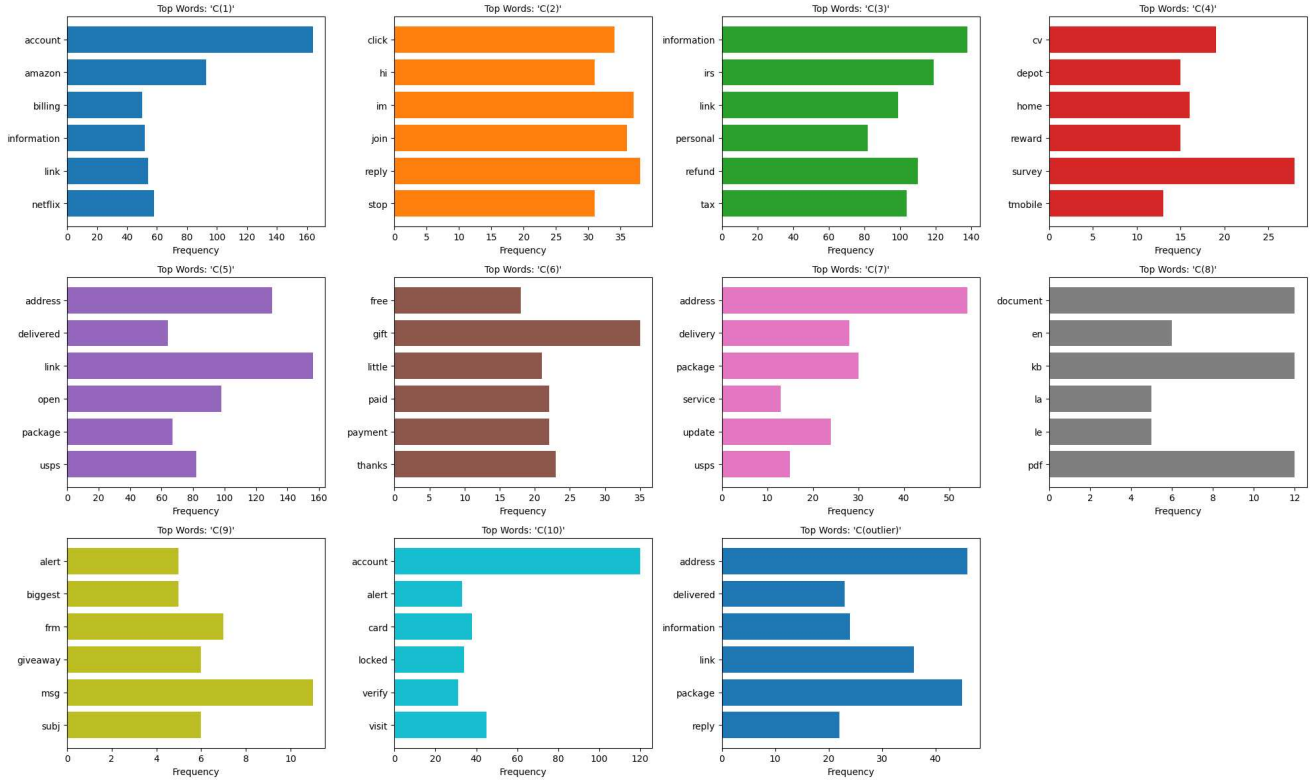

Figure 5: Clustering results based on BERTopic (word distributions within each clusters)

4.3.1 Evidence of Same-Origin Campaign-Operations. Our findings show that several sub-clusters from either same or different parent clusters have contained groups of messages sharing common or similar web entities, indicating they are operated by the same bad actor. For example, if we look deeper in cluster C_3, we find-

- '*direct-capitals.com*' website belongs to both $S_{11} \in C_3$ and $S_{12} \in C3$.
- '*direct-paying.com*' website belongs to both $S_2 \in C_3$ and $S_3 \in C_3$.

Figure 6: Word cloud for cluster C_5 showing keywords that reflect the cluster's theme of USPS delivery scams

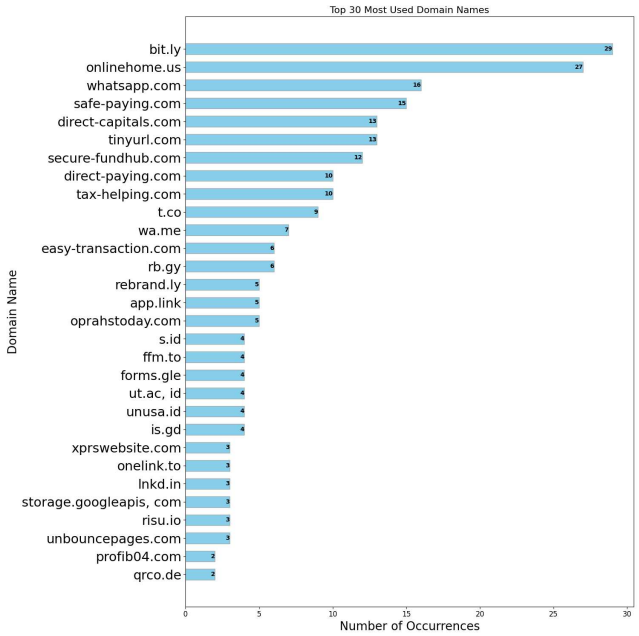

Figure 7: Top domain names by number of unique URLs within smishtank.com

- '*secure-fundhub.com*' website belongs to sub-clusters S_2, S_{11}, S_{13}, and S_{16} within C_3, which is also consistently used for targeting financial scams.

This infrastructure reuse attack tactics suggests coordination by the same bad actor(s) to run and operate smishing campaigns. We also observe examples of campaign graphs within the smishtank dataset which connects multiple sub-clusters such as sub-clusters $S_{11} \in C_3$ and $S_{13} \in C_3$ with common web-entity '*easy-transaction.com*' as reported in Appendix E. Attackers also frequently used URL shortener services such as *bit.ly*, *tinyurl.com*, and *rebrand.ly* across clusters to obscure the actual domains and evade detection mechanisms [7] as the distribution of top frequent domains across clusters are presented in table 4. The above analysis have **answered RQ3** as we have already found campaign-operation connecting multiple sub-clusters.

Table 4: Top Domains Reused Among Multiple Clusters

Domain	Appeared in Number of Clusters
bit.ly	9
tinyurl.com	7
rebrand.ly	4
rb.gy	4
app.link	3
unbouncepages.com.	3
tax-helping.com	3
wa.me	3
whatsapp.com	2

4.3.2 Similar URLs as Evidence of Shared Infrastructure: In addition to exact web-entity matches, we identified cases where attackers registered visually impersonating domain names (*VIDNs*) designed to resemble legitimate domains as suggested by existing literature [29]. These domains often include slight variations, such as character substitutions (e.g., replacing 'o' with '0' or 'l' with '1'), additional characters, or minor alterations to domain structure to make them indistinguishable to a casual observer. These domains which are frequently registered through the same registrar organization during a short period, suggest a coordinated effort to maintain consistency in their campaign operations while diversifying their infrastructure. This tactic enables attackers to keep hiding from domain-based detection systems while continuing to target users through domains that appear legitimate. By using *VIDNs*, attackers expand their access in sub-clusters and create a robust infrastructure to support their malicious smish campaigns.

To illustrate, URLs '*us.ps.track-pack-add.com*' and '*usps.track-pkg.com*' are both observed in a USPS delivery themed SMS phishing campaign. These web-entities exploit the similarity in their appearance to legitimate USPS-related URLs, using slight variations such as subdomain string *us.ps.* and hyphenation *track-pack-add* to deceive users while maintaining thematic consistency. Upon analyzing their registration details, we have found that both domains been registered through the same registrar organization within a two-day period as listed in table 5, strongly indicating a deliberate and coordinated effort for a same-origin campaign. Additionally, the domains were linked to similar smishing messages instructing recipients to track their packages by clicking on the provided URLs, redirecting them to malicious pages. These examples further highlight a coordinated effort by bad actors to prevent detection by registering multiple similar domains, that support our hypothesis of bigger campaign-operations.

Table 5: Registration Details of VIDNs Used in USPS-themed Campaigns

URL	Registrar	Registration Date
us.ps.track-pack-add.com	DNSPod, Inc.	6/14/2023
usps.track-pkg.com	DNSPod, Inc.	6/12/2023

4.4 Campaign Graph Visualization and Analysis of Smishtank Data

In SmishViz, we show that effective visualization of graphs can reveal certain campaign-operation information that can enable defenders with that capability to visualize and analyze graphs and understand various smish campaigns, campaign-operations, campaign size, and campaign thematic topics. It can also aid analysts to find usage of common message templates and reuse of infrastructures, so that they can take further action to dismantle these campaign level attacks. Figure 8 provides a real-world connected tripartite graph to showcase one of the identified campaign-operations within our selected SMS phishing dataset where 6 different web-entities are re-used among 7 different sub-clusters representing 66 different text messages. It is also evident by investigating the associated domain name URLs that the attackers are trying to imitate IRS and tax themed scams for this campaign-operation.

4.4.1 Visualizing Reuses of Domains with Different Messages. We want to understand the distribution of re-usage of web entities across the `smishtank` dataset within different message sub-clusters. Figure 7 present the top 30 domain names, and report their frequency counts of appearing into different messages as unique URLs. Moreover, table 2 shows the unique domain counts for each of the 11 clusters where we observe that clusters have often use a variety of unique domains to carry out their attacks, so that they can not be detected only by domain name based blocking. Another interesting insights is the usage of WhatsApp owned domain names like *whatsapp.com* and *wa.me*, which are used for creating direct chat link to initiate a chat conversation. We have also observed the usage of *app.link*, which is a unique mobile URL that takes users to a specific in-app page if they already have installed that app on their phone, which reveals attackers' tactics of leading users to desired mobile apps. This analysis indicates that visualizing these campaigns as connected graphs and analyzing the web entities may aid the defenders in designing and blocking larger smishing campaigns effectively.

4.4.2 Visualizing Connections Between Thematic Clusters and the Outlier Cluster. Despite thematic clusters showing strong internal cluster consistency in their web-entity usage, meaning no URL is shared between sub-clusters of different thematic parent clusters. However, upon examining the outlier cluster, we extract cases where web-entities from some sub-clusters within a specific thematic cluster also re-used within one of the sub-clusters of our outlier cluster. These overlaps suggest the presence of a potential

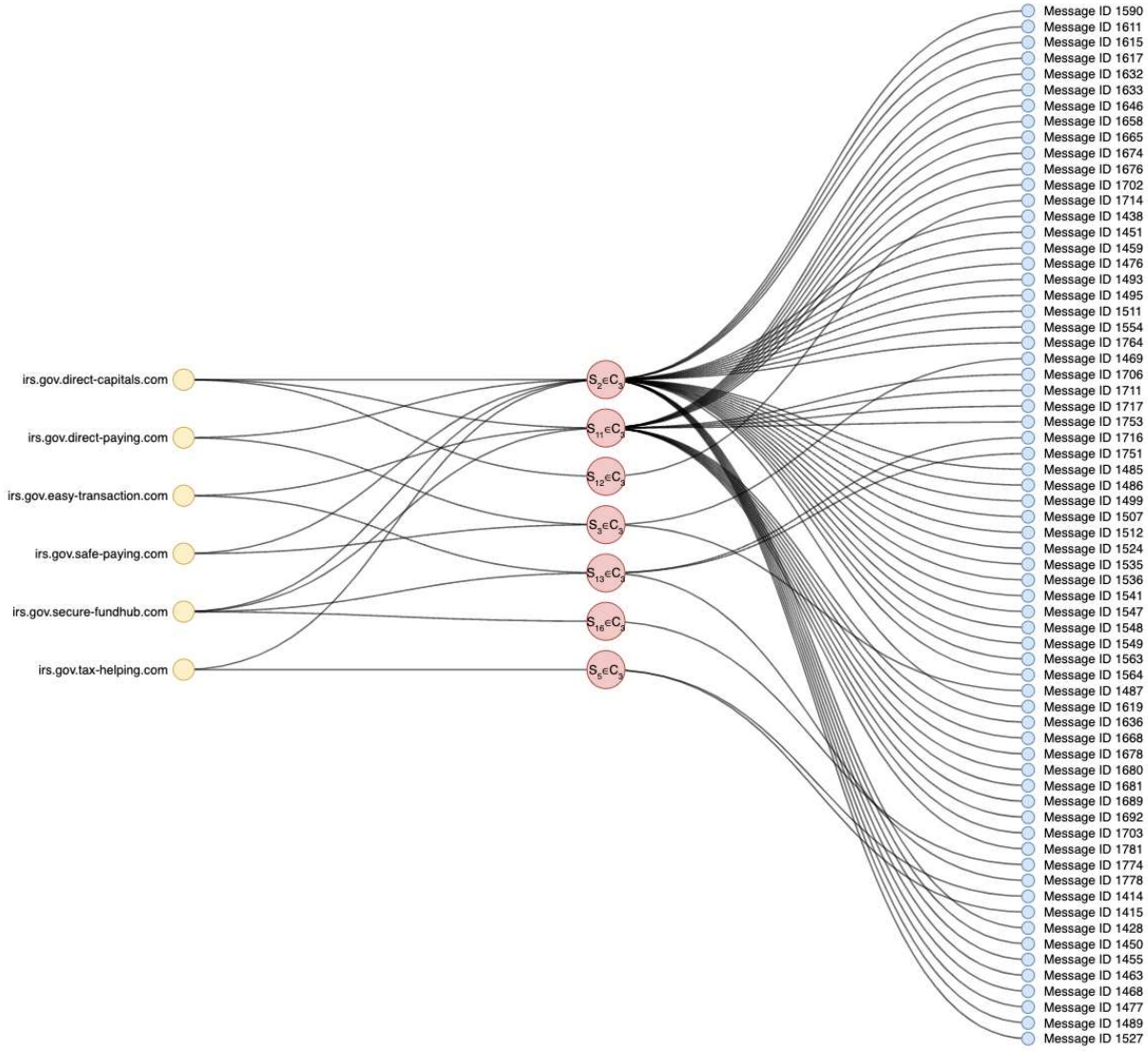

Figure 8: One *campaign-operation* tri-partite graph containing 6 different domain name URLs, 7 message sub-clusters and 66 unique message texts with primarily *IRS and tax fraud* smishing attempts

larger campaign-operation that spans across both thematic and outlier clusters. In Figure 9, we present one of the formed connected tri-partite graphs based on connections between sub-clusters from thematic cluster C_5 and sub-clusters from outlier cluster $C_{Outlier}$. This also suggests that in a live SmishViz system, this $C_{Outlier}$ cluster can host some other future thematic clusters, which do not have enough datapoints to form an independent cluster for now. This observation indicates that attackers might be conducting operations that are not yet fully represented in topic thematic clusters but are connected through the infrastructure present in the outlier cluster.

Example of campaign-operation from regular thematic cluster with outlier cluster: we provide couple of examples from our observation which demonstrate the possibility of larger, evolving campaigns that connect thematic clusters to messages within the outlier cluster through shared web infrastructure.

- Sub-cluster $S_2 \in C_5$ (USPS Delivery scams) shares the domain *uspsusa-us.com* with sub-cluster $S_{102} \in C_{Outlier}$.
- Sub-cluster C_4 (Survey-based Reward scams) shares IP address '*107.175.219.12*' with another sub-cluster within the outlier cluster.

With the above graph-based visualizations, we effectively **answered RQ4** by demonstrating how SmishViz aids defenders in uncovering and understanding interconnected smishing campaigns.

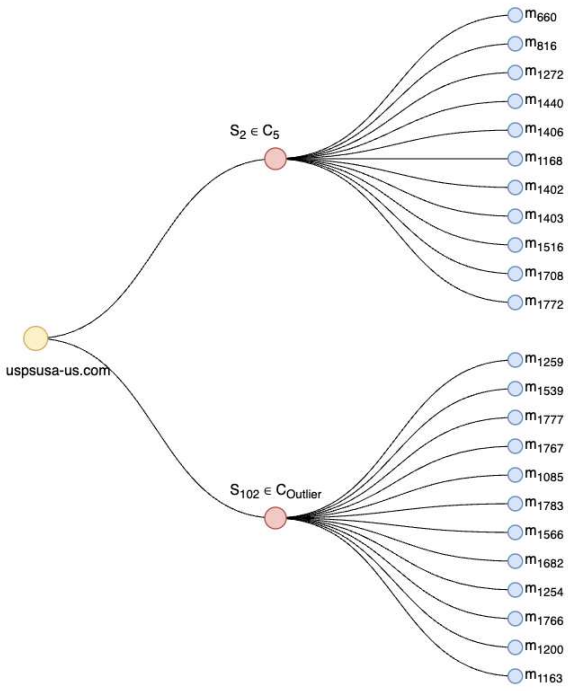

Figure 9: Graphs-based representation showing connections between sub-clusters from thematic clusters and sub-clusters from outliers

5 Limitations

Clustering of messages and connecting them under campaign-operations plays a vital role in understanding the tactics and the current landscape of smishing threats. By identifying message topic themes, similarities in text patterns, and shared infrastructures, analysts can prioritize their defense efforts against bigger and coordinated operations and proactively detect new smishing attacks. Although the *SmishViz* system proposed in this paper offers insightful information, the paper has the following limitations– First, due to the dynamic threat landscape for smishing, the analysis on *SmishTank* may not capture all diverse attacks due to limited data. Second, smishtank community is still not big enough or active enough to vote on validating the smishing texts, which infer that smishtank can not be counted as ground-truth labeled data for smishing. Third, similarity-based message categorization may need further evaluation in different dataset, where we can capture more similar themed messages under the same message clusters. Fourth, the system's reliance on outlier clusters to handle new or unstructured messages highlights a gap in detecting new themed smishing campaigns. Introducing incremental clustering or anomaly detection methods in the future could enhance adaptability to evolving threats. Fifth, the scalability of the tripartite graph-based visualization may pose challenges as datasets grow larger, potentially impacting efficiency and clarity, which can be addressed in the future by exploring graph simplification techniques, hierarchical clustering, or dynamic filtering mechanisms to maintain the system's usability and interpretability at scale. Sixth, our proposed *SmishViz* system is not evaluated for usability metrics by real analysts, which we can explore in future studies where user-study can be conducted to evaluate and improve the functionalities of SmishViz platform. Seventh, we have not fully highlighted the graph analysis capabilities in this paper, which can be explored further in future work.

6 Conclusion

In summary, we took a close look at smishing campaigns using a real-world dataset from `smishtank.com`. We find patterns, themed topics, and common web infrastructures that can help us identify and characterize smishing campaign-operations. Characterizing the campaign-operations help us understand the attacker's strategies and tactics. While our case study has been scoped for a specific dataset, the methods and techniques can be easily applied to any other datasets from various sources. We also envision to build a live system for tracking smishing campaigns and campaign operations in the wild. The proposed *SmishViz* system should be a public facing live system that anyone can request to integrate with newer dataset which will dynamically generate ongoing campaign-operation graphs for visualization and further analysis. We believe a live system can help defenders building defensive strategies against newer smishing attacks by continuously monitoring ongoing smishing campaigns. The code and relevant resources are shared in the following Github repository to enable the reproducibility of our results– https://github.com/varnicm/SmishViz-Project/.

Acknowledgment

We thank all the reviewers for their insightful feedback. This work is partially supported by the National Science Foundation grants #2230609 and #2416990.

References

[1] Oluwatobi Noah Akande, Oluwadara Gbenle, Oluwakemi Christiana Abikoye, Rasheed Gbenga Jimoh, Hakeem Babalola Akande, Abdullateef O Balogun, and Anuoluwapo Fatokun. 2023. SMSPROTECT: An automatic smishing detection mobile application. *ICT Express* 9, 2 (2023), 168–176.

[2] Ahmed Aleroud and Lina Zhou. 2017. Phishing environments, techniques, and countermeasures: A survey. *Computers & Security* 68 (2017), 160–196.

[3] Jalilifard Amir et al. 2020. Semantic sensitive TF-IDF to determine word relevance in documents. *arXiv preprint arXiv:2001.09896* (2020).

[4] Eric Blancaflor, Mariah Anne Romero, Irish Nacu, and Denver Ryan Golosinda. 2023. A Case Study on Smishing: An Assessment of Threats against Mobile Devices. In *Proceedings of the 2023 9th International Conference on Computer Technology Applications*. 172–178.

[5] Mike Bostock. 2011. D3.js - Data-Driven Documents. https://d3js.org/. Accessed: October 1, 2024.

[6] Federal Trade Commission. [n. d.]. Fraud Reports. https://public.tableau.com/app/profile/federal.trade.commission/viz/FraudReports/. Accessed: April, 2024.

[7] Jaime Devesa, Xabier Cantero, Gonzalo Alvarez, and Pablo G Bringas. 2011. An efficient security solution for dealing with shortened url analysis. In *International Workshop on Security in Information Systems*, Vol. 2. SCITEPRESS, 70–79.

[8] Morgan Edwards, Thomas Morris, Jing Chen, and Jeremiah Still. 2023. SMiShing Attack Vector: Surveying End-User Behavior, Experience, and Knowledge. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 67. SAGE Publications Sage CA: Los Angeles, CA, 1911–1915.

[9] Diksha Goel and Ankit Kumar Jain. 2018. Smishing-classifier: a novel framework for detection of smishing attack in mobile environment. In *Smart and Innovative Trends in Next Generation Computing Technologies: Third International Conference, NGCT 2017, Dehradun, India, October 30-31, 2017, Revised Selected Papers, Part II 3*. Springer, 502–512.

[10] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794* (2022).

[11] Maanak Gupta, Charankumar Akiri, Kshitiz Aryal, Eli Parker, and Lopamudra Praharaj. 2023. From ChatGPT to ThreatGPT: Impact of Generative AI

in Cybersecurity and Privacy. *IEEE Access* 11 (2023), 80218–80245. https://doi.org/10.1109/ACCESS.2023.3300381
[12] Syed Md Mukit Hossain, Jami Ahmed Sumon, Arnab Sen, Md Ishrak Alam, Kazi Md Arifuzzaman Kamal, Hissah Alqahtani, and Iqbal H Sarker. 2022. Spam filtering of mobile sms using cnn–lstm based deep learning model. In *Hybrid Intelligent Systems*. Springer, 106–116.
[13] Akshay Kumar Jain, Brij Bhooshan Gupta, Kavita Kaur, Preeti Bhutani, Wadee Alhalabi, and Ahmed Almomani. 2022. A content and url analysis-based efficient approach to detect smishing sms in intelligent systems. *International Journal of Intelligent Systems* 37, 12 (2022), 11117–11141.
[14] Fujiao Ji, Kiho Lee, Hyungjoon Koo, Wenhao You, Euijin Choo, Hyoungshick Kim, and Doowon Kim. 2024. Evaluating the Effectiveness and Robustness of Visual Similarity-based Phishing Detection Models. *arXiv preprint arXiv:2405.19598* (2024).
[15] Claudia Malzer and Marcus Baum. 2020. A hybrid approach to hierarchical density-based cluster selection. In *2020 IEEE international conference on multisensor fusion and integration for intelligent systems (MFI)*. IEEE, 223–228.
[16] Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).
[17] Sandhya Mishra and Devpriya Soni. 2023. DSmishSMS-A System to Detect Smishing SMS. *Neural Comput & Applic* 35, 7 (2023), 4975–4992. https://link.springer.com/10.1007/s00521-021-06305-y
[18] Aleksandr Nahapetyan, Sathvik Prasad, Kevin Childs, Adam Oest, Yeganeh Ladwig, Alexandros Kapravelos, and Bradley Reaves. 2024. On SMS Phishing Tactics and Infrastructure. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society.
[19] Adam Oest, Yeganeh Safaei, Penghui Zhang, Brad Wardman, Kevin Tyers, Yan Shoshitaishvili, and Adam Doup'e. 2020. PhishTime: Continuous longitudinal measurement of the effectiveness of anti-phishing blacklists. (2020), 379–396.
[20] Md Lutfor Rahman, Daniel Timko, Hamid Wali, and Ajaya Neupane. 2023. Users Really Do Respond To Smishing. In *Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy* (<conf-loc>, <city>Charlotte</city>, <state>NC</state>, <country>USA</country>, </conf-loc>) *(CODASPY '23)*. Association for Computing Machinery, New York, NY, USA, 49–60. https://doi.org/10.1145/3577923.3583640
[21] John W Ratcliff, David E Metzener, et al. 1988. Pattern matching: The gestalt approach. *Dr. Dobb's Journal* 13, 7 (1988), 46.
[22] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, 45–50. http://is.muni.cz/publication/884893/en.
[23] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084
[24] S. Saha Roy, P. Thota, K. Naragam, and S. Nilizadeh. 2024. From Chatbots to Phishbots?: Phishing Scam Generation in Commercial Large Language Models. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 221–221. https://doi.org/10.1109/SP54263.2024.00182
[25] Dawn M Sarno and Jeffrey Black. 2024. Who gets caught in the Web of lies?: Understanding susceptibility to phishing emails, fake news headlines, and scam text messages. *Human Factors* 66, 6 (2024), 1742–1753.
[26] Jae Woo Seo, Jong Sung Lee, Hyunwoo Kim, Joonghwan Lee, Seongwon Han, Jungil Cho, and Choong-Hoon Lee. 2024. On-Device Smishing Classifier Resistant to Text Evasion Attack. *IEEE Access* (2024).
[27] Ashfak Md Shibli and Mir Mehedi A. Pritom. 2024. Poster: Use of LLM-based Generative AI Chatbots for Smishing Attacks and Defenses. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, San Francisco, CA, USA.
[28] Ashfak Md Shibli, Mir Mehedi A. Pritom, and Maanak Gupta. 2024. AbuseGPT: Abuse of Generative AI ChatBots to Create Smishing Campaigns. In *2024 12th International Symposium on Digital Forensics and Security (ISDFS)*. 1–6. https://doi.org/10.1109/ISDFS60797.2024.10527300
[29] Geoffrey Simpson, Tyler Moore, and Richard Clayton. 2020. Ten years of attacks on companies using visual impersonation of domain names. In *2020 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 1–12.
[30] SlashNext. [n. d.]. Slashnext's 2023 state of phishing report. https://www.prnewswire.com/news-releases/slashnexts-2023-state-of-phishing-report-reveals-a-1-265-increase-in-phishing-emails-since-the-launch-of-chatgpt-in-november-2022--signaling-a-new-era-of-cybercrime-fueled-by-generative-ai-301971557.html. https://www.prnewswire.com/ Accessed: January 21, 2024.
[31] Gunikhan Sonowal and KS Kuppusamy. 2018. SmiDCA: an anti-smishing model with machine learning approach. *Comput. J.* 61, 8 (2018), 1143–1157.
[32] Daniel Timko and Muhammad Lutfor Rahman. 2024. Smishing Dataset I: Phishing SMS Dataset from Smishtank. com. *arXiv preprint arXiv:2402.18430* (2024).
[33] Longfei Wu, Xiaojiang Du, and Jie Wu. 2015. Effective defense schemes for phishing attacks on mobile computing platforms. *IEEE Transactions on Vehicular Technology* 65, 8 (2015), 6678–6691.
[34] Ezer Osei Yeboah-Boateng and Priscilla Mateko Amanor. 2014. Phishing, SMiShing & Vishing: an assessment of threats against mobile devices. *Journal of Emerging Trends in Computing and Information Sciences* 5, 4 (2014), 297–307.
[35] Juan Zamora. 2017. Recent advances in high-dimensional clustering for text data. *Claudio Moraga: a passion for multi-valued logic and soft computing* (2017), 323–337.

Appendix

A. Details on Cleaning Raw SMS Texts for Fixing Issues With URL

We exhibit that some of the data in public repository is extracted from images of texts and thus used OCR (Optical Character Recognition) techniques to convert texts from images. As OCR process is not completely reliable, there are often some slight changes to the raw text data to include additional space characters or miss a character. The problem is more severe for URL information in the SMS, as additional characters or adding a space in between URL path may not present the correct URL. To address this challenge, we split the URL into individual characters and removed all permutations of spaces within the URL (including variations with spaces inserted between characters). To ensure that this process does not affect other parts of a message, we ran a test to find the number of characters of each message after cleaning the URL:

- L_{MainText}: Length of the original text (`MainText`).
- L_{URL}: Length of the URL (`Url`), treating missing values as 0.
- L_{Cleaned}: Length of the cleaned message (`CleanedMessage`).

The equation to determine if the lengths approximately match is:

$$\text{LengthMatch} = \begin{cases} True, & \text{if } |L_{\text{MainText}} - L_{\text{URL}} - L_{\text{Cleaned}}| \leq 2, \\ False, & \text{otherwise.} \end{cases}$$

This checks whether the absolute difference between the expected and observed lengths after space removal from the URL is within a threshold of ±2 characters.

B. How Does BERTopic Work and Why Are We Using It?

We used BERTopic for clustering based on semantic similarity to group messages by themes, such as delivery or tax-related content. This method aligns with our goal of clustering based on message meaning. BERTopic utilizes sentence embeddings via the Sentence-BERT (SBERT) framework, capturing the meaning of entire messages rather than individual words. This approach improves clustering performance by leveraging pre-trained language models [23]. Given the large size of our dataset, sentence embeddings result in high-dimensional representations, which can complicate clustering. BERTopic addresses this challenge using UMAP [16], a dimensionality reduction technique that employs Riemannian geometry and algebraic topology. UMAP outperforms alternatives like t-SNE by preserving global structure, enhancing visualization, and offering superior runtime performance. Additionally, BERTopic incorporates HDBSCAN [15], a hierarchical clustering method that explores density variations. Unlike DBSCAN, which relies on a fixed epsilon

threshold, HDBSCAN automatically identifies flat clusters by simplifying complex hierarchies. This adaptability is particularly useful for datasets with variable message densities, enabling effective handling of both dense and sparse regions without classifying sparse regions as noise. Finally, BERTopic assigns a topic label to each cluster, summarizing its central theme. By generating topic-level representations, BERTopic links all messages in a cluster, offering an advantage over traditional methods like *TF-IDF* [3] which rely on individual message features. These components make BERTopic an effective and justified approach for our clustering needs.

```
{
    "name": "cgmfdk.com",
    "children": [
        {
            "name": "C(2)-S24",
            "children": [
                {
                    "name": "Message ID 222",
                    "content": "You got an intimate
                    ↪ invite from Sarah. Read
                    ↪ what she said now"
                }
            ]
        },
        {
            "name": "C(2)-S27",
            "children": [
                {
                    "name": "Message ID 225",
                    "content": "Hey hun, wanna chat
                    ↪ and get to know one
                    ↪ another? Anyways let me
                    ↪ know. I'll be here:"
                }
            ]
        }
    ]
}
```

Figure 10: JSON structure of smishing campaign clusters.

C. How Do Ratcliff Pattern Recognition Works?

The Ratcliff string matching algorithm calculates the similarity metric between two strings as twice the number of matching (overlapping) characters between the two strings divided by the total number of characters in the two strings. So, similarity of two messages m_i and m_j can be calculated as $\texttt{Sim}(m_i, m_j) = \frac{2\times\texttt{Matching}_{(i,j)}}{|m_i|+|m_j|}$, where $\texttt{Matching}_{(i,j)}$ presents the number of overlapping characters between messages m_i and m_j.

D. An Example JSON File Structure To Store Campaign-Graph

Figure 10 highlights the JSON structure for storing graph information to join multiple sub-clusters with their corresponding messages if they are using common web-entity infrastructures.

E. Example of Connected Components Via Common Web Entity (Campaigns)

Figure 11 shows an example of two separate sub-clusters (meaning they have differences in message templates) $S_{11} \in C_3$ and $S_{13} \in C_3$ connected via one common web-entity infrastructure such as the domain name `easy-transaction.com`.

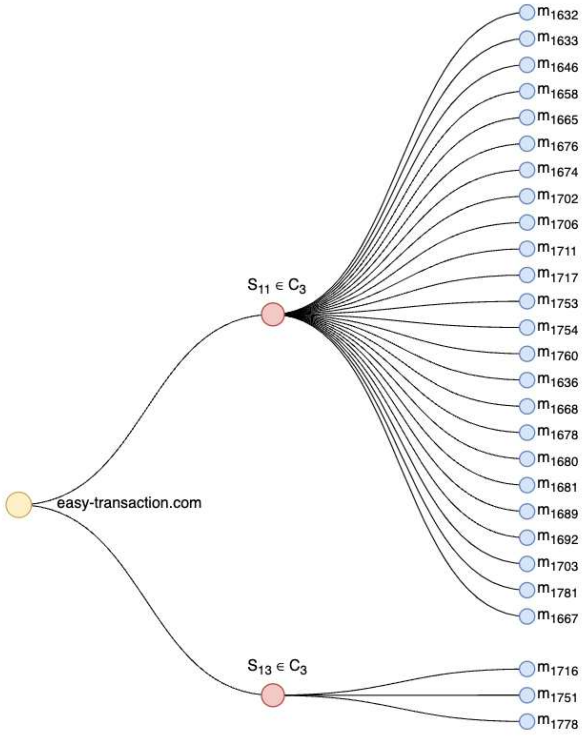

Figure 11: Evidence of same-origin campaign-operation