PASSREFINDER: Credential Stuffing Risk Prediction by Representing Password Reuse between Websites on a Graph

Jaehan Kim, Minkyoo Song, Minjae Seo, Youngjin Jin, Seungwon Shin
KAIST

***Abstract*—The prevalence of credential stuffing has caused devastating harm to online users who tend to reuse passwords across websites. In response, researchers have made efforts to detect users who set the same passwords or malicious logins. However, existing detection methods sacrifice the usability of passwords by inhibiting password creation or website access. Moreover, the complicated mechanisms for sharing account information hinder their deployment in practice. In this work, we propose a risk prediction framework to prevent credential stuffing attacks before disrupting user behaviors rather than relying on detection. To this end, we newly define the relationship between websites in which users are highly likely to reuse passwords and represent it as an edge on a website graph using graph neural networks. We then perform a link prediction task to identify the risk of credential stuffing between websites. Our framework is applicable to a large number of arbitrary websites by utilizing public website information and linking newly observed website nodes to the graph. The evaluation on a real-world credential dataset consisting of 360 million accounts breached from 22,378 websites shows that our model successfully predicts credential stuffing risk among websites by achieving F1-scores of 0.9559 and 0.9100 in two different graph learning settings, respectively. In addition, we demonstrate the effectiveness of each design strategy and validate that the prediction results can be utilized to quantify the expected rates of password reuse as risk scores.**

***Index Terms*—Password authentication, Credential stuffing, Graph neural network**

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 on the Internet [1]. 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 [2]. 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 [3]–[7]. 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 [8].

To protect users from the risk of credential stuffing, security practitioners have provided third-party services called compromised credential checking (C3) services [9] including Have I Been Pwned (HIBP) [10], Google Password Checkup [11]. 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 data breaches. As an alternative, a method of coordinating websites was proposed to detect the creation of the same password [12] or suspicious logins for credential stuffing attacks [13] 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 [14], and the requirement of collecting usernames to map each username to websites discourages the participation of websites.

As such, existing approaches have clear limitations (e.g., relying on known information, sharing sensitive information, and limited scalability), and to address these limitations, we change the viewpoint of credential stuffing detection into a risk prediction task. To this end, we propose PASSREFINDER, a framework to proactively predict the risk of credential stuffing by estimating the rate of password reuse between websites before disrupting user behaviors, so that the cost of false positives is lower than that of the detection task. In addition, we aim to minimize privacy implications by utilizing public website features. Indeed, the correlation between website features and users' tendency to reuse passwords has been repeatedly stated in previous studies [4], [5], [7], [11], [15], [16]. For example, users tend to reuse passwords within the same category of websites or consider the security levels of websites to decide to reuse passwords. Motivated by these insights, we interpret the website features and their implicit influences on the tendency of password reuse through a deep learning approach to draw a model for predicting the risk of credential stuffing between websites. In this regard, we define a *password reuse relation* to present the relationship between websites in which users are highly likely to reuse their passwords. Then, we attempt to predict the existence of a password reuse relation between websites.

To accomplish this goal, we design a new framework

while addressing *three* technical challenges. i) The password reuse relations among a large number of websites are highly complicated because a user's password choice for a given website is determined by considering the existing passwords used in other websites. As a solution, we employ a graph structure to represent a password reuse relation as an edge connecting website nodes based on graph neural networks (GNNs), while benefiting from the propagation and aggregation of complicated neighborhood influences. ii) The diverse data structures of the website features and their different influences on password reuse relations make it difficult for neural networks to model the features. Therefore, we adopt multi-modal learning [17] and attention mechanism techniques [18], [19], which enable the model to interpret each feature separately and reflect the importance of the feature. iii) The graph-based model is hard to learn effective representations related to newly observed website nodes because the nodes are not linked to the graph. To make the prediction model applicable to arbitrary websites, we introduce a novel method of connecting complementary edges called *hidden relations* that can implicitly propagate the information of password reuse relations. In addition, we provide practical strategies to deploy hidden relations in a manner that preserves user privacy.

We evaluate PASSREFINDER by conducting extensive experiments using a large collection of credential data breach datasets consisting of 360 million accounts breached from 22,378 websites. In doing so, we demonstrate that our prediction model can effectively predict password reuse relations among websites. We present the performances on two different graph learning settings (i.e., transductive and inductive settings) following the two working scenarios for different needs in utilizing our framework. Our prediction model greatly outperforms the other baselines, such as a traditional machine learning model, by achieving F1-scores of 0.9559 and 0.9100 in the transductive and inductive settings, respectively. Specifically, we obtain up to 16-24% performance gains from that of the state-of-art GNN models. In the ablation study, we investigate the effectiveness of each component in our model by eliminating each of the website features or architectural methods adopted to address the aforementioned challenges. The result shows that all components of the model are crucial, but the contributions of the architectural methods differ in the two learning settings. More specifically, the method of hidden relation improves the prediction performance by 11% in the inductive setting, where the evaluation entails the newly observed websites, compared to just a 2% gain observed in the transductive setting. In addition, we validate that our prediction results directly quantify the expected rates of password reuse among websites, which can serve as risk scores of credential stuffing. The overall results show that our model learns effective edge representations for predicting password reuse relations on the graph of a large number of website nodes.

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, 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 evaluate our framework on a large-scale credential dataset and show its outstanding performance with F1-scores of 0.9559 and 0.9100 in transductive and inductive graph learning settings, respectively.
- We show that our prediction results can be used to quantitatively measure credential stuffing risk scores among websites.
- We provide three realistic implementations of PASSREFINDER application scenarios in the perspective of administrators to mitigate credential stuffing.

2. Background and Motivation

Here, we first provide the background information and related studies, and we describe the motivation of our work.

Credential stuffing detection. Researchers have striven to prevent credential stuffing by detecting password reuse. One of the most common techniques is a compromised credential checking (C3) service [9], [11], [14], such as Have I Been Pwned (HIBP) [10]. 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 [20]. Hence, C3 services cannot provide timely coping with exploiting new accounts from undisclosed data breaches.

As another type of credential stuffing detection, Wang et al. [12] 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 [13] 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 [14]. 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 pass-

word reuse have been conducted. Specifically, analyzing features affecting password choice and reuse has been one of the main topics [4], [5], [7], [11], [15], [16], [21]–[23]. For example, Alsabah et al. [21] and Mayer et al. [22] claimed that users significantly considered their languages and websites' password composition policies to create their passwords. Pearman et al. [5] 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 [7], [11], [15]. Wash et al. [4] warned that university students were likely to reuse their passwords in other university-related services. Stobert et al. [16] showed that some users carefully consider the gap between the security aspects of two websites when they reused their passwords across websites.

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) [24] performs convolution operations in graphs based on message passing and neighborhood aggregation to represent the hidden features of nodes. GraphSAGE [25] 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 [18] 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 [26]–[29].

Our approach: PASSREFINDER. In order to overcome the limitations of existing credential stuffing detection methods, we shift our viewpoint from detection to risk prediction to help users who are interrupted by the existing detection methods. We preemptively predict the risk of credential stuffing attacks by estimating the rate of password reuse between specific websites. We define a *password reuse relation* between websites to present the relationship where the rate of password reuse between websites is higher than a given threshold. Our goal is to predict the existence of a password reuse relation between two given websites. To this end, we model public website features without sharing private user information and without adopting any sophisticated encryption method. Therefore, it has no privacy implications and will be applicable to a large number of arbitrary websites. However, interpreting complicated aspects of users' password reuse tendency among multiple websites is a demanding task. We tackle this technical problem by representing the password reuse relation as an edge on a graph of website nodes (see the first and second steps in Figure 1) while benefiting from the ability of GNNs in a graph learning task. We design a prediction model using GNN layers tailored to perform a link prediction (i.e., edge prediction) task to predict the existence of password reuse relation edges among arbitrary website nodes at scale.

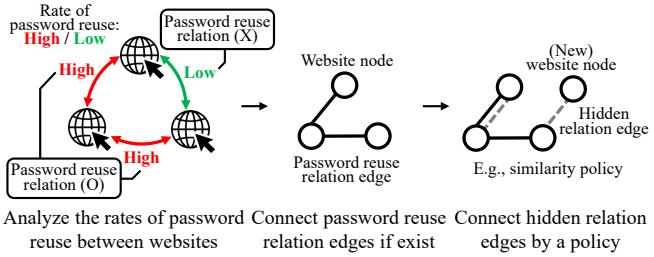


Figure 1: Graph representation of password reuse relations.

3. Technical Challenges

In this section, we present three challenges posed when designing PASSREFINDER to effectively predict password reuse relations and introduce methods to tackle 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 considering the users' passwords already used in other websites. Thus, we 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 [4], [5]. 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 [4], [5], [7], [11], [15], [16]. 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* [17] and attention mechanisms [18], [19] 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, a website administrator may want to predict the password reuse relation between two websites, one or both of which are newly discovered or created (i.e., no password reuse information with any other websites that they manage). Since these *new* website nodes are not able to be connected with others by password reuse relation edges, it is difficult to achieve reasonable performance in predicting password reuse relations related to the new websites on a graph. To address this problem, we propose a novel method of linking new website nodes using complementary edges called *hidden relations*. Hidden relations link arbitrary websites without any information about password reuse relations. The hidden relations are connected according to a *hidden relation policy*, which is an input for the framework (see Figure 1). We expect the hidden relation edges to implicitly propagate information related to password reuse between websites.

4. Problem Definition

The goal of PASSREFINDER is to predict the existence of a password reuse relation — a relation between websites where users are highly likely to reuse passwords. Formally, we 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.

Password reuse graph. Password reuse relations can be represented on a graph of websites. We formulate a *password reuse graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \hat{\mathcal{E}})$ to model the existence of password reuse relations among websites. We denote a set of website nodes as $\mathcal{V}$, a set of password reuse relation edges as $\mathcal{E}$, and a set of hidden relation edges as $\hat{\mathcal{E}}$ following a policy $\mathcal{P}$. Website nodes $u, v \in \mathcal{V}$ are connected by a password reuse relation edge $e_{uv} \in \mathcal{E}$ if a password reuse relation exists between u and v. They are also connected by the other type of edge $\hat{e}_{uv} \in \hat{\mathcal{E}}$ if a hidden 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.

Multi-modal website features. 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 $x_v = \{\mathbf{x}_v^1, \mathbf{x}_v^2, ..., \mathbf{x}_v^M\}$ be a feature set of website node v. 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. We present the details of feature extraction in Section 5.2.2.

Edge representation and task. We aim to predict the existence of a password reuse relation between two given websites by conducting edge representation and link prediction tasks in turn. Given website features $\{x_v : v \in \mathcal{V}\}$, a hidden relation policy $\mathcal{P}$, and a password reuse graph $\mathcal{G}$, the ultimate goal of our model is to generate 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 password reuse relations between websites.

5. Design

In this section, we introduce the overall architecture of PASSREFINDER and provide detailed design strategies for the prediction model to tackle the technical challenges.

5.1. Overall Architecture

We design each component of our framework according to the formulation of the problem (see Figure 2). We aim to address the aforementioned technical challenges by leveraging a series of design strategies to obtain effective edge representations from a password reuse graph.

In *graph construction*, we first build a password reuse graph from a given website dataset. The website nodes are connected by two distinct types of edges if password reuse relations and hidden relations exist between the websites, respectively. Regarding the design choice of constructing a password reuse graph, we provide two learning settings for website administrators who have different needs in utilizing our framework: i) In the *transductive setting*, we construct a graph based on the entire dataset. However, parts of the password reuse relation edges are unlabeled (i.e., disconnected), which are the targets of the link prediction task. ii) The *inductive setting* is a proposed solution for applying the graph-based model to newly observed websites. In this setting, we build a graph only by using websites in the train dataset. The *new* website nodes in the test dataset are connected to the existing graph built during the training process through the hidden relation edges.

In *feature extraction*, we extract 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 [30]–[35]. 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.

In *edge representation learning*, we utilize a GNN-based model tailored to producing representations of password reuse relation edges for predicting their existences. We design a novel neighbor aggregation function Agg to deal with the two types of relations (i.e., password reuse relation and hidden relation). For neighbor aggregation, we adopt the technique of neighborhood attention to assign different importance to neighbor nodes. We can consider

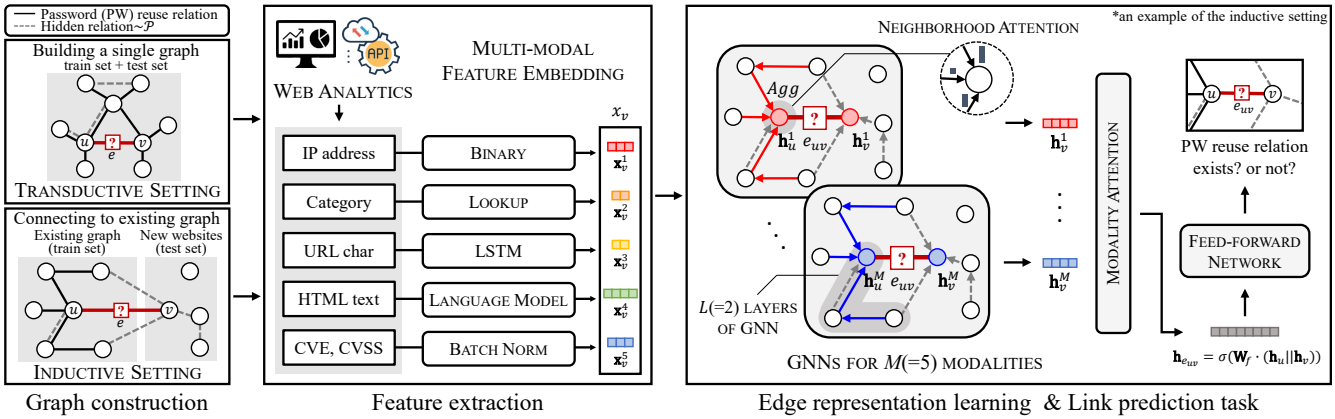


Figure 2: Overview of PASSREFINDER.

1 to L-hop neighborhood by leveraging L number of GNN layers for each of the M modalities of website features. In addition, multi-modality attention is applied to M number of representation vectors of each node to reflect different contributions from features. Then, we obtain the final edge representation by concatenating the two node representation vectors. During *link prediction*, we predict the existence of a password reuse relation edge between two given website nodes using the output of a feed-forward network applied to the edge representation vector. If so, it implies that a password reuse relation exists between the websites.

5.2. Password Reuse Relation Prediction

We describe the architecture of our framework and specific design strategies in more detail to tackle the aforementioned technical challenges from the perspective of a website administrator who utilizes the framework.

5.2.1. Graph Construction. A password reuse graph can be built from a dataset of websites managed by a website administrator who may have partial information on password reuse relations among websites. Specifically, the password reuse graph is constructed using website nodes and two types of edges representing password reuse relations and hidden relations, respectively.

The method of connecting the edges is as follows: first, we connect password reuse relation edges between website nodes if the rate of users who reuse their passwords is higher than the input ground truth threshold τ, and next, we connect hidden relation edges according to an input policy $\mathcal{P}$. We note that the hidden relation edges are not the targets of the prediction task but inputs for improving the performance of the prediction model. A hidden relation policy should be able to connect arbitrary website nodes without the information of password reuse relations among them. Through this approach, newly observed website nodes are linked to the graph, allowing the model to learn implicit information regarding password reuse propagated from the new website nodes. In our prototype, we provide three types of hidden relation policies: a *shared user policy* connecting edges if at least n users are shared, a *similarity policy* connecting edges if the cosine similarity of website features averaged over modalities is higher than ρ, and a *probabilistic policy* connecting edges with a specific probability p.

We provide two learning settings in adopting our framework to meet the different needs of the website administrators: one for predicting password reuse relations among managed websites (i.e., transductive setting) and the other for predicting password reuse relations for websites outside the scope of management (i.e., inductive setting). In the transductive setting, all the websites in the provided dataset are utilized to build a password reuse graph. We split the password reuse relation edges in the constructed graph into the train and test edge datasets. Then, we train the model utilizing the train edges, which are for already known password reuse relations and predict the existence of unlabeled edges in the test dataset. This setting is for website administrators who wish to predict unknown password reuse relations between two websites under their management. The website administrators can extend the information of password reuse relations by fully utilizing the given website dataset. On the other hand, in the inductive setting, we split the websites of the dataset into the train and test node datasets. We build a password reuse graph only from the websites of the train node dataset, which the website administrators can observe in practice. During the testing process, we connect the newly observed website nodes in the test dataset to the existing password reuse graph using only the hidden relation edges. Then, we predict the existence of the unseen password reuse relation edges from new nodes. In the inductive setting, website administrators can predict the password reuse relation between a new website (e.g., websites managed by other administrators) and a website under their management or even between two new websites.

5.2.2. Feature Extraction. 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. 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 [21], [22]. 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* [30], 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 [5], [7], [11], [15]. 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). We utilize the categorization service of McAfee [31]. 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 [36], [37].

Content complements the category feature as website category is sometimes ambiguous [38] (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 [39], which is a powerful model for tasks in various domains [40]–[42]. We utilize the embedding vectors obtained from the pre-trained model [43].

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) [4]. We adopt a method of character-level embedding for URL using Long Short-Term Memory (LSTM), modified from an existing URL embedding approach [44].

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 [15], [16]. We conduct a security analysis on website systems through *Shodan* [32], a search engine for Internet devices, and extract software vulnerabilities in a Common Vulnerabilities and Exposures (CVE) [45] 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 [46] 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 [33], Mozilla Firefox [35], and Apple macOS [34] 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 [47]. To stabilize the neural network training process while managing diverse scales of security posture features, we adopt batch normalization techniques [48].

5.2.3. Edge Representation Learning. We introduce GNNs tailored to learning the edge representations of password reuse relations on a password reuse graph. In particular, we present a novel aggregation function and two different levels of attention mechanisms, which enable us to deal with various website features and two distinct types of edges (i.e., password reuse relations and hidden relations).

We describe the procedure for learning the edge representations in Algorithm 1. The main inputs of the algorithm include a password reuse graph $\mathcal{G}$ and a node feature set x_v in M modalities for each node. The number of GNN layers is defined by L. Then, the algorithm takes $M \times L$ number of uniformly initialized weight matrices $\mathbf{W}^{m,l}$ for each modality and each GNN layer. The algorithm also takes in another weight matrix $\mathbf{W}_f$ and an activation function σ to produce the final edge representation vector. Specifically, two types of neighborhood functions $\mathcal{N}_t$, which return the neighbors of a given node, are required to deal with both types of edges: password reuse relation edges $\mathcal{E}$ and hidden relation edges $\hat{\mathcal{E}}$. Lastly, the neighbor aggregation and the modality attention functions that we defined are taken as inputs for the algorithm. Our goal is to obtain the vector representation $\mathbf{h}_{e_{uv}}$ for each password reuse relation edge.

Feature initialization. In lines 1 to 2, the algorithm begins by initializing the node representation vectors. For each modality m and each edge type t, we assign the input feature $\mathbf{x}_v^m$ to the node representation vector of the initial layer $\mathbf{h}_{v,t}^{m,0}$.

Neighbor aggregation. For each GNN layer l (and for each modality m), the node v's representation vector is produced by aggregating that of its neighbors in lines 4 to 10. The aggregation function is independently run by each edge type t with neighbors determined by the neighborhood function $\mathcal{N}_t$. To prioritize neighbors, we adopt neighborhood attention during the aggregation. It takes the node representation vector $\mathbf{h}_{v,t}^{m,l-1}$ and each neighbor u's node representation vector $\mathbf{h}_{u,t}^{m,l-1}$ 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 of type t. The attention coefficient is formulated as follows:

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

where $\mathbf{W}_t^{m,1} \in \mathbb{R}^{D_m \times d}$ and $\mathbf{W}_t^{m,l(>1)} \in \mathbb{R}^{d \times d}$ 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:

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

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

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

After the aggregation on each node, we impose different weights to the node representation vector $\mathbf{h}_{v,t}^{m,l-1}$ and the aggregated neighborhood vector $\mathbf{h}_{\mathcal{N}_t(v),t}^{m,l}$ by concatenating the two vectors and multiplying its result by the weight matrix $\mathbf{W}^{m,l}$, which is effective for inductive learning [25] (*cf.* adding them in GCN [24]). By taking the average of the node representation vectors over the two edge types, 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 in line 12, we apply the modality attention function across the node representation vectors of M modalities in line 14. The normalized modality attention coefficients are formalized as follows:

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

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:

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

Output representation. In line 15, we concatenate the representation vectors of two nodes u and v. Then, we obtain the edge representation vector $\mathbf{h}_{e_{uv}}$ by applying the weight matrix and the activation function in turn.

5.2.4. Link Prediction Task. To predict the existence of a password reuse relation between two websites on a password reuse graph, we perform a link prediction task by learning the representation of the edge between the website nodes.

Algorithm 1 Edge representation learning

Input: Password reuse graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \hat{\mathcal{E}})$;
the number of modalities M;
the number of GNN layers L;
input node feature set $x_v = \{\mathbf{x}_v^1, \mathbf{x}_v^2, ..., \mathbf{x}_v^M\}$;
weight matrices $\mathbf{W}^{m,l}, \forall m \in \{1, ..., M\}, \forall l \in \{1, ..., L\}$;
weight matrix for edge representation $\mathbf{W}_f$;
activation function σ;
neighborhood functions $\mathcal{N}_t : v \to 2^{\mathcal{V}}, \forall t \in \{\mathcal{E}, \hat{\mathcal{E}}\}$;
neighbor aggregation function AGG;
modality attention function MODALITYATTN
Output: vector representations $\mathbf{h}_{e_{uv}}, \forall e_{uv} \in \mathcal{E}$
1: **for** $m = 1...M$ **do**
2: $\quad \mathbf{h}_{v,t}^{m,0} \leftarrow \mathbf{x}_v^m, \forall v \in \mathcal{V}, \forall t \in \{\mathcal{E}, \hat{\mathcal{E}}\}$
3: $\quad$ **for** $l = 1...L$ **do**
4: $\quad\quad$ **for** $v \in \mathcal{V}$ **do**
5: $\quad\quad\quad$ **for** edge type $t \in \{\mathcal{E}, \hat{\mathcal{E}}\}$ **do**
6: $\quad\quad\quad\quad \mathbf{h}_{\mathcal{N}_t(v),t}^{m,l} \leftarrow \text{AGG}(\mathbf{h}_{u,t}^{m,l-1}, \forall u \in \mathcal{N}_t(v))$
7: $\quad\quad\quad\quad \mathbf{h}_{v,t}^{m,l} \leftarrow \sigma(\mathbf{W}^{m,l} \cdot \text{CONCAT}(\mathbf{h}_{v,t}^{m,l-1}, \mathbf{h}_{\mathcal{N}_t(v),t}^{m,l}))$
8: $\quad\quad\quad$ **end for**
9: $\quad\quad\quad \mathbf{h}_v^{m,l} \leftarrow \text{AVERAGE}(\mathbf{h}_{v,\mathcal{E}}^{m,l}, \mathbf{h}_{v,\hat{\mathcal{E}}}^{m,l})$
10: $\quad\quad$ **end for**
11: $\quad$ **end for**
12: $\quad \mathbf{h}_v^m \leftarrow \mathbf{h}_v^{m,L} / \|\mathbf{h}_v^{m,L}\|_2, \forall v \in \mathcal{V}$
13: **end for**
14: $\mathbf{h}_v \leftarrow \text{MODALITYATTN}(\mathbf{h}_v^m, \forall m \in \{1, ..., M\}), \forall v \in \mathcal{V}$
15: $\mathbf{h}_{e_{uv}} \leftarrow \sigma(\mathbf{W}_f \cdot \text{CONCAT}(\mathbf{h}_u, \mathbf{h}_v)), \forall e_{uv} \in \mathcal{E}$

We apply a feed-forward network f and a *sigmoid* activation function on the edge representation vector $\mathbf{h}_{e_{uv}}$ to obtain the probability on the existence of the edge $\hat{p}_{e_{uv}}$:

$$\hat{p}_{e_{uv}} = \textit{sigmoid}(f(\mathbf{h}_{e_{uv}})). \quad (6)$$

As a consequence, a password reuse relation exists if and only if $\hat{p}_{e_{uv}} \geq \tau$, the ground truth threshold. In our prototype, we simply choose a fully connected layer as the feed-forward network f. For training, we minimize the binary cross entropy loss of the predicted probability $\hat{p}_{e_{uv}}$ and the ground truth $y_{e_{uv}} \in \{0, 1\}$:

$$\mathcal{L} = -\frac{1}{|\mathcal{E}|} \sum_{e_{uv} \in \mathcal{E}} y_{e_{uv}} \log \hat{p}_{e_{uv}} + (1 - y_{e_{uv}}) \log(1 - \hat{p}_{e_{uv}}). \quad (7)$$

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 [49]). 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 using Deep Graph Learning (DGL) [50],

a library providing deep learning modules for graphs. We provide a replication package including the trained classifier, model structure, and an anonymized sample dataset. Access to the release will be granted by the authors upon request for research purposes to address ethical concerns[1].

5.2.5. Privacy-preserving Deployment of Hidden Relation Policy. Linking newly observed websites to password reuse graphs is key to the practical deployment of our framework. In this sense, it is crucial to adopt an effective hidden relation policy to accurately predict password reuse relations between arbitrary websites while maintaining privacy. Therefore, we provide practical strategies for deploying three hidden relation policies: *the shared user policy*, *the similarity policy*, and *the probabilistic policy*.

First, administrators who adopt shared user policy need to investigate shared users between websites. To securely achieve this information, administrators can utilize a privacy-preserving protocol such as private set intersection (PSI) [51], [52]. Each website creates a list of hashed usernames and computes the intersection between the encrypted lists, so that user information cannot be identified/recovered. We simulate the PSI protocol on 9,927 websites in our dataset as described in Section 6.1.1. In cross-validation, an administrator can add one website into the password reuse graph of 9,926 websites within 13.4 seconds on average. This one-time overhead during graph construction has minor impact on scalability. Notably, our framework never shares passwords or matches them to usernames.

Alternatively, we can completely eliminate privacy concerns by using similarity or probabilistic policy. The similarity policy leverages public website features obtained through the feature extraction process, and the probabilistic policy requires neither website nor user information. Hence, administrators can implement these hidden relation policies without resorting to privacy-preserving methods.

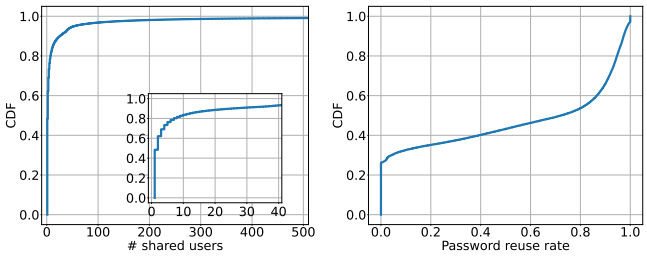
6. Evaluation

In this section, we present our experiment setup on a real-world dataset and the evaluation results of PASSREFINDER. The evaluation includes an ablation study and a risk score analysis as well as the main prediction results.

6.1. Experiment Setup

6.1.1. Dataset. We adopt the `Cit0day` [53] 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.

1. Available upon request at `jaehan@kaist.ac.kr`



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

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

TABLE 1: Dataset statistics. We provide the number of edges in each split set. We also present the ground truth labels of the total edges given $\tau = 0.5$. In the inductive setting, we split nodes instead of edges, and there are no edges between nodes in the validation set and the test set.

	Transductive	Inductive (nodes)
Train set	363,583	209,190 (5,956)
Validation set	121,194	176,405 (1,985)
Test set	121,194	168,644 (1,986)
Total	605,971	554,239 (9,927)
(Positive / Negative)	(344,946 / 261,025)	(317,759 / 236,480)

A single user is indicated by a unique email address (a commonly used user identifier in password analyses [3], [6], [12], [13]). We note that email addresses are entirely anonymized to address ethical considerations. The number of shared users between websites ranges from 0 to 733,300 with most within 500 (see Figure 3a). 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 [54]. We filter out websites that share fewer than 30 users with every other website, leaving 9,927 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 Figure 3b). Since there is no standard for measuring credential stuffing risk, we set the threshold τ to 0.5 in the main experiment. We additionally provide analyses for selecting thresholds of the edge definition and the ground truth in Appendix A.1.

Our finalized dataset for the experiments includes 9,927 website nodes connected by 605,971 password reuse relation edges. The statistics of the dataset for the experiments are described in Table 1. For the training and testing processes, we split the edges into the train set (60%), validation set (20%), and test set (20%) for the transductive setting and split the website nodes into the train (60%), validation (20%), test (20%) sets for the inductive setting to simulate the real-world scenarios of the two learning settings, respectively, as discussed in Section 5.2.1.

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 set a parameter of each hidden relation policy: the shared user policy ($User_{n=30}$) – connecting edges if at least 30 shared users exist for statistical significance; the similarity policy ($Sim_{\rho=avg}$) – connecting edges if the feature similarity is higher than the average of the train set (0.81 in both settings, values normalized into the range [0, 1]); and the probabilistic policy ($Prob_{p=1}$) – connecting edges with a probability of 1. We discuss the impact of hidden relation policy parameters on performance in Appendix A.1. The shared user policy is used as the primary model in our evaluation. For training, we apply a dropout layer with a rate of 0.5 to the output of each GNN layer to mitigate overfitting. 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 [55] 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. In order to show the effectiveness of our framework, we compare our prediction model to traditional machine learning models, a simple neural network, and GNN models, which have been frequently used in risk prediction tasks [56]–[60]. For the baselines, we utilize the concatenation of all feature modalities as input.

- **Similarity classifier** is a simple baseline using the cosine similarity between the feature vectors of two given websites. A password reuse relation exists if and only if the similarity is higher than the average of the train set. We exclude categorical features such as the category and URL of a website and normalize the remaining features into the range of $[0, 1]$ before calculating the similarity.
- **Logistic regression** is a representative linear machine learning classifier. As the embedding layers are unavailable, we encode the category of a website to a single integer and the URL into an integer vector of size 20, which indicates the IDs of the category and the URL characters, respectively. The other features are embedded identically as in our methods.
- **XGBoost** [61] is a tree-based gradient boosting model which has been successful in various classification tasks. For the inputs, we encode the category and the URL of a website in the same manner as logistic regression.
- **Multi-layer perceptron (MLP)** composed of $L = 2$ fully connected layers is used to evaluate a simple neural network without any information of the graph structures.
- **Graph convolutional network (GCN)** introduced by Kipf et al. [24] 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.
- **GraphSAGE** [25] 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.
- **Graph attention network (GAT)** [18] is a GCN variant adopting self-attentional neighbor aggregation for selectively learning the influences of important neighbors.

In addition, we conduct experiments on the GNN models (i.e., GCN, GraphSAGE, and GAT) with multi-modality to provide an explicit comparison of different GNN architectures. In other words, we use $L = 2$ GNN layers for each feature modality instead of feeding the concatenation of the features into unified GNN layers.

6.2. Results

6.2.1. Prediction Performance. The performance of our prediction model and a comparison with the baselines are summarized in Table 2. In the cases of baselines that are not graph-based, the experimental environments in the transductive and inductive settings are technically the same, but we provide both results because the split datasets are different.

Our framework shows outstanding prediction performance while the primary model adopting the shared user policy ($User_{30}$) achieves F1-scores of 0.9559 and 0.9100 in the transductive and inductive settings, respectively. The performance of our model surpasses that of the similarity classifier by a considerable margin, which shows that effective representation of password reuse relations is challenging when relying solely on the similarity between websites. Our model also significantly outperforms traditional machine learning models and MLP. Therefore, we are confident that the graph-based modeling of website features is powerful for predicting password reuse relations. Remarkably, our model also outperforms state-of-the-art GNN models in both learning settings. This is due to the proposed neighbor aggregation and the method of multiple feature modalities. In the transductive setting, the GNN models including ours show higher performance than in the inductive setting because the former allows us to fully utilize the password reuse information of the given website dataset, while the latter provides no information of password reuse relations regarding the *new* website nodes in the test set for the model.

In more detail, we observe noticeable gains compared to GCN in the inductive setting. A possible explanation is that the concatenation approach for self-embedding in the neighbor aggregation of each GNN layer enhances the

TABLE 2: Overall prediction performance.

Model	Multi-modality	Transductive setting			Inductive setting		
		Precision	Recall	F1-score	Precision	Recall	F1-score
Similarity classfier		0.6038	0.6807	0.6399	0.6128	0.6875	0.6480
Logistic regression		0.6485	0.8107	0.7206	0.6291	0.8262	0.7143
XGBoost		0.6911	0.8673	0.7692	0.6559	0.8489	0.7400
MLP		0.6163	0.7961	0.6948	0.5876	0.8972	0.7101
GCN		0.9160	0.8984	0.9071	0.6689	0.8068	0.7314
GraphSAGE		0.9265	0.8883	0.9070	0.7051	0.8292	0.7622
GAT		**0.9706**	0.8804	0.9233	0.7996	0.7745	0.7868
GCN	✓	0.9578	0.9067	0.9315	0.6970	0.7932	0.7420
GraphSAGE	✓	0.9266	0.9319	0.9292	0.7765	0.8065	0.7912
GAT	✓	0.9602	0.9108	0.9348	0.7714	0.9189	0.8387
PASSREFINDER$_{P_1}$	✓	0.9634	0.9367	0.9499	0.7937	0.9138	0.8495
PASSREFINDER$_{Sim_{avg}}$	✓	0.9480	0.9508	0.9494	0.8805	**0.9270**	0.9031
PASSREFINDER$_{User_{30}}$	✓	0.9566	**0.9551**	**0.9559**	**0.8941**	0.9265	**0.9100**

performance of inductive graph learning. This is supported by our evaluation result that GraphSAGE is apparently better than GCN in the inductive setting. GAT shows the best performance among the baselines. However, our model improves most metrics in both learning settings by benefiting from the concatenation of self-embedding and neighborhood attention together. Notably, we put more focus on recall than precision because higher recall implies that fewer risky website relationships are overlooked. Even though the precisions of GNN models (e.g., GAT) in the transductive setting are noteworthy, they mostly show poor levels of recall (maximum of 0.8984). On the other hand, our model in the transductive setting shows a high recall (0.9551) without any significant loss of precision. In the inductive setting, most models prioritize recall over precision, but our model shows exceptionally outstanding performance overall.

Applying multi-modality to the baseline GNN models considerably improves their performances. It is noteworthy that the independent GNN layer for each feature modality contributes to improving recall as well as the other performance metrics. However, none of the baselines is able to reach the performance of our model due to the dedicated design for representing password reuse relations.

Impact of hidden relation policy. We compare different hidden relation policies (see Table 2). Overall, the hidden relation policy successfully enhances the prediction model. There is only a slight performance variation between policies in the transductive setting because the already well-connected website nodes allow for optimal information propagation. In the inductive setting, however, the hidden relation policy significantly improves prediction performance, demonstrating that it is particularly beneficial for linking arbitrary websites. Notably, the primary model with the shared user policy achieves the highest prediction performance, indicating that user information-based hidden relation edges are effective in predicting password reuse relations. Interestingly, applying the probabilistic policy also enhances models without requiring any user information by implicitly propagating information regarding password reuse. The similarity policy is an alternative with no privacy implications. Website feature similarity effectively assists the prediction task on the password reuse graph and achieves comparable performance in both learning settings. Moreover, the similarity policy is highly efficient because it creates a much smaller number of hidden relation edges compared to the probabilistic policy.

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 [10] and determine that it detects reused passwords with an accuracy of 0.6510. In the coordination method [12], 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, our prediction model (with up to 0.9559 F1-score) significantly outperforms these two detection methods.

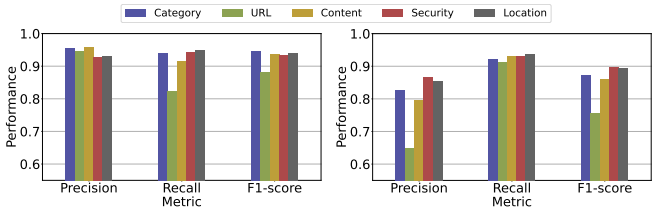
6.2.2. Ablation Study. To examine the effectiveness of each design strategy, we conduct an ablation study on our primary prediction model as shown in Table 3.

Feature-level ablation. We conduct a feature ablation study by individually eliminating each feature, which provides a comprehensive evaluation of feature influences and relationships. To obtain deeper insights, we also present the performance achieved by using each single feature in Figure 4.

The location feature is the most influential in predicting password reuse relations, followed by the security posture feature. This is consistent with previous studies in that users consider the service country and security when reusing passwords [15], [16], [21]. While modeling high-dimensional content feature alone poses challenges due to overfitting, the category feature itself is more effective (see Figure 4). However, when learning multi-modal features comprehensively, the content feature serves as an alternative to the category feature (see Table 3). Notably, our primary model

TABLE 3: Ablation study of PASSREFINDER. We highlight the worst (), the 2nd worst (), and the 3rd worst ().

	Ablation	Transductive setting			Inductive setting		
		Precision	Recall	F1-score	Precision	Recall	F1-score
Feature -level	**Category**	0.9547 **-0.20%**	0.9485 **-0.69%**	0.9516 **-0.45%**	0.8776 **-1.85%**	0.9290 **+0.27%**	0.9026 **-0.81%**
	URL	0.9586 **+0.21%**	0.9559 **+0.08%**	0.9572 **+0.14%**	0.8858 **-0.93%**	0.9137 **-1.38%**	0.8995 **-1.15%**
	Content	0.9512 **-0.56%**	0.9478 **-0.76%**	0.9495 **-0.67%**	0.8836 **-1.17%**	0.9130 **-1.46%**	0.8981 **-1.31%**
	Security posture	0.9500 **-0.69%**	0.9452 **-1.04%**	0.9476 **-0.87%**	0.8649 **-3.27%**	0.9338 **+0.79%**	0.8980 **-1.32%**
	Location	0.9498 **-0.71%**	0.9422 **-1.35%**	0.9460 **-1.04%**	0.8666 **-3.08%**	0.9247 **-0.19%**	0.8947 **-1.68%**
Method -level	**Multi-modality**	0.9523 **-0.45%**	0.9064 **-5.10%**	0.9288 **-2.84%**	0.8834 **-1.20%**	0.8933 **-3.58%**	0.8883 **-2.38%**
	Neighborhood attention	0.8505 **-11.1%**	0.8690 **-9.01%**	0.8597 **-10.1%**	0.8491 **-5.03%**	0.8303 **-10.4%**	0.8396 **-7.74%**
	Hidden relation	0.9459 **-1.12%**	0.9198 **-3.70%**	0.9327 **-2.43%**	0.7593 **-15.1%**	0.8857 **-4.40%**	0.8176 **-10.2%**
	***None* (PASSREFINDER)**	0.9566	0.9551	0.9559	0.8941	0.9265	0.9100



(a) Transductive setting (b) Inductive setting

Figure 4: Performance based on individual feature learning.

achieves better performance by leveraging both the category and content features due to their implicit relationships with other features. Lastly, the URL feature is less important than others because there are few domain-subdomain relationships in the dataset. Nevertheless, it exhibits significant performance in the transductive setting (see Figure 4a). Our further analysis reveals that the model eases the prediction task by exactly identifying websites observed in the transductive training process based on their URL features.

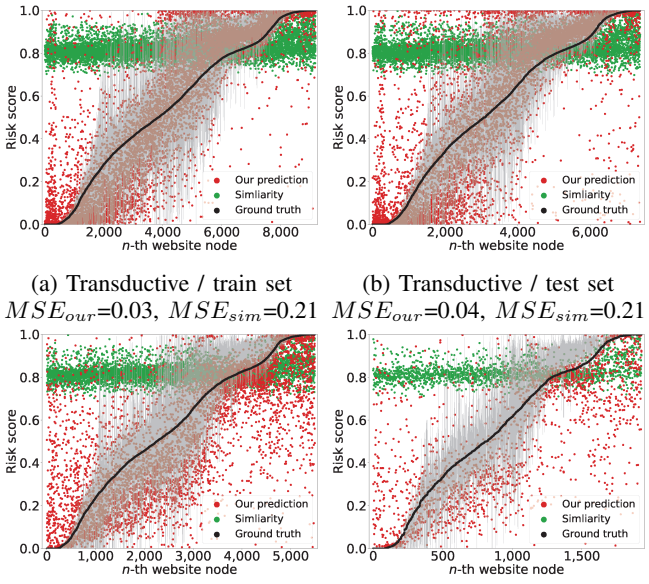
Method-level ablation. We also design other variants to study the contributions of multi-modality, neighborhood attention, and hidden relations. We concatenate all features for the ablation of multi-modality and utilize a mean pooling aggregation for the ablation of neighborhood attention. The results show that the method of multi-modality enhances the prediction performance to similar extents in both learning settings, which is also confirmed by the results of the GNN baseline models in Table 2. However, the contributions of the other methods are significantly different between the two learning settings. In the transductive setting, neighborhood attention can effectively prioritize neighbor nodes by taking advantage of observing all website nodes in the training process. The propagation of implicit influences assisted by hidden relation edges is no longer necessary. Therefore, the method of hidden relation has little impact on the performance in the transductive setting with a 2% gain in F1-score. On the other hand, hidden relations play a key role in the inductive setting (with a gain of 11%), where we have no information on password reuse relation edges for connecting newly observed website nodes.

6.2.3. 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 (6) 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 feature similarities in the similarity classifier, an intuitive method for quantifying password reuse tendencies.

TABLE 4: Evaluation using ranking metrics. We present the results with 64 sampled edges for each website node. For the similarity classifier, the difference between the learning settings lies solely in the split datasets.

Metric	Random	Similarity	PASSREFINDER
		Trans / Induct	Trans / Induct
Prec@1	0.0112	0.0249 / 0.0493	0.0875 / 0.0590
Prec@5	0.0785	0.1011 / 0.1085	0.2001 / 0.1566
Prec@10	0.1594	0.1762 / 0.1804	0.2952 / 0.2479
Prec@30	0.4699	0.4853 / 0.4797	0.6075 / 0.5442
Prec@50	0.7808	0.7847 / 0.7825	0.8692 / 0.8239
nDCG@1	0.5577	0.5534 / 0.6063	0.7498 / 0.6954
nDCG@5	0.5669	0.5603 / 0.6085	0.7682 / 0.7063
nDCG@10	0.5796	0.5706 / 0.6169	0.7790 / 0.7316
nDCG@30	0.6310	0.6185 / 0.6626	0.8113 / 0.7718
nDCG@50	0.6868	0.6729 / 0.7148	0.8377 / 0.8079

We first demonstrate that the prediction probability effectively prioritizes the risks of website pairs. We evaluate our prediction results using ranking metrics such as precision and normalized discounted cumulative gain (nDCG) at k edges of a given website node (see Table 4). Precision@k denotes the proportion of predicted positive edges in the top-k truly positive edges, and nDCG@k reflects the risk score values as well as their order. We establish a naïve baseline (*Random*) that randomly predicts the risk score between two websites based on a uniform distribution. Our model achieves significant gains of up to 30% in Precision@30 in the transductive setting with 64 sampled edges per each website node. It also noticeably outperforms the *Random* baseline and the similarity classifier in nDCG. When replicating the evaluation on 128 sampled edges, we



(a) Transductive / train set MSE_{our}=0.03, MSE_{sim}=0.21

(b) Transductive / test set MSE_{our}=0.04, MSE_{sim}=0.21

(c) Inductive / train set MSE_{our}=0.05, MSE_{sim}=0.20

(d) Inductive / test set MSE_{our}=0.06, MSE_{sim}=0.21

Figure 5: 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.

observe larger gaps between our model and the others. In a more practical environment with numerous websites, our prediction model will achieve much higher performance gains. Consequently, our prediction outcomes are useful for prioritizing pairs of websites by identifying higher-risk pairs.

In addition, we examine whether the prediction probability value can be directly utilized as a risk score. We present the visualization of our prediction probabilities, feature similarities and their errors compared to the true password reuse rates (i.e., the ground truth) in Figure 5. The other baseline results are described in Appendix A.2. Contrary to the similarity classifier, our prediction probabilities successfully fit to the ground truth risk scores. In the transductive setting, the risk scores give more precise estimates compared to the inductive setting, as shown by the lower error. This is because a transductively-learned model was able to observe a part of the password reuse relation edges connected from each website node, whereas an inductively-learned model never learned any edges involving newly observed website nodes. Notably, the usefulness of the prediction probability in the train set is generalized effectively to the test set in both learning settings. This insight shows that our prediction model is capable of measuring the quantitative risk of credential stuffing between websites in practice.

7. Application of PassREFinder

We illustrate the workflow of PassREFinder in Figure 6. The process begins with administrators 1) collecting websites under their management and target websites they

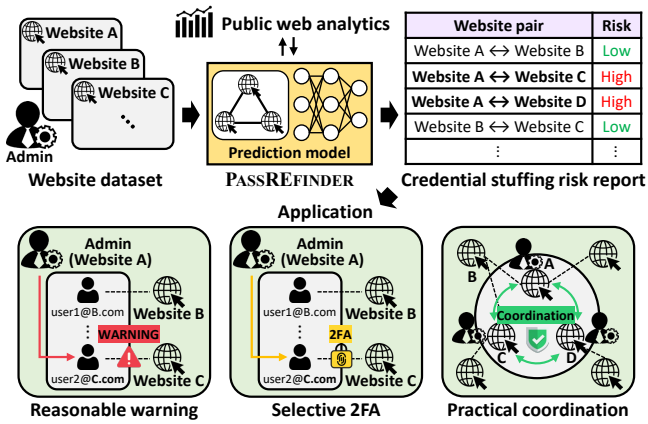


Figure 6: PassREFinder workflow and application.

seek to evaluate credential stuffing risks originating from the managed websites. Then, they 2) construct password reuse graphs and extract website features from public web analytic services. Utilizing the information gathered from the managed websites, administrators 3) establish ground truth labels and train a prediction model. Subsequently, administrators 4) predict credential stuffing risks between websites with a specific focus on the target websites. This process results in a credential stuffing risk report detailing the risk levels associated with each website pair (i.e., whether a credential stuffing risk is high or low). Based on these results, administrators can identify *risky websites* that pose potential threats of credential stuffing against the managed websites. It is noteworthy that this preemptive risk prediction approach can address the limited detection coverage of C3 services and poor password usability and scalability of the coordination methods. Consequently, administrators can proactively mitigate credential stuffing risks based on the insights on risky websites. We describe three practical application scenarios of PassREFinder:

i) The primary goal of PassREFinder is to directly warn users of high-risk relationships between administrators' websites and risky websites. Administrators can broadcast (password reset) warnings about risky websites to *all users* in the managed websites. We posit that administrators employ warning broadcasts appropriately rather than engaging abuse as they are dedicated to protecting their websites and users against attacks. Nevertheless, this inadvertent warning strategy relies on users' security awareness and can even lead to fatigue from constant warning exposure. Therefore, we suggest a selective warning approach targeting specific users. Administrators 1) infer users with accounts on risky websites by matching users' email domains with the risky websites' domains, circumventing the impracticality of identifying detailed user registration information, 2) broadcast warnings or enforce password reset for designated users, and 3) provide users with a convincing explanation on the aforementioned warnings, evidenced by specific website features, rather than simple password reuse notifications [8].

ii) Two-factor authentication (2FA) has been widely de-

ployed to defend against credential-related attacks. However, user-wide 2FA on websites is challenging due to the required levels of effort and time [62], [63]. Instead, selectively activating 2FA for specific users based on our prediction results can balance the user costs and credential stuffing mitigation. To this end, administrators can adopt an approach similar to the selective warning application scenario. They 1) infer users with accounts on risky websites by matching email and website domains, 2) strongly recommend or enforce these users to adopt 2FA, and 3) by elucidating specific website features and the associated risks, convince users to adhere to these precautionary measures. In addition, administrators can institute more fine-grained levels of 2FA enforcement based on the risk scores evaluated by our framework, allowing for judicious consideration of the trade-off between user convenience and confident safeguarding. This approach will be particularly beneficial for large or sensitive services where the enforcement of 2FA is already mandatory.

iii) Although website coordination sharing usernames and passwords is a promising approach to combat credential stuffing, existing implementations of website coordination [12], [13] have suffered from scalability and reluctance from administrators to participate. PASSREFINDER seeks to overcome these challenges by creating modestly sized yet effective coordination pools. To this end, administrators 1) 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 2) provide persuasive explanations, underpinned by website features, to encourage other administrators' active participation in the coordination process. Then, administrators 3) deploy coordination protocols for detecting password reuse or credential stuffing per each website set. While addressing the scalability problem, PASSREFINDER can significantly improve the detection accuracy of coordination methods, as indicated by our comparison results in Section 6.2.1.

8. Limitation and Discussion

Privacy-preserving implementation. Despite our framework's exceptional performance in credential stuffing risk prediction tasks, deploying a fully privacy-preserving hidden relation policy remains a formidable challenge during deployment. To employ the shared user policy, administrators must compute the number of shared users between websites. Although user-related information is highly effective, sharing it among untrustworthy websites raises privacy concerns. In practice, administrators can mitigate these concerns by adopting a simple private intersection technique while accommodating minor overhead, as described in Section 5.2.5. Importantly, we also propose more practical yet competitive alternative implementations of hidden relation policies that entirely circumvent privacy implications. To enhance the reliability of our framework even further, administrators could potentially incorporate privacy-preserving learning methods such as federated learning (FL) [64]–[66].

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 our ablation studies have demonstrated 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.

9. Ethical Considerations

Potential harm in credential leak dataset. We utilize a large collection of accounts, called `Cit0day`, to evaluate our prediction model. The dataset has been publicly available since November 2020 and has already been reported by various security services [67]–[69]. Therefore, we believe that its adverse influences to victim users are limited in practice. In this sense, previous studies have analyzed such publicly disclosed credential datasets to protect users from abusing breached credentials [3], [7], [70]–[72]. In addition, we performed several processes to protect the privacy of the victim users. The dataset contains user email addresses and passwords. To avoid the identification of specific users, we anonymized all email addresses using a one-way hash function (e.g., SHA-256). We stored the dataset in an isolated machine to prevent the risk of an additional breach from our database. These processes could ensure the privacy of the users and avoid any additional harm.

Misuse of PASSREFINDER. There is a possibility that attackers may exploit our framework to facilitate credential stuffing attacks. To minimize the potential misuse of PASSREFINDER, we have incorporated preventive measures when releasing a replication package of our framework. Specifically, we will provide a trained classifier, a model structure, and a sample website dataset. Since the provided model uses the shared user policy, we establish pre-defined edge information and graph structure as files, avoiding the disclosure of shared user information for constructing the graph from scratch. Furthermore, we will not disclose the features of the sample websites, but will instead provide embedding vectors, so as to circumvent the identification or fingerprinting of victim websites. Access to the release will be granted only upon request and exclusively for research purposes, and only to individuals with verifiable identities. We believe that the contributions of our framework on proactively mitigating credential stuffing will significantly outweigh any potential risks associated with its use.

10. Related Work

Compromised credential checking (C3) services. C3 services allow users to cross-reference usernames and passwords based on leaked credential datasets [10], [11], [73]. They also help users deal with credential stuffing attacks using breach-alerts for their passwords. Thomas et al. [11] presented Google Password Checkup (GPC), a protocol for querying usernames and passwords from credential data breaches. Li et al. [9] formally described C3 services and their security requirements. Pal et al. [14] extended a C3 service to target credential tweaking attacks [74]. 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 [14].

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 [12], [13]. As a pioneering work, Wang et al. [12] 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 et al. [13] 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 [75]–[77]. 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 [14]. 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 minimal (or no) privacy implications.

Defense of password guessing attacks. In addition to credential stuffing, password guessing has been another major branch of password attack research [11], [72], [78]. 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 [79], [80]. Beyond addressing guessable password patterns, researchers have explored the detection of abnormal authentication traffic using traffic fingerprinting techniques [75], [76]. 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.

11. Conclusion

We propose PASSREFINDER, a framework for the risk prediction of credential stuffing between websites. We formulate a password reuse relation — the relationship between websites where users highly likely reuse passwords — and represent the password reuse relation on a website graph. We design a GNN-based model to predict the existence of edges for password reuse relations. The model is applicable to a large number of arbitrary websites by extracting public features and learning the information of newly observed websites through complementary edges. The evaluation on a real-world credential dataset shows the effectiveness of our prediction model by achieving F1-scores of 0.9559 and 0.9100 in transductive and inductive graph learning settings. Our experimental results on both learning settings allow website administrators to flexibly adopt our framework to suit their needs, such as applying to new websites. In addition, we show that our model greatly outperforms existing prediction models, and demonstrate that significant gains come from our effective design strategies. By employing PASSREFINDER, we believe that website administrators can deter credential stuffing attacks by preemptively recognizing the risk of password reuse.

Acknowledgement

We would like to thank Seung Ho Na for his comments and support. This research was supported by the Engineering Research Center Program through the National Research Foundation of Korea (NRF) funded by the Korean Government MSIT (NRF-2018R1A5A1059921).

References

[1] Verizon, "Data Breach Investigations Report," https://www.verizon.com/business/resources/reports/dbir/, 2022, accessed: 2022-07-19.

[2] Shape Security, "2018 Credential Spill Report," https://www.shapesecurity.com/reports/2018-credential-spill-report, 2018, accessed: 2022-07-19.

[3] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, "The Tangled Web of Password Reuse," in *NDSS*, 2014.

[4] R. Wash, E. Rader, R. Berman, and Z. Wellmer, "Understanding Password Choices: How Frequently Entered Passwords are Re-used across Websites," in *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, 2016.

[5] S. Pearman, J. Thomas, P. E. Naeini, H. Habib, L. Bauer, N. Christin, L. F. Cranor, S. Egelman, and A. Forget, "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*, 2017.

[6] K. Thomas, F. Li, A. Zand, J. Barrett, J. Ranieri, L. Invernizzi, Y. Markov, O. Comanescu, V. Eranti, A. Moscicki *et al.*, "Data Breaches, Phishing, or Malware? Understanding the Risks of Stolen Credentials," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017.

[7] C. Wang, S. T. Jan, H. Hu, D. Bossart, and G. Wang, "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*, 2018.

[8] M. Golla, M. Wei, J. Hainline, L. Filipe, M. Dürmuth, E. Redmiles, and B. Ur, ""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*, 2018.

[9] L. Li, B. Pal, J. Ali, N. Sullivan, R. Chatterjee, and T. Ristenpart, "Protocols for checking compromised credentials," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[10] "Have i been pwned?" https://haveibeenpwned.com/, accessed: 2022-07-19.

[11] K. Thomas, J. Pullman, K. Yeo, A. Raghunathan, P. G. Kelley, L. Invernizzi, B. Benko, T. Pietraszek, S. Patel, D. Boneh *et al.*, "Protecting accounts from credential stuffing with password breach alerting," in *28th USENIX Security Symposium*, 2019.

[12] K. C. Wang and M. K. Reiter, "How to end password reuse on the web," in *26th Annual Network and Distributed System Security Symposium (NDSS 2019)*. Internet Society, 2019.

[13] ——, "Detecting Stuffing of a User's Credentials at Her Own Accounts," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020.

[14] B. Pal, M. Islam, M. S. Bohuk, N. Sullivan, L. Valenta, T. Whalen, C. Wood, T. Ristenpart, and R. Chatterjee, "Might I Get Pwned: A Second Generation Compromised Credential Checking Service," in *31st USENIX Security Symposium*, 2022.

[15] S. Gaw and E. W. Felten, "Password Management Strategies for Online Accounts," in *Proceedings of the second symposium on Usable privacy and security*, 2006.

[16] E. Stobert and R. Biddle, "The Password Life Cycle: User Behaviour in Managing Passwords," in *10th symposium on usable privacy and security (SOUPS 2014)*, 2014.

[17] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal Deep Learning," in *ICML*, 2011.

[18] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph Attention Networks," *arXiv preprint arXiv:1710.10903*, 2017.

[19] C. Hori, T. Hori, T.-Y. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi, "Attention-based Multimodal Fusion for Video Description," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4193–4202.

[20] IBM Security, "Cost of a Data Breach Report 2022," https://www.ibm.com/reports/data-breach, 2022, accessed: 2023-03-16.

[21] M. AlSabah, G. Oligeri, and R. Riley, "Your Culture is in Your Password: An Analysis of a Demographically-diverse Password Dataset," *Computers & security*, 2018.

[22] P. Mayer, J. Kirchner, and M. Volkamer, "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)*, 2017.

[23] S. Sahin and F. Li, "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*, 2021.

[24] T. N. Kipf and M. Welling, "Semi-supervised Classification with Graph Convolutional Networks," *arXiv preprint arXiv:1609.02907*, 2016.

[25] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," *Advances in neural information processing systems*, 2017.

[26] Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, and Q. He, "Pick and Choose: a GNN-based Imbalanced Learning Approach for Fraud Detection," in *Proceedings of the Web Conference 2021*, 2021.

[27] Z. Yang, W. Pei, M. Chen, and C. Yue, "WTAGRAPH: Web Tracking and Advertising Detection using Graph Neural Networks," in *IEEE Symposium on Security and Privacy*, 2022.

[28] J. Cui, K. Kim, S. H. Na, and S. Shin, "Meta-Path-based Fake News Detection Leveraging Multi-level Social Context Information," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 325–334.

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

[30] urlscan, "urlscan.io," https://urlscan.io/, accessed: 2022-11-02.

[31] McAfee, "Customer URL Ticketing System," https://sitelookup.mcafee.com/en/feedback/url, accessed: 2022-11-02.

[32] Shodan, "Shodan Developer," https://developer.shodan.io/, accessed: 2022-11-03.

[33] G. Chrome, "Chrome Root Program," https://www.chromium.org/Home/chromium-security/root-ca-policy, accessed: 2021-08-13.

[34] Apple, "Available Trusted Root Certificates for Apple Operating Systems," https://support.apple.com/en-us/HT209143, accessed: 2021-08-13.

[35] M. NSS, "CA/Included Certificates," https://wiki.mozilla.org/CA/Included_Certificates, accessed: 2021-08-13.

[36] X. He and T.-S. Chua, "Neural Factorization Machines for Sparse Predictive Analytics," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017, pp. 355–364.

[37] J. Tang and K. Wang, "Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding," in *Proceedings of the eleventh ACM international conference on web search and data mining*, 2018, pp. 565–573.

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

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

[40] S. D. Das, A. Basak, and S. Dutta, "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*. Springer, 2021, pp. 164–176.

[41] A. S. Bozkir, F. C. Dalgic, and M. Aydos, "GramBeddings: A New Neural Network for URL Based Identification of Phishing Web Pages Through N-gram Embeddings," *Computers & Security*, 2022.

[42] A. Khatua and W. Nejdl, "Unraveling Social Perceptions & Behaviors towards Migrants on Twitter," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 16, 2022.

[43] H. Face, "xlm-roberta-base," https://huggingface.co/xlm-roberta-base, 2019, accessed: 2022-11-03.

[44] P. Yang, G. Zhao, and P. Zeng, "Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning," *IEEE access*, vol. 7, pp. 15 196–15 209, 2019.

[45] "CVE," https://cve.mitre.org/, accessed: 2022-11-03.

[46] "Vulnerability Metrics," https://nvd.nist.gov/vuln-metrics/cvss, accessed: 2022-11-03.

[47] S. Singanamalla, E. H. B. Jang, R. Anderson, T. Kohno, and K. Heimerl, "Accept the Risk and Continue: Measuring the Long Tail of Government `https` Adoption," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 577–597.

[48] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing internal Covariate Shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[49] "Stochastic Training on Large Graphs," https://docs.dgl.ai/en/0.8.x/guide/minibatch.html, accessed: 2022-11-09.

[50] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai *et al.*, "Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks," *arXiv preprint arXiv:1909.01315*, 2019.

[51] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient Private Matching and Set Intersection," in *Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23*. Springer, 2004, pp. 1–19.

[52] Apple, "Password Monitoring," https://support.apple.com/guide/security/password-monitoring-sec78e79fc3b/web, 2021, accessed: 2023-03-17.

[53] "Cit0Day Collection," https://raidforums.com/Thread-Cit0Day-Collection-Leaked-Download, accessed: 2021-3-13.

[54] R. V. Hogg, E. A. Tanis, and D. L. Zimmerman, *Probability and Statistical Inference*. Macmillan New York, 1977, vol. 993.

[55] L. N. Smith and N. Topin, "Super-convergence: Very Fast Training of Neural Networks Using Large Learning Rates," in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006. SPIE, 2019, pp. 369–386.

[56] L. Bilge, Y. Han, and M. Dell'Amico, "Riskteller: Predicting the Risk of Cyber Incidents," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017.

[57] K. Soska and N. Christin, "Automatically Detecting Vulnerable Websites before They Turn Malicious," in *23rd USENIX Security Symposium (USENIX Security 14)*, 2014, pp. 625–640.

[58] K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias, and K. Li, "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)*. IEEE, 2016, pp. 49–54.

[59] H. He, Y. Ji, and H. H. Huang, "Illuminati: Towards Explaining Graph Neural Networks for Cybersecurity Analysis," in *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, 2022.

[60] Y. Zhang, X. Dong, L. Shang, D. Zhang, and D. Wang, "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*, 2020.

[61] T. Chen and C. Guestrin, "Xgboost: A Scalable Tree Boosting System," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[62] J. Colnago, S. Devlin, M. Oates, C. Swoopes, L. Bauer, L. Cranor, and N. Christin, ""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*, 2018.

[63] M. Golla, G. Ho, M. Lohmus, M. Pulluri, and E. M. Redmiles, "Driving {2FA} Adoption at Scale: Optimizing {Two-Factor} Authentication Notification Design Patterns," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 109–126.

[64] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for Improving Communication Efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[65] H. Hosseini, S. Yun, H. Park, C. Louizos, J. Soriaga, and M. Welling, "Federated Learning of User Authentication Models," *arXiv preprint arXiv:2007.04618*, 2020.

[66] S. H. Na, H. G. Hong, J. Kim, and S. Shin, "Closing the Loophole: Rethinking Reconstruction Attacks in Federated Learning from a Privacy Standpoint," in *Proceedings of the 38th Annual Computer Security Applications Conference*, 2022, pp. 332–345.

[67] T. Hunt, "Inside the Cit0Day Breach Collection," https://www.troyhunt.com/inside-the-cit0day-breach-collection/, 2020, accessed: 2022-11-28.

[68] C. Cimpanu, "23,600 hacked databases have leaked from a defunct 'data breach index' site," https://www.zdnet.com/article/23600-hacked-databases-have-leaked-from-a-defunct-data-breach-index-site/, 2020, accessed: 2022-11-28.

[69] D. Hétu, "57% of Cit0Day leaked credentials linked to popular free email service providers," https://flare.systems/learn/resources/57-of-cit0day-leaked-credentials-linked-to-popular-free-email-service-providers/, 2021, accessed: 2022-11-28.

[70] Y. Li, H. Wang, and K. Sun, "A Study of Personal Information in Human-chosen Passwords and Its Security Implications," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016.

[71] B. Ur, S. M. Segreti, L. Bauer, N. Christin, L. F. Cranor, S. Komanduri, D. Kurilova, M. L. Mazurek, W. Melicher, and R. Shay, "Measuring Real-world Accuracies and Biases in Modeling Password Guessability," in *24th USENIX Security Symposium*, 2015.

[72] D. Wang, Z. Zhang, P. Wang, J. Yan, and X. Huang, "Targeted Online Password Guessing: An Underestimated Threat," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016.

[73] K. Lauter, S. Kannepalli, K. Laine, and R. C. Moreno, "Password Monitor: Safeguarding passwords in Microsoft Edge," https://www.microsoft.com/en-us/research/blog/password-monitor-safeguarding-passwords-in-microsoft-edge/, 2021, accessed: 2022-11-25.

[74] B. Pal, T. Daniel, R. Chatterjee, and T. Ristenpart, "Beyond Credential Stuffing: Password Similarity Models using Neural Networks," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.

[75] C. Herley and S. E. Schechter, "Distinguishing Attacks from Legitimate Authentication Traffic at Scale." in *NDSS*, 2019.

[76] Y. Tian, C. Herley, and S. Schechter, "Stopguessing: Using Guessed Passwords to Thwart Online Password Guessing," *IEEE Security & Privacy*, vol. 18, no. 3, pp. 38–47, 2020.

[77] M. Seo, J. Kim, E. Marin, M. You, T. Park, S. Lee, S. Shin, and J. Kim, "Heimdallr: Fingerprinting SD-WAN Control-Plane Architecture via Encrypted Control Traffic," in *Proceedings of the 38th Annual Computer Security Applications Conference*, 2022.

[78] D. Pasquini, A. Gangwal, G. Ateniese, M. Bernaschi, and M. Conti, "Improving Password Guessing via Representation Learning," in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021.

[79] M. Dell'Amico and M. Filippone, "Monte Carlo Strength Evaluation: Fast and Reliable Password Checking," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015.

[80] D. L. Wheeler, "zxcvbn: Low-Budget Password Strength Estimation." in *USENIX security symposium*, 2016, pp. 157–173.

[81] Y. Piao, S. Lee, D. Lee, and S. Kim, "Sparse Structure Learning via Graph Neural Networks for Inductive Document Classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Appendix A.

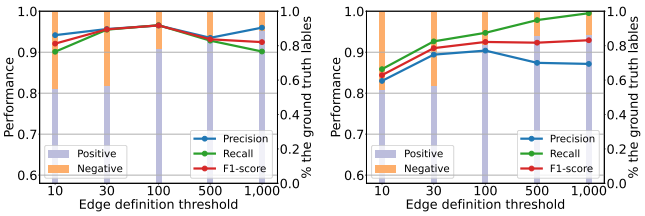
A.1. Sensitivity Analysis

To construct a password reuse graph and model the risk prediction task, we need to set certain thresholds and parameters that may influence prediction performance. Since there is no standard, we determine reasonable and high-performance values based on a sensitivity analysis.

Edge definition and ground truth. We establish the ground truth labels of password reuse relation edges only between websites that share at least a specific number of users, referred to as the edge definition threshold[2]. We examine the impact of various thresholds on prediction performance in Figure 7. A trade-off exists between the confidence of password reuse rates and the sparsity of password reuse relation edges. For a low threshold (e.g., 10), password reuse rates lack statistical significance due to limited user samples, resulting in performance degradation. In contrast, with a large threshold, password reuse relation edges are insufficient for effectively training the model. Furthermore, the model generates only positive predictions for all inputs, particularly in the inductive setting due to label imbalance. Thus, we choose an edge definition threshold of 30 shared users, which represents the minimum number for statistical significance and attains reasonable performance by robustly learning password reuse relations from balanced labels.

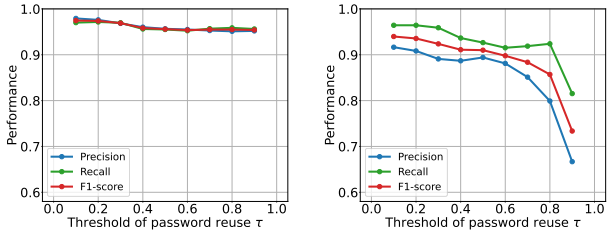
The ground truth labels themselves are also contingent on a threshold of password reuse rate τ (see Figure 8). Our prediction model demonstrates consistent performance in the transductive setting, wherein the model can (partially) learn the information of password reuse relations among all given websites. In the inductive setting, however, performance significantly declines with an increase in the threshold, as an inductively trained GNN model is highly sensitive to data imbalance [81]. Additionally, a trade-off exists between performance and granularity in predicting password reuse rates in the inductive setting. For example, while the model can achieve high performance with a large threshold, it generates many positive predictions, encompassing edges with low password reuse rates. This ground truth threshold should be carefully selected to generalize the model to new websites. In the main experiments, we set 0.5 for τ to balance the trade-off while maintaining competitive performance.

2. It is different to the parameter of the shared user hidden relation policy.



(a) Transductive setting (b) Inductive setting

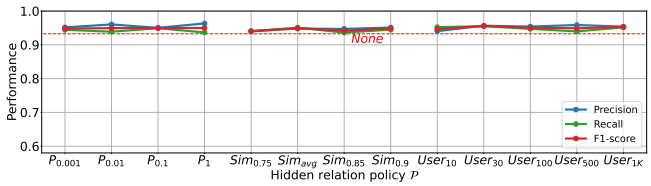
Figure 7: The performance of our prediction model along with different edge definition thresholds. We present the ratios of positive and negative labels in the train set.



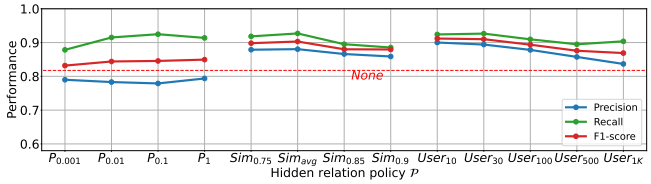
(a) Transductive setting (b) Inductive setting

Figure 8: The performance of our prediction model along with different thresholds of password reuse rate.

Hidden relation policy. We examine the impact of each hidden relation policy with different parameter values on prediction performance (see Figure 9). Overall, in the transductive setting, parameter values exert no significant influence due to the redundancy of hidden relation edges. When evaluating the probabilistic policy, performance slightly increases with the probability p. Although excessive hidden relation edges may incur overhead in processing the graph neural networks, we select $p = 1$, which yields the best performance, in the main experiments. The similarity hidden relation policy generally performs well, but its performance slightly declines as parameter ρ deviates from the average similarity ($\rho = avg$). While this may not be the optimal choice, we adopt the average for the similarity hidden relation policy in the main evaluation to provide intuitive explanations. Regarding the shared user hidden relation policy, we evaluate five values of parameter n, consistent with the sensitivity analysis of the edge definition threshold. It is important to note that the edge definition threshold is fixed at 30 for the consistency of the test set. Lower parameter values exhibit high prediction performance, particularly in the inductive setting, due to the denser connections of hidden relation edges. However, a limited number of shared users (e.g., $n = 10$) cannot effectively link related websites, which may even yield a negative influence in the transductive setting. Consequently, we choose $n = 30$ to maintain confidence in website relationships.



(a) Transductive setting

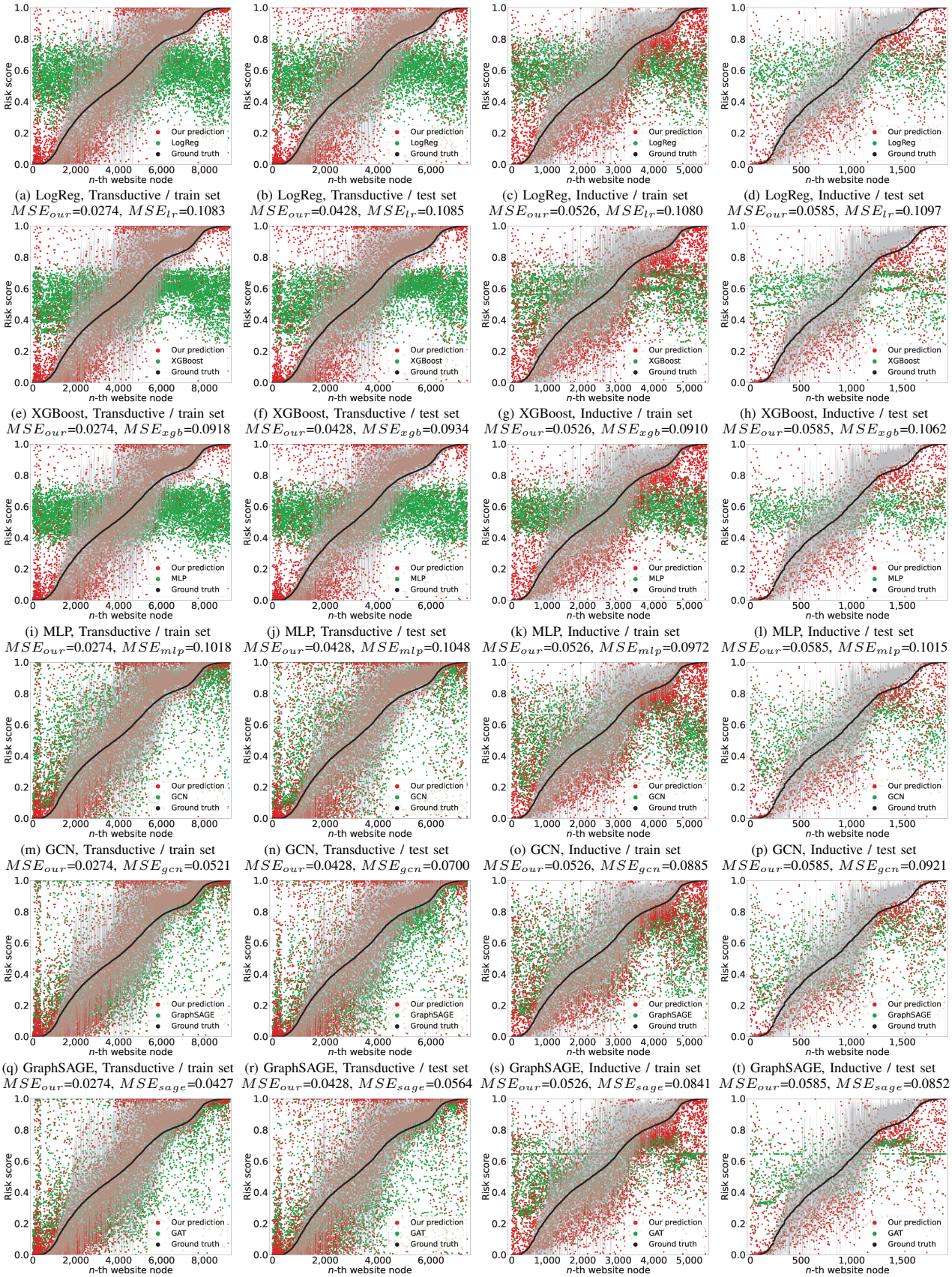


(b) Inductive setting

Figure 9: The performance with different parameter values of each hidden relation policy. *None* denotes performance with no hidden relations.

A.2. Risk Score Analysis: Overall Results

In Section 6.2.3, we present a risk score analysis on our prediction model and a comparison with the similarity classifier. Additionally, we evaluate risk scores measured by other baselines in Figure 10. The overall insights of the results align with those of the similarity classifier; all the baselines achieve lower errors in risk scoring in the transductive setting compared to the inductive setting, and the ability to fit the ground truth is effectively generalized to the test set. Nevertheless, all other baseline models exhibit lower errors than the similarity classifier as they optimize the probability of password reuse relations through advanced learning approaches. In accordance with the unsatisfactory prediction performance of the traditional machine learning models and MLP, their risk scores fail to precisely fit the exact password reuse rates due to the absence of neighborhood influences. On the other hand, the graph-based models produce very practical risk scores. However, in the inductive setting, predicted results substantially deviate from the ground truth when scores are either extremely low or high, due to the absence of hidden relation. We further analyze the prediction results of GAT, which produces a fixed score (~ 0.65) for some inputs. When aggregating neighborhood features, the graph attention layers prioritize website nodes with predominantly zero features due to sparse connectivity in the inductive setting. This eventually results in overfitting the risk scores between these websites to a fixed value.



(a) LogReg, Transductive / train set MSE_{our}=0.0274, MSE_{lr}=0.1083

(b) LogReg, Transductive / test set MSE_{our}=0.0428, MSE_{lr}=0.1085

(c) LogReg, Inductive / train set MSE_{our}=0.0526, MSE_{lr}=0.1080

(d) LogReg, Inductive / test set MSE_{our}=0.0585, MSE_{lr}=0.1097

(e) XGBoost, Transductive / train set MSE_{our}=0.0274, MSE_{xgb}=0.0918

(f) XGBoost, Transductive / test set MSE_{our}=0.0428, MSE_{xgb}=0.0934

(g) XGBoost, Inductive / train set MSE_{our}=0.0526, MSE_{xgb}=0.0910

(h) XGBoost, Inductive / test set MSE_{our}=0.0585, MSE_{xgb}=0.1062

(i) MLP, Transductive / train set MSE_{our}=0.0274, MSE_{mlp}=0.1018

(j) MLP, Transductive / test set MSE_{our}=0.0428, MSE_{mlp}=0.1048

(k) MLP, Inductive / train set MSE_{our}=0.0526, MSE_{mlp}=0.0972

(l) MLP, Inductive / test set MSE_{our}=0.0585, MSE_{mlp}=0.1015

(m) GCN, Transductive / train set MSE_{our}=0.0274, MSE_{gcn}=0.0521

(n) GCN, Transductive / test set MSE_{our}=0.0428, MSE_{gcn}=0.0700

(o) GCN, Inductive / train set MSE_{our}=0.0526, MSE_{gcn}=0.0885

(p) GCN, Inductive / test set MSE_{our}=0.0585, MSE_{gcn}=0.0921

(q) GraphSAGE, Transductive / train set MSE_{our}=0.0274, MSE_{sage}=0.0427

(r) GraphSAGE, Transductive / test set MSE_{our}=0.0428, MSE_{sage}=0.0564

(s) GraphSAGE, Inductive / train set MSE_{our}=0.0526, MSE_{sage}=0.0841

(t) GraphSAGE, Inductive / test set MSE_{our}=0.0585, MSE_{sage}=0.0852

(u) GAT, Transductive / train set MSE_{our}=0.0274, MSE_{gat}=0.0459

(v) GAT, Transductive / test set MSE_{our}=0.0428, MSE_{gat}=0.0531

(w) GAT, Inductive / train set MSE_{our}=0.0526, MSE_{gat}=0.0713

(x) GAT, Inductive / test set MSE_{our}=0.0585, MSE_{gat}=0.0724

Figure 10: Overall results of the risk score analysis following Figure 5.

Appendix B.
Meta-Review

B.1. Summary

This paper develops a graph neural network model (GNN) for how likely two websites are to have shared users that re-use passwords. The authors utilize two types of features: 1) historical password reuse relations, and 2) website features (i.e., IP, website category, content, URL, and security posture). The authors find that their GNN outperforms a large number of traditional ML and other graph networks. The envisioned use for the GNN is to predict the risk of credential reuse between websites and enable website administrators to leverage this information to strengthen their authentication policies to better protect users.

B.2. Scientific Contributions

- Creates a new tool to enable future science
- Provides a valuable step forward in an established field
- Addresses a long-known issue

B.3. Reasons for Acceptance

1) The paper provides a valuable step forward in an established field, i.e., strengthening password-based authentication by providing tools to identify vulnerable passwords. Different from prior work, the proposed method does not rely on sharing encrypted user data between websites and aims to be more proactive than compromised credential checking (C3) services.

2) In addition, the paper creates a new tool to enable future science: the proposed framework will be made available to enable researchers to further explore the adoption of the proposed method. The release will be limited to accredited researchers to mitigate the ethical concern of potential abuse by attackers.

B.4. Noteworthy Concerns

1) The paper discusses several promising practical applications for the proposed framework. However, these applications are not implemented or evaluated in the paper, and therefore left for future work. Such an evaluation should consider whether:

   a) Broadcasting warning messages for risky websites could be abused by administrators and could potentially harm the reputation of other websites.

   b) Users could suffer from warning fatigue or confusion if the messages are not sufficiently informative or actionable.

   c) Focusing on selective use of 2FA may not be the best strategy given the wide adoption of 2FA by large websites and the gradual shift toward passwordless authentication protocols.