Expert Systems With Applications 299 (2026) 130111

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

PassREfinder-FL: Privacy-preserving credential stuffing risk prediction via graph-based federated learning for representing password reuse between websites ⋆

Jaehan Kim [a], Minkyoo Song [a], Minjae Seo [b], Youngjin Jin [a], Seungwon Shin [a], Jinwoo Kim [c,*]

[a] KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon, 34141, Republic of Korea
[b] ETRI, 218 Gajeong-ro, Yuseong-gu, Daejeon, 34129, Republic of Korea
[c] Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul, 01897, Republic of Korea

ARTICLE INFO

Keywords:
Password authentication
Credential stuffing
Graph neural network
Federated learning

ABSTRACT

Credential stuffing attacks have caused significant harm to online users who frequently reuse passwords across multiple websites. While prior research has attempted to detect users with reused passwords or identify malicious login attempts, existing methods often compromise usability by restricting password creation or website access, and their reliance on complex account-sharing mechanisms hinders real-world deployment. To address these limitations, we propose PassREfinder-FL, a novel framework that predicts credential stuffing risks across websites. We introduce the concept of password reuse relations—defined as the likelihood of users reusing passwords between websites—and represent them as edges in a website graph. Using graph neural networks (GNNs), we perform a link prediction task to assess credential reuse risk between sites. Our approach scales to a large number of arbitrary websites by incorporating public website information and linking newly observed websites as nodes in the graph. To preserve user privacy, we extend PassREfinder-FL with a federated learning (FL) approach that eliminates the need to share user sensitive information across administrators. Evaluation on a real-world dataset of 360 million breached accounts from 22,378 websites shows that PassREfinder-FL achieves an F1-score of 0.9153 in the FL setting. We further validate that our FL-based GNN achieves a 4–11 % performance improvement over other state-of-the-art GNN models through an ablation study. Finally, we demonstrate that the predicted results can be used to quantify password reuse likelihood as actionable risk scores. Our implementation is available at https://github.com/jaehanwork/PassREfinder-FL.

1. Introduction

Credential stuffing—*the use of a breached or stolen account from an online service to sign in to other online services*—is one of the most severe cyberattacks (Verizon, 2022). Due to a surge of massive credential data breaches, individual users and even various industries have suffered from the threat of credential stuffing. For example, retail industries have been imposed losses of 6 billion dollars per year (Shape Security, 2018). The devastating impact of credential stuffing has fundamentally resulted from the use of the same password by users across different online services. The problem of *password reuse* has been repeatedly claimed by security researchers (Das et al., 2014; Pearman et al., 2017; Thomas et al., 2017; Wang et al., 2018; Wash et al., 2016). Even though administrators have warned about the reuse of passwords to users, it was not sufficient to raise awareness for users to change their passwords (Golla et al., 2018). Moreover, usability issues in current password managers remain, which can lead users to reuse the same passwords (Hutchinson et al., 2024).

To protect users from the risk of credential stuffing, several third-party services called compromised credential checking (C3) services (Li et al., 2019) have been developed, such as Have I Been Pwned (HIBP) (Have i been pwned?, 2022), Google Password Checkup (Thomas et al., 2019). The C3 services alert users to reset passwords when the same passwords are reported from credential data breaches of other websites. Unfortunately, these services are limited to detecting the reuse of passwords disclosed by already known credential

⋆ This work extends our previous study, PassREfinder (Kim et al., 2024), which was presented at IEEE S&P 2024.

* Corresponding author.
E-mail addresses: jaehan@kaist.ac.kr (J. Kim), minkyoo9@kaist.ac.kr (M. Song), ms4060@etri.re.kr (M. Seo), ijinjin@kaist.ac.kr (Y. Jin), claude@kaist.ac.kr (S. Shin), jinwookim@kw.ac.kr (J. Kim).

https://doi.org/10.1016/j.eswa.2025.130111
Received 5 April 2025; Received in revised form 2 September 2025; Accepted 17 October 2025
Available online 22 October 2025

data breaches. As an alternative, a method of coordinating websites was proposed to detect the creation of the same password (Wang & Reiter, 2019) or suspicious logins for credential stuffing attacks (Wang & Reiter, 2020) by sharing usernames and passwords under privacy-preserving protocols. However, these approaches degrade the usability of passwords through direct interruption when choosing a password or denial of access to the websites due to the detection of false positives. More critically, the burdensome encryption methods preclude the deployment of their protocols at scale (Pal et al., 2022), and the requirement of collecting usernames to map each username to websites discourages the participation of websites.

As such, existing approaches face clear limitations, including reliance on known data, exposure of sensitive information, and limited scalability. To address these issues, we propose PassREfinder-FL, a privacy-preserving framework that predicts the risk of credential stuffing. Prior studies (Gaw & Felten, 2006; Pearman et al., 2017; Stobert & Biddle, 2014; Thomas et al., 2019; Wang et al., 2018; Wash et al., 2016) have shown that website characteristics are closely linked to users' password reuse behavior—for example, users often reuse passwords across websites in the same category or based on perceived security levels. Motivated by these findings, we analyze website features to capture their inherent influence on reuse behavior and predict the likelihood of credential stuffing between websites. To this end, we define a *password reuse relation* as a relationship between websites where users are highly likely to reuse passwords, and our framework predicts the existence of such relations.

To accomplish this goal, however, *three* technical challenges must be addressed. First, password reuse relations across a large number of websites are inherently complex, as a user's password choice for one website is influenced by passwords used on others. To capture this, PassREfinder-FL models password reuse as edges in a graph of website nodes, leveraging graph neural networks (GNNs) to aggregate and propagate neighborhood influences. Second, the heterogeneous structure of website features and their varying impact on password reuse make them difficult to model with standard neural networks. To overcome this, PassREfinder-FL incorporates multi-modal learning (Ngiam et al., 2011) and attention mechanisms (Hori et al., 2017; Veličković et al., 2017), enabling the model to process each feature type independently and weigh their importance. Lastly, predicting relations involving unknown or new websites typically requires sharing password reuse information or user private information across potentially untrustworthy administrators, raising privacy concerns (Kim et al., 2024). PassREfinder-FL mitigates this by employing a federated learning (FL) approach, allowing participants to share only model gradients or weights rather than sensitive information, thus preserving user privacy.

We evaluate PassREfinder-FL through extensive experiments on a large-scale dataset comprising credential data breaches involving 360 million accounts across 22,378 websites. Our results demonstrate that the proposed GNN model under the FL setting effectively predicts password reuse relations between websites. In particular, it significantly outperforms our original framework, PassREfinder (Kim et al., 2024) and its variants (i.e., with different hidden relation policies), achieving an F1-score of 0.9153. We also show that our GNN design yields performance improvements of up to 4–11 % compared to other state-of-the-art GNN models. We attribute the notable prediction performance of PassREfinder-FL to its FL-based approach, which effectively leverages the password reuse information across multiple administrators' websites. Furthermore, its privacy-preserving design enables administrators to confidently deploy the framework in real-world scenarios.

In the ablation study, we assess the impact of each GNN modeling design choice in PassREfinder-FL. While the neighborhood attention method contributes less significantly under the FL setting—compared to its effectiveness in our original framework, PassREfinder (Kim et al., 2024), the multi-modality feature design consistently enhances prediction performance. Furthermore, we validate that the predicted probabilities directly reflect the expected rates of password reuse between websites, enabling their interpretation as credential stuffing risk scores. Overall, the results demonstrate that our model effectively learns edge representations for predicting password reuse relations across a large-scale graph of website nodes.

Contributions. Our contributions are summarized as follows:

- We introduce the concept of password reuse relation to present the tendency of password reuse between websites and formalize the password reuse relation by leveraging a graph structure.
- We propose a new framework, PassREfinder-FL, to proactively predict the credential stuffing risk among websites based on graph neural networks, developing strategic techniques that can address technical challenges in the risk prediction task.
- We address the privacy concerns inherent in PassREfinder, a credential stuffing risk prediction framework, by adopting an FL-based approach.
- We evaluate our framework on a large-scale credential dataset, achieving an F1-score of 0.9153 and outperforming PassREfinder while ensuring user privacy.
- We show that our prediction results can be used to quantitatively measure credential stuffing risk scores among websites.
- We provide three practical application scenarios of PassREfinder-FL in the perspective of administrators to mitigate credential stuffing.

2. Background

In this section, we present the background knowledge and related studies.

Credential stuffing detection. Credential stuffing is recognized as a serious threat, and researchers have striven to prevent it by detecting password reuse. One of the most common techniques is a compromised credential checking (C3) service (Li et al., 2019; Pal et al., 2022; Thomas et al., 2019), such as Have I Been Pwned (HIBP) (Have i been pwned?, 2022). C3 services provide breach-alerts when the same passwords are breached from other websites. However, existing C3 services can only detect reused passwords discovered from already disclosed data breaches. The discovery of data breach incidents took an average of 207 days, and containment lasted 70 days in 2022 (IBM Security, 2022). Hence, C3 services cannot provide timely coping with exploiting new accounts from undisclosed data breaches.

As another type of credential stuffing detection, Wang and Reiter (2019) proposed a framework for detecting users who create the same passwords in different websites by coordinating the websites and sharing the information of usernames and passwords under a variant of private set-membership-test (PMT) protocol. Their following work (Wang & Reiter, 2020) adopted the previous PMT protocol but modified its design for directly detecting login trials for credential stuffing attacks. Despite the promise of these approaches, they still have critical limitations. The detection frameworks extremely degrade the usability of passwords by immediately interrupting a user's password creation or denying website access due to false positives in detection. Moreover, the complicated encryption mechanisms make it difficult to deploy their protocols at scale (Pal et al., 2022). Their username-website mapping system is also problematic because it requires all usernames from the websites, which discourages participation in the website coordination in practice.

Password reuse analysis. To prevent the devastating impact of credential stuffing, many studies on the risk of password reuse have been conducted. Specifically, analyzing features affecting password choice and reuse has been one of the main topics (AlSabah et al., 2018; Gaw & Felten, 2006; Mayer et al., 2017; Moh et al., 2024; Pearman et al., 2017; Sahin & Li, 2021; Stobert & Biddle, 2014; Thomas et al., 2019; Wang et al., 2018; Wash et al., 2016). For example, AlSabah et al. (2018) and Mayer et al. (2017) claimed that users significantly considered their languages and websites' password composition policies to create their passwords. Pearman et al. (2017) presented a user study with 154 participants and discovered that the reuse of a password was influenced

by its characteristics (e.g., password length and strength). In addition, they showed that the category of website performed a significant role in determining the reuse of passwords, as shown in several other works (Gaw & Felten, 2006; Thomas et al., 2019; Wang et al., 2018). Wash et al. (2016) warned that university students were likely to reuse their passwords in other university-related services. Stobert and Biddle (2014) showed that some users carefully consider the gap between the security aspects of two websites when they reused their passwords across websites. Moh et al. (2024) found that a user often reuses the same password across personal and shared accounts, such as those shared with family members.

Graph neural networks (GNNs). Graph neural networks (GNNs) are neural networks for processing graph data, and have been widely utilized to represent relations between objects. As one of the most common types of GNN, graph convolutional neural network (GCN) (Kipf & Welling, 2016) performs convolution operations in graphs based on message passing and neighborhood aggregation to represent the hidden features of nodes. GraphSAGE (Hamilton et al., 2017) is a variant of GCN for inductive representation of a large-sized graph by reflecting the self-embedding separately and sampling the neighborhood for aggregation. GAT (Veličković et al., 2017) adopts a self-attentional aggregation approach to specify different weights to different nodes. GNN variants have achieved notable performance for various learning tasks in a wide range of security research domains (Cui et al., 2022; Kim et al., 2023; Lee et al., 2024; Liu et al., 2021; Seo et al., 2024; Yang et al., 2022).

Federated learning (FL) on graph. Federated learning (FL) has recently emerged as a promising paradigm for training neural networks on decentralized data, addressing privacy concerns in the collaborative development of deep learning models across multiple entities (Konečný et al., 2016a,b). In graph-based FL (He et al., 2021), multiple clients independently train GNNs using their local subgraphs. Each client computes local updates and transmits the resulting gradients or model weights to a central server or neighboring clients. These updates are then aggregated to refine a global GNN model, subsequently redistributed to the clients, thus preserving data privacy by avoiding direct sharing of raw graph data. Scardapane et al. (2020) introduce a fully distributed framework for training GCNs, enabling distributed inference and optimization across clients communicating over a sparse network without a central server. Wu et al. (2021) propose FedGNN, an FL framework specifically designed for recommendation systems, incorporating local differential privacy and pseudo-interaction sampling to securely model user-item interactions. Recent research efforts have focused on improving communication efficiency within federated GNN training frameworks, aiming to reduce communication overhead and enhance scalability (Yao et al., 2023; Zhang et al., 2021). Recently, graph federated learning has been widely applied to security domains such as anomaly detection (Kong et al., 2024; Mao et al., 2025), fraud detection (Tang & Liang, 2024), and malware detection (Amjath et al., 2025). In contrast, PassREFinder-FL introduces a novel problem formulation for the application of graph federated learning to protect user credentials.

3. Motivation

Here, we motivate the need for our work by introducing our previous study (Kim et al., 2024), which highlights the limitations of existing approaches and lays the foundation for our proposed solution.

Credential stuffing risk prediction. As previously discussed, detecting credential stuffing is a reactive measure that fails to prevent attacks before they occur. A more proactive approach is to predict a website's risk level by learning patterns such as website-specific characteristics and password reuse behaviors across websites. PassREFinder (Kim et al., 2024) introduces a state-of-the-art method in this domain, leveraging GNNs to model users' password reuse tendency. It frames the task as a link prediction problem—representing password reuse as a *password reuse relation* and estimating whether a password used on one website is likely to be reused on another, as illustrated in Fig. 1. However, directly

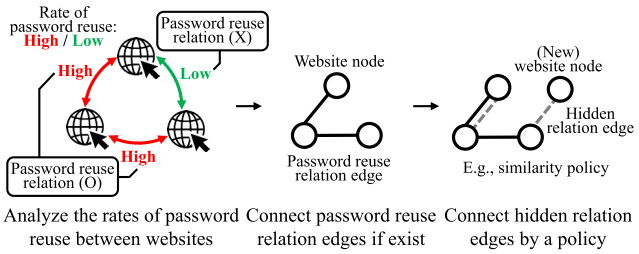


Fig. 1. Graph representation of password reuse relations.

exchanging password reuse information for its training or inference process raises significant ethical and privacy concerns. To address this, PassREFinder introduced *hidden relation*, a complementary edge connected by a pre-defined policy, propagating implicit information about password reuse tendencies to effectively learn credential stuffing risks across websites managed by different administrators. This allows administrators to *proactively* identify potential vulnerabilities to credential stuffing, rather than responding only after an attack has occurred.

Limitations of existing prediction methodology. PassREFinder has two key limitations: i) its sub-optimal design relies on heuristically learning implicit information through hidden relations, rather than leveraging ground truth password reuse information from other administrators; and ii) it poses inherent privacy risks, as it requires access to users' private information (e.g., overlapping usernames) across potentially untrustworthy administrators to achieve optimal performance. These critical limitations hinder the practical deployment of PassREFinder in real-world settings involving untrusted parties. This highlights the need for a more advanced approach that both preserves user privacy and effectively predicts credential stuffing risks.

Our approach: PassREFinder-FL. Our goal is to predict password reuse relations between websites without compromising user privacy. To this end, we extend PassREFinder by removing the sub-optimal and privacy-sensitive hidden relation method and incorporating federated learning (FL), which mitigates the privacy risks associated with sharing password reuse data across websites. Instead of exchanging raw data, administrators share only the gradients or weights of their local GNNs, which are then used to collaboratively train a global model and construct a *cross-admin password reuse relation*. Thanks to the privacy-preserving nature of FL, these shared gradients or weights do not disclose any sensitive information. In this way, our approach enables secure and privacy-preserving prediction of credential stuffing risks.

4. PassREFinder-FL overview

The goal of PassREFinder-FL is to predict the existence of a password reuse relation—a relation between websites where users are highly likely to reuse passwords. Consistent with our original work, PassREFinder (Kim et al., 2024), we formally define that a password reuse relation exists between two given websites if the rate of *users who reuse passwords* to *users registered in both websites* (i.e., password reuse rate) is larger than the ground truth threshold τ[1] To evaluate our novel approach based on graph federated learning, we particularly focus on predicting cross-admin password reuse relations, which are relations among websites managed by different administrators.

[1] Although prior measurement studies have commonly estimated credential stuffing risk using the values of password reuse rates or password strengths (Das et al., 2014; Hanamsagar et al., 2018; Pearman et al., 2017; Thomas et al., 2017; Wang et al., 2016a, 2023a), we require a binary criterion to classify risk as high or low for a concrete and intuitive evaluation of our method. Accordingly, we introduce a tunable ground truth threshold τ, consistent with PassREFinder (our prior work), for balancing a trade-off between prediction performance and granularity.

4.1. Challenges and approaches

We now discuss three key challenges in achieving the aforementioned goal, along with our approaches to address them.

Complicated password reuse relations (C1). The password reuse relations among multiple websites are highly complicated to be represented formally because users' password choice is determined by the passwords already used in other websites. Thus, we need to analyze thousands of websites and their mutual influences in an attempt to predict password reuse relations at the website level. An intuitive method is to utilize the similarity between websites (Pearman et al., 2017; Wash et al., 2016). However, in our evaluation, we present that password reuse relations are not able to be represented solely by the similarity of website features due to their implicit nature. Even existing prediction models such as machine learning classifiers have difficulty representing the complicated relations of password reuse among a large number of websites.

To address this challenge, we leverage a graph structure that can represent edges for password reuse relations among website nodes. Then, we predict the existence of a password reuse relation edge between two given website nodes based on graph neural networks (GNNs), which are dedicated to learning complicated relations by benefiting from the propagation and aggregation of neighborhood influences.

Diverse data structures and different influences of features (C2). According to previous empirical studies, the extent of password reuse among websites is affected by diverse types of website features (Gaw & Felten, 2006; Pearman et al., 2017; Stobert & Biddle, 2014; Thomas et al., 2019; Wang et al., 2018; Wash et al., 2016). For example, users can sign in to online news websites with general passwords, but they use unique passwords for financial websites, and they tend to use different passwords for websites that have different security levels. In other words, it is challenging to comprehensively consider the website features for the task of predicting password reuse relations. The first reason is that the data structures of the website features are entirely different from each other (e.g., location, URL, and text). This makes it difficult for neural networks to learn the representations and leads to performance degradation in the prediction (see Section 6.2.2). Moreover, the influence of a feature differs not only by the feature but also by specific website relationships.

Therefore, defining which is more important for prediction is complicated. Thus, we adopt the technique of *multi-modal features* (Ngiam et al., 2011) and attention mechanisms (Hori et al., 2017; Veličković et al., 2017) to learn the diverse data structures of the website features separately and reflect their different quantitative importance.

Difficulty in applying a graph-based model to arbitrary websites (C3). In practice, website administrators may seek to predict the password reuse relation between a website they manage and one managed by another administrator, referred to as a *cross-admin password reuse relation*. However, predicting these relations is challenging due to the limited access to password reuse information from external websites. To address this, PassREfinder (Kim et al., 2024) introduces the *hidden relation*, which links external website nodes using complementary edges rather than password reuse relation edges. Despite its effectiveness, the original design of PassREfinder has two critical limitations. First, the method is inherently sub-optimal as it establishes hidden relations heuristically rather than based on ground truth (i.e., password reuse information) from external websites, which is crucial for accurately predicting cross-admin password reuse relations. Second, it requires private user information (e.g., matched users across websites) to optimize prediction performance. This potential privacy risk may discourage website administrators from jointly deploying PassREfinder although broad participation is essential for maximizing its accuracy and coverage.

To overcome these limitations, we propose a graph federated learning approach to predict password reuse relations in a privacy-preserving manner. This enhancement allows PassREfinder-FL to independently learn the password reuse relations associated with each administrator's websites without sharing sensitive information, such as passwords and usernames, across different administrators.

4.2. Overall architecture

We design each component of PassREfinder-FL (see Fig. 2) to address the aforementioned challenges through a series of strategic design choices. Specifically, we aim to derive effective edge representations from a password reuse graph in a privacy-preserving manner. In contrast to the original work (Kim et al., 2024), our enhanced framework is explicitly designed for deployment across multiple administrators who do not necessarily trust each other. Specifically, PassREfinder-FL incorporates three components:

Graph construction. PassREfinder-FL first builds a password reuse graph using websites under that administrator's management. Two website nodes are connected by a password reuse relation edge if a password reuse relation exists between them. We note that administrators do not need to share any information about their password reuse graphs with one another. New administrators can integrate into our framework alongside existing administrators by constructing password reuse graphs from their own websites.

Feature extraction. Next, PassREfinder-FL extracts various website features that can affect user password reuse tendencies. The features of a given website are automatically extracted by querying the information of the website from several public sources of web analytics (Apple, 2025; Google Chrome, 2025; McAfee, 2025; Mozilla NSS, 2025; Shodan, 2025; urlscan, 2025). The method of feature extraction utilizing public information of websites enables our framework to be applied to arbitrary websites on the Internet. To reflect various types of website features, we adopt an embedding method dedicated to vectorizing each data structure as an effective input for a neural network. In addition, we establish the website features in multiple modalities to interpret different types of features separately.

Graph federated learning & Cross-admin edge representation. PassREfinder-FL adopts a federated learning approach to train a GNN-based model tailored to producing representations of password reuse relation edges for predicting their existences. The detailed architecture of the GNN model follows the original design (Kim et al., 2024). During training, each administrator updates the weights of the corresponding local GNN based on the local password reuse graph and then sends it to the central server for storing the global GNN. These updated weights are then sent to a central server, where they are aggregated to form the global GNN model. The server subsequently distributes the aggregated global GNN weights back to all administrators for the next training step.

During inference, each administrator computes the node representation on the local password reuse graph and the global GNN weights they have received. To make a prediction on a cross-admin password reuse relation edge, the involved administrators exchange only their respective node representation vectors without sharing any sensitive information such as passwords and usernames. The edge representation vector then is constructed by concatenating the two exchanged node representation vectors. Finally, we predict the existence of a cross-admin password reuse relation edge by processing this edge representation vector using a feed-forward network.

5. PassREfinder-FL design

This section details the design of PassREfinder-FL.

5.1. Graph construction

Password reuse relations can be represented on a graph of websites. We formulate a *password reuse graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to model the existence of password reuse relations among websites. We denote a set of website nodes as $\mathcal{V}$ and a set of password reuse relation edges as $\mathcal{E}$. Website nodes $u, v \in \mathcal{V}$ are connected by a password reuse relation edge $e_{uv} \in \mathcal{E}$

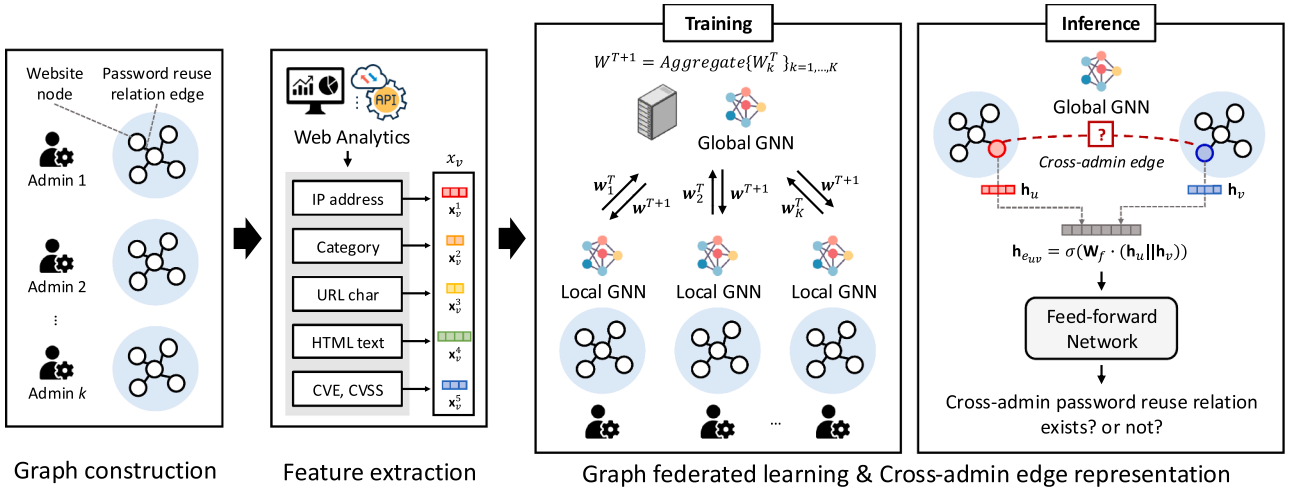


Fig. 2. Overview of PassRefinder-FL.

if a password reuse relation exists between u and v. Note that a password reuse graph is undirected and has no self-loops because a password reuse relation can be defined for an unordered pair of two websites.

In PassRefinder-FL, each administrator constructs a local password reuse graph based on (partial) information about password reuse relations among the websites the administrator manages. Specifically, the graph consists of website nodes connected by edges representing password reuse relations. An edge is established between two website nodes if the proportion of users reusing their passwords across those sites exceeds the ground truth threshold τ. These local password reuse graphs are not interconnected—there are no password reuse relation edges across websites managed by different administrators. These cross-admin password reuse relation edges are the prediction targets. Unlike the original design of PassRefinder (Kim et al., 2024), which incorporates hidden relations, our approach ensures that no privacy-sensitive data and even the local password reuse graph information are shared across administrators.

An administrator who deploys the framework uses their own constructed password reuse graph to train the GNN model, which limits the model's ability to generalize to password reuse graph patterns of websites managed by other administrators. In contrast, PassRefinder-FL can leverage a broader range of password reuse patterns across all participating administrators by learning from distributed information of their local password reuse graphs.

5.2. Feature extraction

Website node $v \in \mathcal{V}$ has an input feature (set) x_v which is established from the website's characteristics. There are diverse website features affecting users' tendency to reuse their passwords, and the data structures of such features are different from each other. Therefore, we extract features based on M *modalities*, each of which denotes a single type of website feature. We let $\mathbf{x}_v^m \in \mathbb{R}^{D_m}$ be the m-th feature vector with dimension D_m and $x_v = \{\mathbf{x}_v^1, \mathbf{x}_v^2, \ldots, \mathbf{x}_v^M\}$ be a feature set of website node v[2] The features of website nodes are extracted from website characteristics that affect password choice and reuse. We note that the website features are obtained from the public services of web analytics. In our prototype, we establish five modalities of website features with different data structures to effectively represent password reuse relations. In what follows, we describe each of the features, the source of information, and the embedding method for creating input for the GNN layers.

Location is an effective website feature for understanding the reuse of passwords. For example, user password selection habits (e.g., password strength) significantly differ by country due to specific cultures of password composition policies and languages used to create passwords (AlSabah et al., 2018; Mayer et al., 2017). Then, users may decide to reuse passwords according to the location where websites are serviced. We identify the locations of websites using their IP addresses retrieved from *urlscan.io* (urlscan, 2025), a popular website scanning service. To consider subnets in a more fine-grained manner, we encode the IP addresses into 32-bit binary embedding vectors.

Category is one of the well-known features correlated to password reuse. Users typically prioritize their accounts by the importance of the website (Gaw & Felten, 2006; Pearman et al., 2017; Thomas et al., 2019; Wang et al., 2018). In particular, websites related to financial and governmental services show that users highly avoid reusing passwords in other websites due to high concerns about the exfiltration of private or sensitive information. On the other hand, users are more likely to reuse the passwords of unimportant websites (e.g., video streaming and disposable account websites). For the classification, we utilize the categorization service of McAfee (McAfee, 2025). Then, we merge a few categories to mitigate the inconsistencies in the levels of some categories and finally limit the results to 20 categories. The category of a website cannot be represented by a numerical value or a vector. Therefore, we utilize the method of lookup embedding to obtain learnable vectors, which has been widely adopted to produce dense representations of (categorical) data (He & Chua, 2017; Tang & Wang, 2018).

Content complements the category feature as website category is sometimes ambiguous (Sherman, 2020) (e.g., providing multiple services). For the content feature, the HTML text of website pages can be used to extract information explicitly related to their services. Nevertheless, we use both the category and content features as some websites contain few texts on their pages. Notably, the text content can effectively reflect the languages of websites and users. In this regard, we use a transformer-based multilingual language model, XLM-RoBERTa (Conneau et al., 2019), which is a powerful model for tasks in various domains (Bozkir et al., 2022; Das et al., 2021; Khatua & Nejdl, 2022). We utilize the embedding vectors obtained from the pre-trained model (Face, 2019).

URL is a significant indicator of the relationship between two websites, such as the root domain and subdomain. If the URLs of two websites are related to each other, users tend to use the same usernames and passwords in the websites. For example, university students reuse their passwords in other university services that have similar URLs to that of the university homepage (i.e., the root domain) (Wash et al., 2016). We adopt a method of character-level embedding for URL

[2] In our prototype, we select website features of five modalities (i.e., $M = 5$), including the *location*, *category*, *content*, *URL*, and *security posture* of a website.

using Long Short-Term Memory (LSTM), modified from an existing URL embedding approach (Yang et al., 2019).

Security posture is an important feature that users consider when reusing their passwords because it is directly related to the level of trust on a website (Gaw & Felten, 2006; Stobert & Biddle, 2014). We conduct a security analysis on website systems through *Shodan* (Shodan, 2025), a search engine for Internet devices, and extract software vulnerabilities in a Common Vulnerabilities and Exposures (CVE) (CVE, 2022) format. We first select the number of software operating in a website system as a large number of software extends the attack surface. We also utilize the average number of CVEs in each software and the average/maximum values of Common Vulnerability Scoring System (CVSS) scores (Vulnerability Metrics, 2022) to measure the severities of the vulnerabilities. In addition, we examine HTTPS adoption, a more conspicuous security posture for users by validating website certificate chains. We use *OpenSSL* by referring to the trusted certificate authorities of Google Chrome (Google Chrome, 2025), Mozilla Firefox (Mozilla NSS, 2025), and Apple macOS (Apple, 2025) to analyze certificate errors such as certificate expiration, self-signed certificate (chain), unverifiable first certificate in chain and unverifiable local issuer, which is similar to the existing approach (Singanamalla et al., 2020). To stabilize the neural network training process while managing diverse scales of security posture features, we adopt batch normalization techniques (Ioffe & Szegedy, 2015).

5.3. Graph federated learning & cross-admin edge representation

We follow the design of GNNs introduced in our previous work, PassRefinder (Kim et al., 2024), which demonstrates its tailored capability in representing password reuse relations. Specifically, we adopt the novel aggregation function and two different levels of attention mechanisms, which enable us to model various website features and edges related to users' password reuse tendency. Subsequently, we introduce a federated learning algorithm for representing password reuse relations based on local password reuse graphs. We note that the hidden relation method of PassRefinder has been entirely removed in PassRefinder-FL.

Formally, we consider a central server and K website administrators as clients. Each administrator $k \in \{1, \dots, K\}$ builds a password reuse graph $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k)$ based on the set $\mathcal{V}_k$ of websites that the administrator manages and the set $\mathcal{E}_k$ of password reuse relations among these websites. Initially, the password reuse graph of an administrator is disconnected with other administrators' graphs. Now, we want to predict the existence of a cross-admin password reuse relation edge e_{uv}, where $u \in V_k, v \in V_{k'}$ $(k \neq k')$. To this end, the node representation of website node u is obtained by applying GNNs on the password reuse graph $\mathcal{G}_k$ with website features $\{x_w : w \in \mathcal{V}_k\}$. Independently, the node representation of website node v is obtained from the password graph $\mathcal{G}_{k'}$ with website features $\{x_w : w \in \mathcal{V}_{k'}\}$. Based on the node representations of u and v, our model generates d-dimensional password reuse relation edge representations $\{\mathbf{h}_e \in \mathbb{R}^d : e \in \mathcal{E}\}$ for facilitating the link prediction task regarding (cross-admin) password reuse relations.

We now describe how to learn vector representations for website nodes in each administrator's local password reuse graph, using website-specific features. The overall procedure is illustrated in Algorithm 1. The main inputs to the algorithm include a local password reuse graph $\mathcal{G}$ and, for each node, a feature set x_v spanning M modalities. The number of GNN layers is denoted by L. For each modality and layer, the algorithm uses a total of $M \times L$ uniformly initialized weight matrices $\mathbf{W}^{m,l}$. The algorithm further relies on a neighborhood function $\mathcal{N}$, which returns the neighbors of a given node in $\mathcal{G}$. Lastly, it takes as input the neighbor aggregation and modality attention functions we defined. The goal is to obtain a vector representation $\mathbf{h}_v$ for each website node in the graph $\mathcal{G}$.

Feature initialization. The algorithm begins by initializing the node representation vectors (lines 1 to 2). For each modality m, we assign the input feature $\mathbf{x}_v^m$ to the node representation vector of the initial layer $\mathbf{h}_v^{m,0}$.

Algorithm 1 Local node representation learning.

Input: Local password reuse graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; the number of modalities M; the number of GNN layers L; input node feature set $x_v = \{\mathbf{x}_v^1, \mathbf{x}_v^2, \dots, \mathbf{x}_v^M\}$; weight matrices $\mathbf{W}^{m,l}, \forall m \in \{1, \dots, M\}, \forall l \in \{1, \dots, L\}$; activation function σ; neighborhood functions $\mathcal{N} : v \to 2^{\mathcal{V}}$; neighbor aggregation function AGG; modality attention function ModalityAttn

Output: vector representations $\mathbf{h}_v, \forall v \in \mathcal{V}$

1: **for** $m = 1 \dots M$ **do**
2: $\quad \mathbf{h}_v^{m,0} \leftarrow \mathbf{x}_v^m, \forall v \in \mathcal{V}$
3: $\quad$ **for** $l = 1 \dots L$ **do**
4: $\quad\quad$ **for** $v \in \mathcal{V}$ **do**
5: $\quad\quad\quad \mathbf{h}_{\mathcal{N}(v)}^{m,l} \leftarrow \text{AGG}(\mathbf{h}_u^{m,l-1}, \forall u \in \mathcal{N}(v))$
6: $\quad\quad\quad \mathbf{h}_v^{m,l} \leftarrow \sigma(\mathbf{W}^{m,l} \cdot \text{CONCAT}(\mathbf{h}_v^{m,l-1}, \mathbf{h}_{\mathcal{N}(v)}^{m,l}))$
7: $\quad\quad$ **end for**
8: $\quad$ **end for**
9: $\quad \mathbf{h}_v^m \leftarrow \mathbf{h}_v^{m,L} / \|\mathbf{h}_v^{m,L}\|_2, \forall v \in \mathcal{V}$
10: **end for**
11: $\mathbf{h}_v \leftarrow \text{ModalityAttn}(\mathbf{h}_v^m, \forall m \in \{1, \dots, M\}), \forall v \in \mathcal{V}$

Neighbor aggregation. A website node's representation is shaped not only by its own features, but also by the features of its neighbors. Thus, for each GNN layer l (and for each modality m), the node v's representation vector is produced by aggregating that of its neighbors (lines 4 to 7). Aggregated neighbors are determined by the neighborhood function $\mathcal{N}$. To prioritize neighbors, we adopt neighborhood attention during the aggregation. It takes the node representation vector $\mathbf{h}_v^{m,l-1}$ and each neighbor u's node representation vector $\mathbf{h}_u^{m,l-1}$ as inputs to produce the attention coefficient $\alpha_v^{m,l}(u,t)$ that denotes the influence of a neighbor node u on the given node v propagated through edges. The attention coefficient is formulated as follows:

$$\widetilde{\alpha}_v^{m,l}(u) = \sigma(\mathbf{a}_{m,l}^T(\mathbf{W}^{m,l}\mathbf{h}_u^{m,l-1} + \mathbf{W}^{m,l}\mathbf{h}_v^{m,l-1})), \tag{1}$$

where $\mathbf{W}^{m,1} \in \mathbb{R}^{D_m \times d}$ and $\mathbf{W}^{m,l(>1)} \in \mathbb{R}^{d \times d}$ denote weight matrices for the node representation vectors, $\mathbf{a}_{m,l} \in \mathbb{R}^d$ denotes a weight vector for parameterizing a single scalar value, and $\cdot^T$ is the transposition operator.

Then, we normalize the attention coefficients across all neighbor nodes using the softmax function:

$$\alpha_v^{m,l}(u) = \frac{\exp(\widetilde{\alpha}_v^{m,l}(u))}{\Sigma_{w \in \mathcal{N}(v)} \exp(\widetilde{\alpha}_v^{m,l}(w))}. \tag{2}$$

We can obtain the aggregated neighborhood vector using a linear combination of the neighbors' node representation vectors based on the normalized attention coefficients:

$$\mathbf{h}_{\mathcal{N}(v)}^{m,l} = \sum_{u \in \mathcal{N}(v)} \alpha_v^{m,l}(u)\mathbf{h}_u^{m,l-1}. \tag{3}$$

After the aggregation on each node, we impose different weights to the node representation vector $\mathbf{h}_v^{m,l-1}$ and the aggregated neighborhood vector $\mathbf{h}_{\mathcal{N}(v)}^{m,l}$ by concatenating the two vectors and multiplying its result by the weight matrix $\mathbf{W}^{m,l}$, which is effective for inductive learning (Hamilton et al., 2017) (*cf.* adding them in GCN (Kipf & Welling, 2016)). Subsequently, we obtain node v's representation vector $\mathbf{h}_v^{m,l}$ in layer l.

Multi-modality attention. After we normalize the node representation vectors of the last layer L (line 9), we apply the modality attention function across the node representation vectors of M modalities (line 11). The normalized modality attention coefficients are formalized as follows:

$$\beta_v(m) = \frac{\exp(\sigma(\mathbf{b}_m^T\mathbf{W}_1\mathbf{h}_v^m + b))}{\Sigma_{m'=1}^M \exp(\sigma(\mathbf{b}_{m'}^T\mathbf{W}_1\mathbf{h}_v^{m'} + b))}, \tag{4}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d \times d}$ denotes a weight matrix for the node representations of the modalities, $\mathbf{b}_m \in \mathbb{R}^d$ denotes a weight vector for parameterizing a single scalar, and b denotes a scalar bias. The node representation vector $\mathbf{h}_v$ is obtained through a linear combination of the modalities' node representation vectors multiplied by the normalized modality attention coefficients:

$$\mathbf{h}_v = \sum_{m=1}^{M} \beta_v(m)\mathbf{h}_v^m. \tag{5}$$

Edge representation under the FL setting. To infer password reuse relationships between websites managed by different administrators—without exposing sensitive user data—we extend our original work (Kim et al., 2024) with a federated learning (FL) approach. In this setting, each administrator maintains a local password reuse graph $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k), \forall k \in \{1, \dots, K\}$ and we aim to collaboratively train a GNN model across all K administrators. Our objective is to learn an edge representation vector $\mathbf{h}_{e_{uv}}$ for each pair of websites $u \in \mathcal{V}_k$ and $v \in \mathcal{V}_{k'}$ belonging to different administrators $k \neq k'$.

In the inference phase (forward pass), each administrator first computes local node representation vectors using Algorithm 1. These vectors are then shared across administrators, enabling the construction of edge representations between nodes owned by different parties. Specifically, the edge representation $\mathbf{h}_{e_{uv}}$ is derived by concatenating the node embeddings of u and v, followed by a linear transformation and activation:

$$\mathbf{h}_{e_{uv}} = \sigma(\mathbf{W}_f \cdot \textsc{Concat}(\mathbf{h}_u, \mathbf{h}_v)), \tag{6}$$

where σ is an activation function and $\mathbf{W}_f$ is the weight matrix for edge representation learning. To predict whether an actual password reuse relationship exists between u and v, we apply a feed-forward network f followed by a sigmoid activation function:

$$\hat{p}_{e_{uv}} = sigmoid(f(\mathbf{h}_{e_{uv}})). \tag{7}$$

Here, $\hat{p}_{e_{uv}}$ denotes the predicted probability of an edge. A threshold τ is applied such that a reuse relation is considered to exist if and only if $\hat{p}_{e_{uv}} \geq \tau$. In our implementation, f is a simple fully connected layer. Unlike prior FL-GNN approaches (He et al., 2021; Scardapane et al., 2020; Yao et al., 2023), our method avoids message passing across administrative domains. Instead, cross-admin edges are treated solely as prediction targets during training. This design choice ensures that no password reuse information is directly exchanged across administrators, thereby preserving user privacy.

In the training phase, our goal is to optimize the global model parameters

$$\mathbf{w} = \{\mathbf{W}^{m \in \{1,\dots,M\}, l \in \{1,\dots,L\}}, \mathbf{W}_f\}, \tag{8}$$

which are initially stored on a central server and distributed to all participating administrators. The optimization objective is defined as:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}_1,\dots,\mathbf{w}_K} \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_k(\mathbf{w}_k; x_k; \hat{y}_k, y_k)$$
$$\text{s.t.} \quad \mathbf{w} = \mathbf{w}_1 = \mathbf{w}_2 = \cdots = \mathbf{w}_K, \tag{9}$$

where x_k denotes the node features of $\mathcal{V}_k$, and $\hat{y}_k$ and y_k are the predicted and ground-truth labels of password reuse edges $e \in \mathcal{E}_k$ within the local graph. Following the distributed GCN training strategy (Scardapane et al., 2020), this problem can be solved in a fully distributed manner.

Algorithm 2 describes the details of the federated learning process. Each administrator independently minimizes its local loss function $\mathcal{L}_k$ using only its local password reuse graph $\mathcal{G}_k$. The local model update at training step t is computed as:

$$\mathbf{w}_k^{t+1} = \mathbf{w}_k^t - \eta^t \nabla_{\mathbf{w}_k} \mathcal{L}_k(\mathbf{w}_k; x_k; \hat{y}, y), \tag{10}$$

where η^t is the learning rate at step t. After local training, the updates from all administrators are sent to the central server, which aggregates them to update the global model as follows:

$$\mathbf{w}^{t+1} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{w}_k^{t+1} \tag{11}$$

For the local loss function $\mathcal{L}_k$, we use binary cross-entropy between the predicted probability $\hat{p}_e$ and the ground truth label $y_e \in \{0, 1\}$:

$$\mathcal{L}_k = -\frac{1}{|\mathcal{E}_k|} \sum_{e \in \mathcal{E}_k} y_e \log \hat{p}_e + (1 - y_e) \log(1 - \hat{p}_e). \tag{12}$$

Algorithm 2 Graph federated learning.

Input: Central server S; Clients (administrators) $1, \dots, K$; Local password reuse graphs $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k), \forall k \in K$; Training steps T; Learning rates $\eta^t, \forall t \in T$
Output: Trained GNN weights

for step $t \in T$ **do**
  for client $k \in K$ **do**
    (Client side)
    $\mathbf{w}_k^{t+1} = \mathbf{w}_k^t - \eta^t \nabla_{\mathbf{w}_k} \mathcal{L}_k(\mathbf{w}_k; \mathcal{G}_k)$
    Transmit $\mathbf{w}_k^{t+1}$ to central server S.
  end for
  (Sever side)
  $\mathbf{w}^{t+1} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{w}_k^{t+1}$
  Transmit the copies of $\mathbf{w}^{t+1}$ to every client.
end for

Mini-batch training and implementation. We train all weights of the embedding layers (e.g., lookup embedding, LSTM, and batch normalization layers), the GNN layers, and the final feed-forward network in an end-to-end manner. Fully training a large password reuse graph is difficult due to GPU memory requirements, proportional to the number of nodes (Stochastic Training on Large Graphs, 2022). To handle this problem, we utilize mini-batches of edges in a graph and recursively train every sub-graph which is composed of 1 to L-hop neighborhood of two nodes of a given password reuse relation edge. While edge sampling and sub-graph building tasks are CPU-bound, training each sub-graph is conducted on a GPU. We implement PassREfinder-FL using Deep Graph Learning (DGL) (Wang et al., 2019), a library providing deep learning modules for graphs.

6. Evaluation

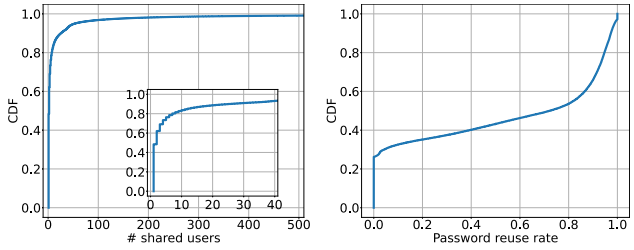
In this section, we present our experiment setup on a real-world dataset and the evaluation results of PassREfinder-FL. We primarily compare PassREfinder-FL with its original design, PassREfinder (Kim et al., 2024). Additionally, we provide the extended results of the FL setting, including the effectiveness of tailored GNN models, an ablation study on the modeling choices, and a risk score analysis.

6.1. Experiment setup

6.1.1. Dataset

We adopt the `CitOday` (Troy Hunt, 2025; Wayback Machine, 2025) dataset, which is a large-scale credential dataset from a credential selling service leak in November 2020. The dataset contains 360,115,257 accounts (i.e., email addresses and passwords) breached from 22,378 websites in the real world. Since features of defunct websites cannot be extracted from public web analytics, we filter them out and leave 19,062 reachable websites. Among them, we first select 15,636 websites where users' plain-text passwords are identifiable to obtain precise information on password reuses.

A single user is indicated by a unique email address (a commonly used user identifier in password analyses (Das et al., 2014; Thomas et al., 2017; Wang & Reiter, 2019, 2020)). We note that email addresses are entirely anonymized to address ethical considerations. The number



(a) The number of shared users (b) Password reuse rate

Fig. 3. The distributions of the number of shared users and password reuse rate between websites.

Table 1
Dataset statistics. We present the number of edges in each split set. *Local* edges refer to those within each local password reuse graph $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k), k \in \{1, \ldots, K = 10\}$. *Cross-admin* edges represent those connected across different administrators' websites and serve as the prediction targets in our experiment.

Edge type	Split	# of edges
Local ($\mathcal{E}_k$)	Train	48,990/14,484/16,000/15,144/11,242 /7,594/14,110/23,466/5,842/9,820
Cross-admin	Valid	24,421
	Test	498,204
Total		605,971

of shared users between websites ranges from 0 to 733,300 with most within 500 (see Fig. 3). To ensure the confidence of analysis, we define the ground truth of password reuse relation edges only when the number of shared users is at least 30, which is the minimum number of observations required for statistical significance (Hogg et al., 1977). We filter out websites that share fewer than 30 users with every other website, leaving 9927 websites in the end. Among the shared users between two given websites, the password reuse relation edge is labeled as positive if the rate of users who reuse passwords in the websites is higher than the threshold τ, or labeled as negative otherwise. Website pairs having a zero rate account for 26.2 % of the total (see Fig. 3). Consistent with our prior work, PassReFinder, we set the threshold τ to 0.5 in the main experiment.

Our final dataset for the experiments consists of 9927 website nodes connected by 605,971 password reuse relation edges. To simulate an FL setting with $K = 10$ participating administrators, we partition the original graph into ten disjoint subgraphs. Each subgraph corresponds to one administrator, with 1000 nodes assigned per administrator, except for the last subgraph containing 927 nodes. As a result, all cross-admin password reuse relation edges are removed during partitioning and excluded for message passing in the GNN model's forward pass. For evaluation, we use cross-admin edges between two randomly selected administrators as the validation set, while the remaining cross-admin edges are used for testing. Detailed statistics of the dataset are presented in Table 1.

6.1.2. Training and hyperparameters

In our prototype framework, we set the dimension D_m of the lookup embedding vectors and the LSTM output vectors to 256. We use the embedding vector of XLM-RoBERTa with the default 768 dimensions. For each of the $M = 5$ modalities, we train $L = 2$ layers of the GNNs with $d = 256$ dimensions of hidden vectors. In each GNN layer, we leverage all neighbors of each node, instead of sampling from them, to fully learn the neighborhood information. We also adopt leaky ReLU with 0.2 negative slope for the activation function σ and a fully connected layer for the feed-forward network f as our design choice.

We evaluate various settings of hyperparameters: maximum epochs of 100, 200, 500, and 1,000; (edge) batch sizes of 2^{12}, 2^{14}, 2^{16}, and 2^{18}; maximum learning rates of 10^{-4}, 10^{-3}, 10^{-2}, and 10^{-1}. Then, we select a maximum epoch of 200 with 40 epochs for early stopping, a batch size of 2^{16}, and a maximum learning rate of 10^{-3} using the Adam optimizer, of which combination shows the best performance. In addition, 1 cycle policy (Smith & Topin, 2019) with a warmup of 10 % is adopted to schedule the learning rate. We evaluate our framework on a machine with 20 Intel i9-10900K CPU cores, 128GB of memory, and a single NVIDIA RTX 3090 GPU.

6.1.3. Baselines

PassReFinder (Kim et al., 2024). We evaluate the credential stuffing risk prediction framework proposed in our prior work, which leverages hidden relation modeling to estimate risks across websites managed by different administrators. Specifically, we assess three variants based on different hidden relation policies: *No hidden relation*, PassReFinder$_{Sim_{avg}}$, and PassReFinder$_{User30}$[3] For training PassReFinder, we use only the first local password reuse graph $\mathcal{G}_1$, which consists of 1000 nodes and 48,990 edges (see Table 1).

GNN variants with FL. Under the same FL setting, we compare various GNN model architectures, which have been frequently used in risk prediction tasks (Bilge et al., 2017; He et al., 2022; Soska & Christin, 2014; Veeramachaneni et al., 2016; Zhang et al., 2020):

- **Graph convolutional network (GCN)** (Kipf & Welling, 2016) is a GNN with a layer-wise propagation rule and neighbor aggregation based on spectral graph convolutions. We adopt $L = 2$ layers of GCNs for learning representations from multiple hops of neighbors as in our GNN model.
- **Graph attention network (GAT)** (Veličković et al., 2017) is a GCN variant adopting self-attentional neighbor aggregation for selectively learning the influences of important neighbors.
- **GraphSAGE** (Hamilton et al., 2017) is a GCN variant for effectively performing inductive learning by concatenating the self-embedding with the neighbor aggregation vector. We set a mean pooling function for neighbor aggregation.

For a fair comparison with PassReFinder-FL, we utilize multi-modality node features rather than relying on the conventional feature concatenation approach.

6.2. Results

6.2.1. Prediction performance

The performance of PassReFinder-FL and its comparison with the original framework, PassReFinder (Kim et al., 2024), are summarized in Table 2. PassReFinder-FL achieves superior prediction performance with an F1-score of 0.9153 while preserving privacy under the FL-based setting. This notable effectiveness is attributed to its ability to comprehensively leverage ground truth (i.e., password reuse information) from websites managed by multiple administrators during both training and inference. In contrast, PassReFinder is limited in its access to such information beyond a single administrator's websites, resulting in lower performance. Among the PassReFinder variants, the implementation based on overlapping user information achieves the highest F1-score of 0.8234, consistent with prior results (Kim et al., 2024). Notably, this approach requires sharing usernames across potentially untrustworthy administrators, posing significant privacy risks to users. Privacy-preserving implementations maintain better privacy: the one with no hidden relation and the website feature similarity-based hidden relation method. However, their prediction performance is limited due to the smaller size of the available training data.

To evaluate the effectiveness of our GNN design within the FL-based framework, we compare several GNN variants under the same setting

[3] The PassReFinder$_{Sim_{avg}}$ model creates a hidden relation edge if the feature similarity between two websites exceeds the average similarity observed in the training set. The PassReFinder$_{User30}$ model creates an edge when at least 30 shared users exist, providing statistical significance (Kim et al., 2024).

Table 2
Overall prediction performance. The best and the second best scores are in bold and underlined, respectively.

Model	Hidden relation policy	Privacy-preserving	Performance		
			Precision	Recall	F1-score
PassREfinder (Kim et al., 2024)	*No hidden relation*	✓	0.9238	0.6405	0.7565
	Website feature similarity (Sim_{avg})	✓	0.8851	0.6971	0.7799
	Overlapped user number ($User_{30}$)	✗	**0.9484**	0.7274	0.8234
PassREfinder-FL	*No hidden relation*	✓	0.9425	**0.8896**	**0.9153**

Table 3
Prediction performance across GNN variants under the FL setting. The best and the second best scores are in bold and underlined, respectively.

Model	Performance		
	Precision	Recall	F1-score
GCN (Kipf & Welling, 2016)	0.8146	0.8243	0.8195
GAT (Veličković et al., 2017)	0.7732	**0.9555**	0.8547
GraphSAGE (Hamilton et al., 2017)	**0.9462**	0.8207	0.8790
PassREfinder-FL	0.9425	0.8896	**0.9153**

as PassREfinder-FL. The experimental results are presented in Table 3. The GNN model used in PassREfinder-FL achieves the highest F1-score among the variants, effectively balancing precision and recall. Similar to previous observations for PassREfinder, GCN and GAT suffer from low precision scores. In contrast, GraphSAGE—known for its robustness in inductive learning settings—outperforms other variants with a competitive F1-score of 0.8790.

Comparison with existing detection methods. We conduct a comparative analysis between our website-level prediction model and existing user-level detection methods. The detection accuracy of a C3 service can be approximated based on its database coverage. We evaluate the HIBP database before the leak (Have i been pwned?, 2022) and determine that it detects reused passwords with an accuracy of 0.6510. In the coordination method (Wang & Reiter, 2019), we evaluate the probability of a user's password being reused in at least one of 64 queried websites: $1-(\mathbb{P}(\pi=\Psi_i(a)))^{64}$ for password π of user a, where $\Psi_i(a)$ denotes the password of user a in website i. This exhibits an average accuracy of 0.7857 in our dataset. In addition to the advantages of scalability and minimal privacy implications, PassREfinder-FL (with 0.9153 F1-score) significantly outperforms these two detection methods.

6.2.2. Ablation study

To evaluate the effectiveness of each design strategy, we conduct an ablation study on the GNN modeling components of PassREfinder-FL, as presented in Table 4. Specifically, we examine the contributions of the multi-modality feature and the neighborhood attention methods (Section 5).

For the ablation of the multi-modality features, we replace the separate processing of different feature modalities with a simple concatenation of all website node features, followed by a single unified GNN. The results show a substantial drop in prediction performance from 0.9153 to 0.7383, highlighting the importance of multi-modality feature modeling even under the FL setting. This finding is consistent with prior observations in PassREfinder (Kim et al., 2024).

To assess the impact of the neighborhood attention mechanism, we replace it with a standard mean pooling aggregation. While this method showed clear benefits in the inductive setting of PassREfinder, its contribution in PassREfinder-FL is less significant, with only a slight gain in recall. We attribute this difference to the FL-based approach's ability to leverage comprehensive website information across multiple administrators, allowing PassREfinder-FL to maintain remarkable prediction performance without requiring fine-grained prioritization of neighborhood nodes.

Table 4
Ablation study on the GNN modeling methods.

Ablation	Performance		
	Precision	Recall	F1-score
Multi-modality	0.8730	0.6396	0.7383
Neighborhood attention	0.9429	0.8606	0.8999
None (PassREfinder-FL)	0.9425	0.8896	0.9153

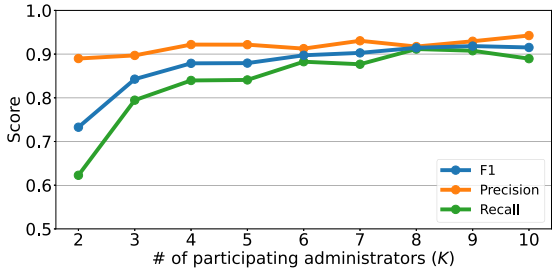


Fig. 4. Prediction performance with varying numbers of administrators participating in federated learning.

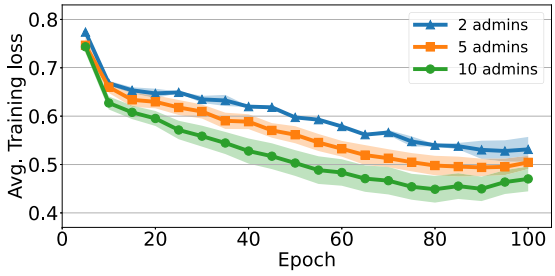


Fig. 5. Training dynamics across federation sizes. The shaded regions indicate standard errors across participating administrators.

6.2.3. Size of federation

In federated learning, the number of participating local clients is crucial for achieving satisfactory performance. To evaluate the robustness of PassREfinder-FL in the federated setup, we examine prediction performance as the number of participating administrators varies.

The results are shown in Fig. 4. For fair evaluation, we fix the test set to the cross-admin edges across all local password reuse graphs. Prediction performance, measured by F1, precision, and recall, improves with greater participation. This improvement is attributed to expanded learned knowledge and generalization capability, particularly on cross-admin edges across local password reuse graphs. With only 2 administrators, the F1 score is 0.7327, falling short of PassREfinder (our prior work). Nevertheless, practical performance (F1-scores of >0.9) is achieved with at least 5 administrators. Consequently, PassREfinder-FL remains effective with a reasonable federation size, demonstrating practical advantages under limited administrator participation in real-world scenarios.

To further examine the impact of federation size, we track the averaged loss (i.e., $\mathbb{E}_{k\in\{1,\ldots,K\}}\mathcal{L}_k$) over training, as shown in Fig. 5.

Overall, PassREFinder-FL exhibits successful convergence in the federated learning setup, indicating that it learns effective representations of cross-admin password reuse relations from local password reuse graphs. As the number of participating administrators increases, the loss converges more stably and to a lower value, consistent with the improvements in prediction performance in Fig. 4.

6.2.4. Risk score analysis

Some administrators may prefer a more fine-grained outcome for quantifying risk rather than a binary result indicating whether a password reuse rate is high or low. Therefore, we show that the prediction probability $\hat{p}_{e_{uv}}$ in Equation (7) can serve as *risk scores* of credential stuffing—the expected password reuse rates between two website nodes u and v. Specifically, we calculate the average risk scores between a given website and the others to obtain the website's risk score. We examine two tasks: prioritizing website pairs by their relative risks and estimating the risk score value. For comparison, we present the results of PassREFinder with the user information-based hidden relation method ($User_{30}$).

We first demonstrate that the prediction probability produced by our prediction model effectively prioritizes the risk levels between pairs of websites. To evaluate this, we use ranking-based metrics: precision and normalized discounted cumulative gain (nDCG) at top-k predicted edges for a given website node (see Fig. 6). Precision@k measures the proportion of truly positive edges among the top-k predicted edges, while nDCG@k considers both the relevance (i.e., risk score) and the ranking order. As a baseline, we introduce a naive method (*Random*) that randomly predicts the risk score between two websites based on a uniform distribution.

PassREFinder-FL significantly outperforms the *Random* baseline in both precision@k and nDCG@k, demonstrating its effectiveness in prioritizing website pair across different administrators by identifying higher-risk password reuse relations. PassREFinder performs much worse than PassREFinder-FL in the link prediction task (as shown in Table 2). In contrast, it serves as a competitive baseline for risk score prediction, where PassREFinder-FL achieves only marginal improvements. Nonetheless, the primary advantage of PassREFinder-FL lies in its privacy-preserving design, making it highly practical for real-world deployment without requiring local password reuse information sharing between administrators.

Additionally, we examine whether the prediction probability can be directly interpreted as a risk score. In Fig. 7, we visualize the prediction probabilities generated by PassREFinder and PassREFinder-FL, along with their errors from the true password reuse rates (i.e., Ground truth). PassREFinder-FL's predictions closely align with the ground truth, achieving slight improvements over PassREFinder. These results suggest that our GNN model, under the FL setting, can quantitatively measure credential stuffing risks between websites in practical scenarios.

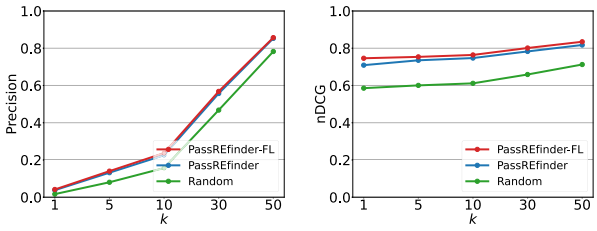
6.2.5. Communication cost analysis

We theoretically estimate the communication overheads of PassREFinder-FL in the training and inference processes, respectively.

Training. Every federated learning round t, the participating administrators $k \in \{1, \dots, K\}$ upload their local updates and download the aggregated global parameters $\mathbf{w}$ (see Algorithm 2). Here, the total model size is computed as $|\mathbf{w}| = \sum_{m,l} |\mathbf{W}^{m,l}| + |\mathbf{W}_f|$. The total communication cost of training is:

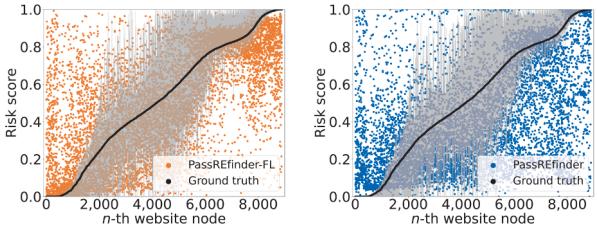
$$C_{\text{train}} = \sum_{t=1}^{T} \left(C_{\text{up}}^{(t)} + C_{\text{down}}^{(t)} \right) \approx 2\, b\, |\mathbf{w}| K T, \tag{13}$$

where b denotes the number of bytes per scalar. which grows linearly with the number of participating administrators and rounds. Since $|\mathbf{w}|$ is fixed, the overhead remains predictable and scalable, ensuring practicality even in large federations.



(a) Precision@k (b) nDCG@k

Fig. 6. Evaluation using ranking metrics. We present the results with 64 sampled edges for each website node.



(a) PassREFinder-FL MSE=0.0581 (b) PassREFinder MSE=0.0925

Fig. 7. Risk scores averaged over the edges from each website node. The shaded regions are the range between the 25th and 75th percentiles of the ground truth risk scores. MSE denotes the mean squared error of each result.

Inference. At the inference time, cross-admin prediction requires only transmitting node embedding vectors. For a test edge set $\mathcal{Q}$ with embedding dimension d, the communication cost is:

$$C_{\text{infer}}(\mathcal{Q}) = \delta\, b\, d\, |\mathcal{Q}|, \tag{14}$$

where $\delta \in 1, 2$ indicates whether the implementation transmits embedding vectors in one way ($\delta = 1$) or bidirectionally ($\delta = 2$). In line with the communication cost during training, this cost scales linearly with the size of the test dataset and the embedding dimension, and is significantly lighter than exchanging raw graphs.

Overall, PassREFinder-FL achieves communication efficiency by design: both training and inference overheads increase only *linearly* with federation sizes or test dataset sizes, highlighting its scalability and practicality for real-world deployment.

7. Practical implementation and application of PassREFinder-FL

We illustrate the workflow of PassREFinder-FL in Fig. 8. PassREFinder-FL can be implemented as an application (i.e., PassREFinder-FL App), and administrators deploy the PassREFinder-FL App in their local environments by linking the websites' credential databases. For a secure and privacy-preserving implementation, administrators should pseudonymize usernames and hash passwords before providing them to the PassREFinder-FL App. In each administrator's local environment: i) the PassREFinder-FL App explores the credential databases of the administrator's websites and assesses credential stuffing risks among them. ii) It constructs a local password reuse graph and extracts website features from public web analytics services. iii) Under federated learning, it trains the local GNN model using local credential stuffing risk information in a privacy-preserving manner. iv) By transmitting only the inferred embedding vectors, it predicts credential stuffing risks between websites, particularly across different administrators (i.e., cross-admin password reuse relations). This process produces a credential stuffing risk report that specifies the risk level for each website pair (i.e., whether the risk is high or low). Based on these results, administrators can identify *risky websites* that pose potential threats to their

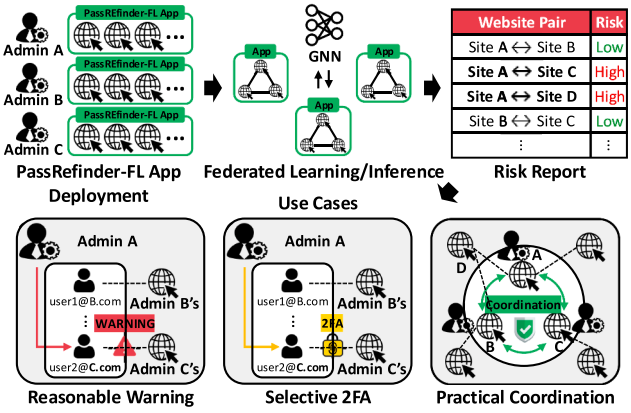


Fig. 8. PassREfinder-FL workflow and its practical use cases.

local websites. Notably, this preemptive risk prediction approach addresses the limited detection coverage of C3 services and the poor password usability and scalability of coordination methods. Consequently, administrators can proactively mitigate credential stuffing risks using insights into risky websites. We describe three practical use cases of PassREfinder-FL:

Reasonable warning. One use case of PassREfinder-FL is to selectively warn users of high-risk relationships between administrators' websites and risky websites. To illustrate this, consider the case where PassREfinder-FL is not deployed but administrators are aware of risky websites (e.g., Admin C's website in Fig. 8). A conventional approach is for administrators to broadcast password-reset warnings to *all users* across their managed websites. However, this blanket strategy relies heavily on users' security awareness and may lead to fatigue from repeated exposure to warnings. Another existing approach is password reuse notifications (e.g., Google Password Manager (Thomas et al., 2019)), which inform users when the same password is reused from breached websites. Yet, such tools do not estimate the risk of password reuse relations and are not proactive solutions.

PassREfinder-FL instead provides *selective warnings*, targeting only specific users based on estimated risks. It first infers potentially affected users by matching Email domains with risky website domains, thereby avoiding the impracticality of directly identifying user registration data. Then, PassREfinder-FL delivers warnings exclusively to those at risk, accompanied by detailed reasoning evidence (e.g., website features). This strategy not only reduces the overhead of broadcast warnings but also enhances users' security awareness through targeted education.

Selective 2FA. Two-factor authentication (2FA) has been widely deployed to defend against credential-related attacks. However, enforcing 2FA for all users on a website is often impractical due to the required effort and time (Colnago et al., 2018; Golla et al., 2021). Instead, selectively activating 2FA for specific users based on our prediction results can strike a balance between user costs and credential stuffing mitigation. To this end, PassREfinder-FL enables administrators to adopt an approach similar to the selective warning use case. PassREfinder-FL first infers users with accounts on risky websites by matching their Email domains with those websites. It then selectively recommends or enforces 2FA adoption for these users, providing detailed explanations of website features and associated risks to encourage compliance. Furthermore, administrators can apply fine-grained levels of 2FA enforcement based on the risk scores evaluated by PassREfinder-FL, thereby allowing careful consideration of the trade-off between user convenience and strong protection.

Practical coordination. Although website coordination sharing usernames and passwords is a promising approach to combat credential stuffing, existing implementations of website coordination (Wang & Reiter, 2019, 2020) have suffered from scalability and reluctance from administrators to participate. PassREfinder-FL seeks to overcome these challenges by creating modestly sized yet effective coordination pools. To this end, administrators i) share prediction results to select each website set effective for coordination, where each website within a set has credential stuffing risks with the others. They can ii) provide persuasive explanations, underpinned by website features, to encourage other administrators' active participation in the coordination process. Then, administrators iii) deploy coordination protocols for detecting password reuse or credential stuffing per each website set. While addressing the scalability problem, PassREfinder-FL can significantly improve the detection accuracy of coordination methods, as indicated by our comparison results in Section 6.2.1.

8. Limitation and discussion

Reliability of website features. The feature extraction process yields valuable website features for modeling password reuse tendencies. Nonetheless, discrepancies may arise between a user's password creation time and the feature extraction time, potentially affecting the ground truth password reuse rate. To minimize this disparity, even in the absence of precise dates of website creation or user registration, we focus on active websites and randomly split the dataset for our training and testing processes.

In addition, multiple domains within a single cloud hosting server could exhibit varying security postures. In the absence of administrative privileges, our framework relies on server-specific security postures to achieve the capability of deployment on a vast number of arbitrary websites. It is worth noting that the effectiveness of server-specific security postures for the risk prediction task was demonstrated through our ablation studies. Nevertheless, we could further enhance the reliability of our framework by achieving domain-specific security postures. In addition, a single cloud-hosting server could host multiple domains, each exhibiting unique security postures. In scenarios where administrative privileges are not accessible, our framework adopts the server-specific security postures of websites. This enables the framework's deployment across an extensive range of arbitrary websites.

It is noteworthy that the evaluation results demonstrate the effectiveness of server-specific security postures in predicting credential stuffing risk. Still, the reliability of our framework could be further enhanced by achieving domain-specific security postures. The IP addresses of cloud hosting servers also may not accurately represent website service locations. Employing *Shodan*, we ascertain that only 13.6 % of websites in our dataset are hosted by major cloud providers, and conservatively estimate that the hosting countries of one-third of these websites correspond with their website service countries, as inferred from their top-level domains. Furthermore, the multi-modal attention can penalize insignificant features (e.g., global website locations). Consequently, we posit that these limitations exert minimal impact on our findings.

Effectiveness in asymmetric federation. Our experiments assume a balanced federation setup in which each local password reuse graph contains 1000 nodes. This setting may not fully reflect real-world scenarios where the number of websites associated with an administrator varies significantly. In this asymmetric federation setup, the prediction capability of PassREfinder-FL can be generalized to major administrators' websites by benefiting from sufficient information propagation across their large local password reuse graphs, while performance may degrade for minor administrators with small local password reuse graphs. Nevertheless, we expect the practical impact of this limitation on risk prediction to be negligible because credential stuffing risks associated with major administrators' websites are more crucial and broadly influential in practice and should be prioritized.

Risks of credential tweaking. Several studies have examined a related vulnerability in which users slightly modify their passwords instead of reusing them verbatim, known as credential tweaking (Das et al., 2014; Pal et al., 2019; Pearman et al., 2017; Wang et al., 2018, 2023b; Xu et al., 2023). Nevertheless, our default design focuses on credential stuffing for two reasons: i) Credential stuffing is more practical and

has broader impact in real-world settings. Credential tweaking typically requires multiple guesses and repeated login attempts guided by a modification rule or a password guessing model, which imposes substantial overhead and limits large scale untargeted attacks. For example, a simulation of credential tweaking (Pal et al., 2019) shows that gaining an additional $+15.8\,\%p$ in success rate requires about 1,000× more login attempts with the prediction cost of the guessing model; attackers may prefer credential stuffing, which can achieve about a 40 % success rate with a single login attempt per user. ii) The risk of credential stuffing can be quantified systematically, enabling concrete evaluation in practice. In contrast, credential tweaking depends on chosen modification rules or guessing models, making a consistent and robust definition of *reuse* difficult. Therefore, we estimate a conservative lower bound on risk from password reuse by restricting our scope to credential stuffing. Nonetheless, PassREfinder-FL can be extended to predict the risks of credential tweaking by explicitly defining *password reuse* through a selected password modification rule or a password guessing model.

9. Related work

Compromised credential checking (C3) services. C3 services allow users to cross-reference usernames and passwords based on leaked credential datasets (Have i been pwned?, 2022; Lauter et al., 2021; Thomas et al., 2019). They also help users deal with credential stuffing attacks using breach-alerts for their passwords. Thomas et al. (2019) presented Google Password Checkup (GPC), a protocol for querying usernames and passwords from credential data breaches. Li et al. (2019) formally described C3 services and their security requirements. Pal et al. (2022) extended a C3 service to target credential tweaking attacks (Pal et al., 2019). However, existing C3 services cannot prevent attacks that exploit new accounts from undisclosed credential data breaches and suffer from adversarial exploits to extract credential data (Pal et al., 2022).

In contrast, our proposed framework predicts the potential risk of credential stuffing residing between arbitrary websites. Therefore, we can prevent credential stuffing attacks before accounts are breached without relying on unreliable databases of leaked accounts.

Coordination for detecting credential stuffing. Website coordination has emerged to actively protect user accounts (Wang & Reiter, 2019, 2020). As a pioneering work, Wang and Reiter (2019) proposed a framework for preventing a user in a *requester* website from setting the same password in *responder* websites. They developed an improved private set-membership-test (PMT) protocol underlying their framework to minimize the leakage of user information. Unfortunately, the coordination method degrades the usability of passwords by interrupting password choice. Wang and Reiter (2020) mitigated this problem by modifying the design to directly detect credential stuffing attacks. Their protocol relied on a black-box anomaly detection system such as traffic fingerprinting techniques (Herley & Schechter, 2019; Seo et al., 2022; Tian et al., 2020). However, the cost of false positives in detection is expensive due to the immediate denial of user access to websites. While they claimed that their protocols were able to handle 64 or 256 *responder* websites, scaling to millions or billions of users was infeasible (Pal et al., 2022). Moreover, their username-website mapping systems are problematic in negotiating the coordination of unreliable websites.

Our proposed framework predicts the potential risk of credential stuffing attacks between websites before any disruptions can occur to users, and enables the deployment of coordination protocols by selecting effective *responder* websites for given a *requester* website. Furthermore, our framework is able to process a large number of arbitrary websites with no privacy implications.

Defense of password guessing attacks. In addition to credential stuffing, password guessing has been another major branch of password attack research (Pasquini et al., 2021; Thomas et al., 2019; Wang et al., 2016b). Attackers aim to recover billions of encrypted passwords offline or submit guessable passwords to online login interfaces. To combat these attacks, researchers have proposed password strength meters that guide users to create strong passwords based on statistical prediction models (Dell'Amico & Filippone, 2015; Wheeler, 2016). Beyond addressing guessable password patterns, researchers have explored the detection of abnormal authentication traffic using traffic fingerprinting techniques (Herley & Schechter, 2019; Tian et al., 2020). These solutions have been widely adopted and enhanced password security in general. However, we need completely distinct countermeasures for credential stuffing attacks, which exploit reused passwords regardless of their guessability.

10. Conclusion

We propose PassREfinder-FL, a framework for predicting the risk of credential stuffing between websites. We introduce the concept of a password reuse relation—the relationship between websites where users are highly likely to reuse passwords—and represent these relations as edges in a website graph. To predict the existence of such edges, we design a graph neural network (GNN)-based model. Our approach is scalable to a large number of arbitrary websites by extracting public features and incorporating newly observed websites through complementary edges. To preserve privacy, we adopt a federated learning (FL) approach that enables risk prediction without sharing user sensitive information from potentially untrustworthy administrators. Evaluation on a real-world credential breach dataset demonstrates the effectiveness of our model, achieving an F1-score of 0.9153 in the FL setting. Furthermore, we show that our FL-based GNN outperforms other state-of-the-art GNN models, with performance gains of 4–11 %. By employing PassREfinder-FL, we believe that website administrators can proactively deter credential stuffing attacks by identifying password reuse risks in advance.

Acknowledgement

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2021-II210118, Development of Decentralized Consensus Composition Technology for Large-Scale Nodes, 40%, and No.RS-2023-00215700, Trustworthy Metaverse: Blockchain-Enabled Convergence Research, 30%) and the Research Grant of Kwangwoon University in 2024 (30%).

CRediT authorship contribution statement

Jaehan Kim: Conceptualization, Methodology, Writing – original draft; **Minkyoo Song:** Validation, Software; **Minjae Seo:** Writing – review & editing; **Youngjin Jin:** Writing – review & editing; **Seungwon Shin:** Project administration, Funding acquisition; **Jinwoo Kim:** Supervision, Writing – review & editing.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

AlSabah, M., Oligeri, G., & Riley, R. (2018). Your culture is in your password: An analysis of a demographically-diverse password dataset. *Computers & security*, *77*, 427–441.

Amjath, M., Henna, S., & Rathnayake, U. (2025). Graph representation federated learning for malware detection in internet of health things. *Results in Engineering*, *25*, 103651.

Apple (e). Available Trusted Root Certificates for Apple Operating Systems. https://support.apple.com/en-us/HT209143. Accessed: 2025-04-01.

Bilge, L., Han, Y., & Dell'Amico, M. (2017). Riskteller: Predicting the risk of cyber incidents. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*.

Bozkir, A. S., Dalgic, F. C., & Aydos, M. (2022). Grambeddings: A new neural network for URL based identification of phishing web pages through N-gram embeddings. *Computers & Security*, .

CVE. https://cve.mitre.org/. Accessed: 2022-11-03.

Colnago, J., Devlin, S., Oates, M., Swoopes, C., Bauer, L., Cranor, L., & Christin, N. (2018). "It'S not actually that horrible" exploring adoption of two-Factor authentication at a university. In *Proceedings of the 2018 CHI conference on human factors in computing systems*.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2019). Unsupervised Cross-lingual Representation Learning at Scale. *arXiv preprint arXiv:1911.02116*, .

Cui, J., Kim, K., Na, S. H., & Shin, S. (2022). Meta-Path-based fake news detection leveraging multi-level social context information. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 325–334).

Das, A., Bonneau, J., Caesar, M., Borisov, N., & Wang, X. (2014). The tangled web of password reuse. In *Ndss*.

Das, S. D., Basak, A., & Dutta, S. (2021). A heuristic-driven ensemble framework for COVID-19 fake news detection. In *International workshop on combating online hostile posts in regional languages during emergency situation* (pp. 164–176). Springer.

Dell'Amico, M., & Filippone, M. (2015). Monte carlo strength evaluation: Fast and reliable password checking. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*.

Face, H. (2019). xlm-roberta-base. https://huggingface.co/xlm-roberta-base. Accessed: 2022-11-03.

Gaw, S., & Felten, E. W. (2006). Password management strategies for online accounts. In *Proceedings of the second symposium on usable privacy and security*.

Golla, M., Ho, G., Lohmus, M., Pulluri, M., & Redmiles, E. M. (2021). Driving {2FA} adoption at scale: Optimizing {Two-Factor} authentication notification design patterns. In *30Th USENIX security symposium (USENIX security 21)* (pp. 109–126).

Golla, M., Wei, M., Hainline, J., Filipe, L., Dürmuth, M., Redmiles, E., & Ur, B. (2018). "What was that site doing with my facebook password?" designing password-Reuse notifications. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*.

Google Chrome (f). Chrome Root Program. https://www.chromium.org/Home/chromium-security/root-ca-policy. Accessed: 2025-04-01.

Have i been pwned? https://haveibeenpwned.com/. Accessed: 2022-07-19.

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, .

Hanamsagar, A., Woo, S. S., Kanich, C., & Mirkovic, J. (2018). Leveraging semantic transformation to investigate password habits and their causes. In *Proceedings of the 2018 CHI conference on human factors in computing systems* (pp. 1–12).

He, C., Balasubramanian, K., Ceyani, E., Yang, C., Xie, H., Sun, L., He, L., Yang, L., Yu, P. S., Rong, Y. et al. (2021). Fedgraphnn: A federated learning system and benchmark for graph neural networks. *arXiv preprint arXiv:2104.07145*, .

He, H., Ji, Y., & Huang, H. H. (2022). Illuminati: Towards explaining graph neural networks for cybersecurity analysis. In *2022 IEEE 7Th european symposium on security and privacy (euros&p)*.

He, X., & Chua, T.-S. (2017). Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (pp. 355–364).

Herley, C., & Schechter, S. E. (2019). Distinguishing attacks from legitimate authentication traffic at scale. In *Ndss*.

Hogg, R. V., Tanis, E. A., & Zimmerman, D. L. (1977). Probability and Statistical Inference (vol. 993). Macmillan New York.

Hori, C., Hori, T., Lee, T.-Y., Zhang, Z., Harsham, B., Hershey, J. R., Marks, T. K., & Sumi, K. (2017). Attention-based multimodal fusion for video description. In *Proceedings of the IEEE international conference on computer vision* (pp. 4193–4202).

Troy Hunt (g). Inside the Cit0Day Breach Collection. https://www.troyhunt.com/inside-the-cit0day-breach-collection/. Accessed: 2025-8-28.

Hutchinson, A., Tang, J., Aviv, A. J., & Story, P. (2024). Measuring the prevalence of password manager issues using in-situ experiments. In *Symposium on usable security and privacy (USEC'24)*.

IBM Security (2022). Cost of a Data Breach Report 2022. https://www.ibm.com/reports/data-breach. Accessed: 2023-03-16.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448–456). PMLR.

Khatua, A., & Nejdl, W. (2022). Unraveling social perceptions & behaviors towards migrants on twitter. In *Proceedings of the international AAAI conference on web and social media. (vol. 16)*.

Kim, H., Cui, J., Jang, E., Lee, C., Lee, Y., Chung, J.-W., & Shin, S. (2023). DRAINCLoG: Detecting Rogue Accounts with Illegally-obtained NFTs using Classifiers Learned on Graphs. *arXiv preprint arXiv:2301.13577*, .

Kim, J., Song, M., Seo, M., Jin, Y., & Shin, S. (2024). PassREfinder: Credential stuffing risk prediction by representing password reuse between websites on a graph. In *2024 IEEE Symposium on security and privacy (SP)* (pp. 1385–1404). IEEE.

Kipf, T. N., & Welling, M. (2016). Semi-supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907*, .

Konečnỳ, J., McMahan, H. B., Ramage, D., & Richtárik, P. (2016a). Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, .

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016b). Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, .

Kong, X., Zhang, W., Wang, H., Hou, M., Chen, X., Yan, X., & Das, S. K. (2024). Federated graph anomaly detection via contrastive self-supervised learning. *IEEE Transactions on Neural Networks and Learning Systems*, *36*(5), 7931–7944.

Lauter, K., Kannepalli, S., Laine, K., & Moreno, R. C. (2021). Password Monitor: Safeguarding passwords in Microsoft Edge. https://www.microsoft.com/en-us/research/blog/password-monitor-safeguarding-passwords-in-microsoft-edge/. Accessed: 2022-11-25.

Lee, S., Kim, J., Seo, M., Na, S. H., Shin, S., & Kim, J. (2024). Censor: Detecting illicit bitcoin operation via gcn-based hyperedge classification. *IEEE Access*, .

Li, L., Pal, B., Ali, J., Sullivan, N., Chatterjee, R., & Ristenpart, T. (2019). Protocols for checking compromised credentials. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*.

Liu, Y., Ao, X., Qin, Z., Chi, J., Feng, J., Yang, H., & He, Q. (2021). Pick and choose: A GNN-based imbalanced learning approach for fraud detection. In *Proceedings of the web conference 2021*.

Mao, Q., Lin, X., Xu, W., Qi, Y., Su, X., Li, G., & Li, J. (2025). Fecograph: Label-aware federated graph contrastive learning for few-shot network intrusion detection. *IEEE Transactions on Information Forensics and Security*, *20*, 2266–2280.

Mayer, P., Kirchner, J., & Volkamer, M. (2017). A second look at password composition policies in the wild: Comparing samples from 2010 and 2016. In *Thirteenth symposium on usable privacy and security (SOUPS 2017)*.

McAfee (h). Customer URL Ticketing System. https://sitelookup.mcafee.com/en/feedback/url. Accessed: 2025-04-01.

Moh, P., Yang, A., Malkin, N., & Mazurek, M. L. (2024). Understanding how people share passwords. In *Twentieth symposium on usable privacy and security (SOUPS 2024)* (pp. 219–237).

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning. In *Icml*.

Mozilla NSS (i). CA/Included Certificates. https://wiki.mozilla.org/CA/Included_Certificates. Accessed: 2025-04-01.

Pal, B., Daniel, T., Chatterjee, R., & Ristenpart, T. (2019). Beyond credential stuffing: Password similarity models using neural networks. In *2019 IEEE Symposium on security and privacy (SP)*.

Pal, B., Islam, M., Bohuk, M. S., Sullivan, N., Valenta, L., Whalen, T., Wood, C., Ristenpart, T., & Chatterjee, R. (2022). Might i get pwned: A second generation compromised credential checking service. In *31St USENIX security symposium*.

Pasquini, D., Gangwal, A., Ateniese, G., Bernaschi, M., & Conti, M. (2021). Improving password guessing via representation learning. In *2021 IEEE Symposium on security and privacy (SP)*.

Pearman, S., Thomas, J., Naeini, P. E., Habib, H., Bauer, L., Christin, N., Cranor, L. F., Egelman, S., & Forget, A. (2017). Let'S go in for a closer look: Observing passwords in their natural habitat. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*.

Sahin, S., & Li, F. (2021). Don'T forget the stuffing! revisiting the security impact of typo-Tolerant password authentication. In *Proceedings of the 2021 ACM SIGSAC conference on computer and communications security*.

Scardapane, S., Spinelli, I., & Di Lorenzo, P. (2020). Distributed training of graph convolutional networks. *IEEE Transactions on Signal and Information Processing over Networks*, *7*, 87–100.

Seo, M., Kim, J., Marin, E., You, M., Park, T., Lee, S., Shin, S., & Kim, J. (2022). Heimdallr: Fingerprinting SD-WAN control-Plane architecture via encrypted control traffic. In *Proceedings of the 38th annual computer security applications conference*.

Seo, M., Kim, J., You, M., Shin, S., & Kim, J. (2024). Gshock: A GNN-based fingerprinting system for permissioned blockchain networks over encrypted channels. *IEEE Access*, .

Shape Security (2018). 2018 Credential Spill Report. https://www.shapesecurity.com/reports/2018-credential-spill-report. Accessed: 2022-07-19.

Sherman, A. (2020). Websites ranking platform Alexa set to retire Top Sites By Category list as it introduces most popular articles ranking. https://reporter254.com/websites-ranking-platform-alexa-set-to-retire-top-sites-by-category-list-as-it-introduces-most-popular-articles-ranking/. Accessed: 2022-12-01.

Shodan (j). Shodan Developer. https://developer.shodan.io/. Accessed: 2025-04-01.

Singanamalla, S., Jang, E. H. B., Anderson, R., Kohno, T., & Heimerl, K. (2020). Accept the risk and continue: Measuring the long tail of government adoption. In *Proceedings of the ACM internet measurement conference* (pp. 577–597).

Smith, L. N., & Topin, N. (2019). Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications* (pp. 369–386). SPIE (*vol. 11006*).

Soska, K., & Christin, N. (2014). Automatically detecting vulnerable websites before they turn malicious. In *23Rd USENIX security symposium (USENIX security 14)* (pp. 625–640).

Stochastic Training on Large Graphs. https://docs.dgl.ai/en/0.8.x/guide/minibatch.html. Accessed: 2022-11-09.

Stobert, E., & Biddle, R. (2014). The password life cycle: User behaviour in managing passwords. In *10Th symposium on usable privacy and security (SOUPS 2014)*.

Tang, J., & Wang, K. (2018). Personalized top-N sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 565–573).

Tang, Y., & Liang, Y. (2024). Credit card fraud detection based on federated graph learning. *Expert Systems with Applications*, *256*, 124979.

Thomas, K., Li, F., Zand, A., Barrett, J., Ranieri, J., Invernizzi, L., Markov, Y., Comanescu, O., Eranti, V., Moscicki, A. et al. (2017). Data breaches, phishing, or malware? understanding the risks of stolen credentials. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*.

Thomas, K., Pullman, J., Yeo, K., Raghunathan, A., Kelley, P. G., Invernizzi, L., Benko, B., Pietraszek, T., Patel, S., Boneh, D. et al. (2019). Protecting accounts from credential stuffing with password breach alerting. In *28Th USENIX security symposium*.

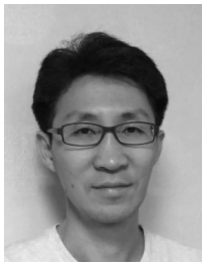
Tian, Y., Herley, C., & Schechter, S. (2020). Stopguessing: Using guessed passwords to thwart online password guessing. *IEEE Security & Privacy*, *18*(3), 38–47.
urlscan (k). urlscan.io. https://urlscan.io/. Accessed: 2025-04-01.
Vulnerability Metrics. https://nvd.nist.gov/vuln-metrics/cvss. Accessed: 2022-11-03.
Veeramachaneni, K., Arnaldo, I., Korrapati, V., Bassias, C., & Li, K. (2016). AIˆ 2: Training a big data machine to defend. In *2016 IEEE 2Nd international conference on big data security on cloud (bigdatasecurity), IEEE international conference on high performance and smart computing (HPSC), and IEEE international conference on intelligent data and security (IDS)* (pp. 49–54). IEEE.
Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph Attention Networks. *arXiv preprint arXiv:1710.10903*, .
Verizon (2022). Data Breach Investigations Report. https://www.verizon.com/business/resources/reports/dbir/. Accessed: 2022-07-19.
Wang, C., Jan, S. T. K., Hu, H., Bossart, D., & Wang, G. (2018). The next domino to fall: Empirical analysis of user passwords across online services. In *Proceedings of the eighth ACM conference on data and application security and privacy*.
Wang, D., He, D., Cheng, H., & Wang, P. (2016a). FuzzyPSM: A new password strength meter using fuzzy probabilistic context-free grammars. In *2016 46Th annual IEEE/IFIP international conference on dependable systems and networks (DSN)* (pp. 595–606). IEEE.
Wang, D., Shan, X., Dong, Q., Shen, Y., & Jia, C. (2023a). No single silver bullet: Measuring the accuracy of password strength meters. In *32Nd USENIX security symposium (USENIX security 23)* (pp. 947–964).
Wang, D., Zhang, Z., Wang, P., Yan, J., & Huang, X. (2016b). Targeted online password guessing: An underestimated threat. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*.
Wang, D., Zou, Y., Xiao, Y.-A., Ma, S., & Chen, X. (2023b). Pass2edit: A multi-step generative model for guessing edited passwords. In *32Nd USENIX security symposium (USENIX security 23)* (pp. 983–1000).
Wang, K. C., & Reiter, M. K. (2019). How to end password reuse on the web. In *26Th annual network and distributed system security symposium (NDSS 2019)*. Internet Society.
Wang, K. C., & Reiter, M. K. (2020). Detecting stuffing of a user's credentials at her own accounts. In *29Th USENIX security symposium (USENIX security 20)*.
Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y. et al. (2019). Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks. *arXiv preprint arXiv:1909.01315*, .
Wash, R., Rader, E., Berman, R., & Wellmer, Z. (2016). Understanding password choices: How frequently entered passwords are re-used across websites. In *Twelfth symposium on usable privacy and security (SOUPS 2016)*.
Wayback Machine (l). Internet Archive: cit0day.in. https://web.archive.org/web/20200926134818/https://cit0day.in/. Accessed: 2025-8-28.
Wheeler, D. L. (2016). Zxcvbn: Low-Budget password strength estimation. In *Usenix security symposium* (pp. 157–173).
Wu, C., Wu, F., Cao, Y., Huang, Y., & Xie, X. (2021). Fedgnn: Federated graph neural network for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925*, .
Xu, M., Yu, J., Zhang, X., Wang, C., Zhang, S., Wu, H., & Han, W. (2023). Improving real-world password guessing attacks via bi-directional transformers. In *32Nd USENIX security symposium (USENIX security 23)* (pp. 1001–1018).
Yang, P., Zhao, G., & Zeng, P. (2019). Phishing website detection based on multidimensional features driven by deep learning. *IEEE Access*, *7*, 15196–15209.
Yang, Z., Pei, W., Chen, M., & Yue, C. (2022). WTAGRAPH: Web tracking and advertising detection using graph neural networks. In *Ieee symposium on security and privacy*.
Yao, Y., Jin, W., Ravi, S., & Joe-Wong, C. (2023). FedGCN: Convergence-communication tradeoffs in federated training of graph convolutional networks. *Advances in neural information processing systems*, *36*, 79748–79760.
Zhang, K., Yang, C., Li, X., Sun, L., & Yiu, S. M. (2021). Subgraph federated learning with missing neighbor generation. *Advances in neural information processing systems*, *34*, 6671–6682.
Zhang, Y., Dong, X., Shang, L., Zhang, D., & Wang, D. (2020). A multi-modal graph neural network approach to traffic risk forecasting in smart urban sensing. In *2020 17Th annual IEEE international conference on sensing, communication, and networking*.

Jaehan Kim is a PhD candidate in the School of Electrical Engineering at KAIST. He received his M.S. degree and B.S. degree in School of Electrical Engineering at KAIST. His current research interests mainly focus on AI security, particularly on the safety of large language models (LLMs), and AI-driven solutions for cybersecurity.

Minkyoo Song is a PhD candidate in the School of Electrical Engineering at KAIST. He received his M.S. degree in Electrical Engineering at KAIST and his B.S. degree in Industrial Engineering and double majored in Electrical Engineering at KAIST. His current research interests mainly focus on Large language model security, AI security, and Data mining.

Minjae Seo is a Researcher at ETRI, Daejeon, South Korea. He received his M.S. degree from the Graduate School of Information Security at KAIST and his B.S. degree in Computer Engineering from Mississippi State University. His current research interests include network fingerprinting, deep learning-based network system, and AI for security.

Youngjin Jin is a PhD candidate in the School of Electrical Engineering at KAIST. He received his M.S. degree in Electrical Engineering at KAIST and his B.S. degree in Electrical Engineering with a minor in Computer Science at KAIST. His current research interests are on large language model security, with a focus on applications on cybersecurity and cyber threat intelligence.



Seungwon Shin is an Associate Professor in the School of Electrical Engineering at KAIST and an Executive Vice President at Samsung Electronics. He received his Ph.D. in Computer Engineering from the Electrical and Computer Engineering Department at Texas A&M University and his M.S. degree and B.S. degree from KAIST, both in Electrical and Computer Engineering. His research interests span the areas of Software-defined networking security, DarkWeb analysis, and Cyber threat intelligence.

Jinwoo Kim is an Assistant Professor in the School of Software at Kwangwoon University, Seoul, Republic of Korea. He received his Ph.D. from the School of Electrical Engineering at KAIST, his M.S. degree from the Graduate School of Information Security at KAIST, and his B.S. degree from Chungnam National University in Computer Science and Engineering. His research topic focuses on investigating security and privacy issues with software-defined networks, cloud systems, and VR.