Contents lists available at ScienceDirect

Online Social Networks and Media

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

Evaluating password strength based on information spread on social networks: A combined approach relying on data reconstruction and generative models

Maurizio Atzori [b], Eleonora Calò [a], Loredana Caruccio [a], Stefano Cirillo [a,*], Giuseppe Polese [a], Giandomenico Solimando [a]

[a] *Department of Computer Science, University of Salerno, Fisciano, Salerno, Italy*
[b] *Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy*

ARTICLE INFO

Keywords:
Privacy-preserving
Password-disclosure
Data wrapping
Data reconstruction
Social network

ABSTRACT

Ensuring the security of personal accounts has become a key concern due to the widespread password attack techniques. Although passwords are the primary defense against unauthorized access, the practice of reusing easy-to-remember passwords increases security risks for people. Traditional methods for evaluating password strength are often insufficient since they overlook the public personal information that users frequently share on social networks. In addition, while users tend to limit access to their data on single profiles, personal data is often unintentionally shared across multiple profiles, exposing users to password threats. In this paper, we present an extension of a data reconstruction tool, namely SODA ADVANCE, which incorporates a new module to evaluate password strength based on publicly available data across multiple social networks. It relies on a new metric to provide a comprehensive evaluation of password strength. Moreover, we investigate the capabilities and risks associated with emerging Large Language Models (LLMs) in evaluating and generating passwords, respectively. Specifically, by exploiting the proliferation of LLMs, it has been possible to interact with many LLMs through Automated Template Learning methodologies. Experimental evaluations, performed with 100 real users, demonstrate the effectiveness of LLMs in generating strong passwords with respect to data associated with users' profiles. Furthermore, LLMs have proved to be effective also in evaluation tasks, but the combined usage of LLMs and SODA ADVANCE guaranteed better classifications up to more than 10% in terms of F1-score.

1. Introduction

In today's digital landscape, where our personal information is constantly exposed to cyber threats, the security of our online accounts has become paramount. Passwords, as the primary line of defense against unauthorized access, play a crucial role in safeguarding our digital identities. However, the increasing spread of password-cracking techniques in combination with the problem of password reuse, has exposed individuals to heightened security risks. In this context, evaluating the effectiveness of user-generated passwords has emerged as a critical challenge.

Conventional password strength evaluation methods, such as complexity rules and dictionary checks, often fail to adequately assess the true security posture of passwords. These methods rely on static criteria, such as the syntax of the words adopted as passwords, that may not sufficiently capture the semantics and context of password usage patterns. Indeed, a user who generally chooses a password for their account tends to use keywords or phrases that are easy to remember, i.e., connected to a semantically close context, such as family members' names, favorite sports teams, or significant dates like birthdays and anniversaries. Much of this information is often shared on social networks and is accessible online on different user profiles, potentially exposing a user to password security issues. Although each social network allows users to define restrictions for access to their data, users who are registered on multiple social networks often unconsciously share information that they have privatized on one profile on another profile [1]. In this way, through data reconstruction tools or advanced crawlers, it is possible to reconstruct information semantically connected to a context close to the users [2].

In this scenario, the ever-increasing diffusion of Large Language Models could represent a useful tool for evaluating passwords based

[*] Corresponding author.
E-mail addresses: atzori@unica.it (M. Atzori), ecalo@unisa.it (E. Calò), lcaruccio@unisa.it (L. Caruccio), scirillo@unisa.it (S. Cirillo), gpolese@unisa.it (G. Polese), gsolimando@unisa.it (G. Solimando).

https://doi.org/10.1016/j.osnem.2024.100278
Received 7 March 2024; Received in revised form 25 May 2024; Accepted 27 May 2024
Available online 14 June 2024

on personal data, but also a significant threat in the password reconstruction process. LLMs, trained on massive amounts of text and code, possess the ability to learn and understand the semantics and the context of human language, including patterns and relationships in password formation. By leveraging public user data or data available online, as in the case of Google Gemini, LLM can provide a more comprehensive evaluation of password strength, considering not only complexity but also the context of password usage. On the other hand, LLMs can represent a useful tool for malicious users trying to infer passwords linked to personal profiles.

In this paper, we analyze the privacy issues associated with the sharing of sensitive personal data in social networks and explore the capabilities of Large Language Models (LLMs) in password evaluation and generation. To this end, we first propose a new extension of the data reconstruction tool SODA, namely SODA ADVANCE, which includes a new module for evaluating password strength based on information publicly available on social networks. This module exploits some of the most known approaches for evaluating password strength starting from personal information, i.e., CUPP, LEET, COVERAGE, and FORCE, and introduces a new cumulative metric, namely Cumulative Password Strength (CPS), which combines the results achieved by each approach to provide a more comprehensive evaluation of each password. Furthermore, we design several new strategies for interacting with LLMs, i.e., pipelines, with the aim of investigating the capabilities and threats associated with generating strong passwords and evaluating password strength using some of the emerging LLMs, such as Google Gemini, ChatGPT, Claude, Dolly, Falcon, and LLaMa.

In this paper, we try to answer the following research questions (RQs):

RQ1: Can we rely on LLMs to suggest complex and easy-to-remember passwords based on publicly available information on social networks?
RQ2: Can LLMs represent a valid tool to support users in evaluating the strength of passwords based on personal information?
RQ3: How does the public availability of personal information across multiple social networks impact the capabilities of LLMs to generate and evaluate password strength?
RQ4: How effective is the prompt-based methodology for password generation and evaluation compared to state-of-the-art models?

The main contributions of the proposed study are summarized as follows:

- A new extension of the data reconstruction tool SODA [1], namely SODA ADVANCE, which extends the capabilities of the previous version and introduces a new module for the evaluation of password strength;
- A new cumulative metric for evaluating password strength based on personal information, namely Cumulative Password Strength (CPS), which combines the results of different metrics in order to obtain a more detailed evaluation of the strength of a password;
- Three new interaction pipelines that enable us to perform an in-depth analysis of the capabilities of the most recent LLMs in the context of generating and evaluating passwords based on public personal information reconstructed from social networks;
- Different new prompting functions for generating and evaluating password strength generated with new automated and manual prompting engineering approaches;
- An extended experimental evaluation involving users and real data to evaluate password security threats when using public information reconstruction tools on social networks and LLMs.
- A comparative evaluation demonstrating the effectiveness of prompt-based strategies in generating and evaluating passwords with respect to state-of-the-art models.

The remainder of the paper is organized as follows: Section 2 discusses the most recent works concerning password disclosure and password evaluation metrics and frameworks; Section 3 presents an overview of the password strength problem and the importance of defining a strong password for users that use social networks in the era of LLMs; Section 4 presents the modules of the new data reconstruction tool capable of assessing passwords strength based on reconstructed data; Section 5 provides preliminary notions about LLMs involved in our study and the prompt engineering strategies for interacting with them; Section 6 introduces the process pipelines underlying our study to address the problem of generating strong password and evaluating their strength combining LLMs and the new proposed tool; Section 7 shows the experimental evaluation performed involving real users and the results achieved by LLMs for the different evaluation sessions; Section 8 discusses potential alternatives to strengthening weak user passwords; conclusions and future directions are provided in Section 9.

2. Related work

Many recent works have investigated the threats related to the public information available on the web, password disclosure, and privacy-preserving techniques. To this end, in this section, we first analyze the most recent literature addressing these challenges. Then, we delve into methodologies utilized for password evaluation and generation, including the utilization of specific tools and deep learning techniques.

Privacy-preserving. Privacy-preserving aims to protect the privacy of users in the context of data management and processing. Many recent works have investigated attacks based on data extracted from social networks, mainly proposing thorough investigations into effective privacy-preserving techniques for social network data publication [3–6]. Moreover, due to the widespread sharing of sensitive information, several frameworks have been proposed to increase user awareness with respect to the spread of their sensitive information [7,8]. In [1,2], the authors focus on collecting user data to uncover privacy threats from inappropriate data sharing on social media, by highlighting that when data is not adequately protected with advanced privacy settings, across multiple social networking platforms, users become susceptible to significant risks to their privacy and security, such as the risk of inadvertent password exposure. In [9], the authors examine users' passwords in relation to the information shared on social network platforms, revealing that approximately 48% of users disclose their passwords on social network platforms. Other vulnerabilities related to the spread of publicly available user information have been discussed in [10,11]. In particular, the authors considered different attacks, such as brute force and dictionary attacks, to highlight the potential threats in the exploitation of users' information. For instance, privacy disclosure represents one of the most important threats to be managed, which led to the definition of new models and tools for user privacy protection. As an example, in [12], the authors have performed extensive experimental evaluations using both synthetic and real users' profiles with a new model exploiting public information related to users' subnetworks, such as number of friends, age of user accounts, and friendship duration, in order to provide a naive user with a good means for privacy protection. Moreover, the recent diffusion of Large Language Models (LLMs) has inspired several studies focusing on privacy disclosure and protection. Specifically, they leverage the capabilities of LLMs to automate the process of evaluating privacy practices across different platforms, enabling researchers to efficiently evaluate the level of privacy protection offered by various platforms [13,14].

Password evaluation metrics and frameworks. The exponential growth of cyber-attacks has prompted many platforms to adopt more stringent password composition policies to generate strong passwords. Password measures typically assess strength based on length and the presence of numbers and special characters [15]. However, despite efforts to educate users on the use of complex passwords, the definition of strong passwords continues to be a challenging task. In [16], the authors warn of the vulnerability of easy-to-remember passwords, emphasizing the risk of using guessable words or phrases. Other approaches, such as [17], evaluate the strength of passwords using leaked sensitive data and introduce a new metric. The latter is based on the coverage of sensitive information in the password and is weighted with a specific weight for each information identified. Specifically, [18] highlights the risk associated with the use of personal information in passwords and presents Personal-PCFG, a semantic method that leverages personal data to generate highly personalized passwords, demonstrating superior effectiveness compared to the original PCFG method. Other approaches, such as the one described in [19], use tools to perform dictionary attacks and check whether the password entered by the user can be associated with passwords easily generated by the tools themselves. In [20], the authors propose a data-driven framework that uses neural networks to evaluate password strength and suggest improvements. Instead, in [21], the authors investigate the application of deep learning techniques, demonstrating greater performance in evaluating the strength of passwords compared to traditional methods.

Although many studies propose suitable methods for evaluating and generating passwords, there are few studies in the literature that investigate the effectiveness of generative large language models in this specific context. To the best of our knowledge, this represents one of the first studies that aims to combine the capabilities of a data reconstruction tool with the capabilities of generative large language models to evaluate password strength based on personal data publicly available on social networks.

AI-based methodologies for generating passwords. To access an account, users must create a password that is strong enough to protect the personal information stored. Passwords can be generated using a variety of techniques, the most common of which is the use of random generators. However, artificial intelligence techniques can also be used to generate strong passwords. Several studies have proposed new strategies that combine artificial intelligence and NLP techniques for achieving these goals [22,23]. In [24], the authors propose a new tool for cracking passwords that relies on recent machine-learning techniques. It generates highly customized password dictionaries based on meaningful string segments extracted from a given password, leading to obtaining highly effective passwords with respect to the ones generated by traditional tools based on rules and alphanumeric patterns. Moreover, in [25], PassGAN is introduced as a novel approach that replaces human-generated password rules with theory-based machine-learning algorithms. Using a Generative Adversarial Network (GAN), it autonomously learns the distribution of real passwords from leaked data, outperforming traditional tools.

On the other hand, in [26], the potential of using fully unsupervised representation learning in the domain of password guessing has been demonstrated. Specifically, the authors introduce a probabilistic, unsupervised model called Completely Probabilistic Generation (CPG) for generating passwords. The model is used by both adversaries, to improve side-channel attacks and password-like attacks, and legitimate users, who may be interested in recovering their forgotten password while remembering a partial template. They also present the Dynamic Password Guessing (DPG) approach, which dynamically adapts the password-guessing strategy based on feedback received during an attack, significantly improving the impact of the attack itself.

A recent work presents two tools for generating secure passwords based on GPT-2, i.e., PassGPT and PassVQT [14]. The former is an implementation of the GPT-2 architecture that generates passwords sequentially, by sampling one character at a time, rather than generating the entire password at once. This approach allows for a more controlled and guided generation of passwords, enabling the satisfaction of specific constraints. It is trained on password leaks, i.e., datasets of compromised passwords from various sources, to learn patterns, structures, and characteristics of passwords commonly used by individuals. PassGPT outperforms existing methods based on generative adversarial networks (GANs) by guessing twice as many previously unseen passwords. Instead, PassVQT introduces vector quantization techniques that lead to a more diverse and complex password-generation process, which improves the generation capabilities of PassGPT.

It is important to notice that most of the previous models are used to randomly generate passwords based on datasets of leaked passwords. These datasets are used to train models capable of generating passwords with similar complexity and structure. Consequently, the generated passwords may or may not be related to the user requesting them. This aspect is fundamental, especially for non-expert users who aim to define complex but easy-to-remember passwords often related to their personal lives. On the other hand, this poses a threat to password security as sometimes information about users can be reconstructed from social profiles. In fact, such information can be used by attackers to perform more targeted cracking attacks, leading to the need to consider the semantic perspective during password strength evaluation processes. To the best of our knowledge, all the above-mentioned tools neglect this aspect and only exploit statistics on syntactic patterns extracted from passwords leaked onto the Web. For this reason, it is crucial to raise awareness among social network users to increase online security, not only by using strong and unique passwords but also by improving policies to evaluate password strength considering the semantic perspective. In this work, we use LLMs for both the generation and the evaluation of passwords by exploiting all information collected about a user. The goal is to understand whether although LLMs are not specifically trained for this purpose, they are capable of generating structurally complex but also easy to remember passwords.

3. Password strength in the era of LLMs and social networks

Passwords are the first defense in safeguarding access to digital services, such as e-commerce, online banking, and public administration platforms, which are common in the daily lives of users. The usage of weak passwords represents a relevant threat since they can lead to unauthorized access to sensitive data, identity theft, compromising privacy, and potentially accessing private information. Despite the growing awareness of the importance of establishing passwords with a sufficient combination of complexity, length, and originality, many users continue to adopt words related to their lives, such as family members, names of pets, or their favorite football team.

Recently, the National Institute of Standards and Technology (NIST) has published standard guidelines for suggesting users and companies how to improve password strength against unwanted attacks and intrusions [27]. In particular, NIST provides general syntactic guidelines, such as maintaining a length of the passwords of at least 8 characters and introducing a combination of uppercase letters, lowercase letters, numeric digits, and special symbols. Other guidelines suggest to companies and organizations to not frequently require users to update their passwords, since people tend to reuse old passwords by only adding a few changes to them, leading to a risk in their accounts. In fact, if a previous password has been compromised, even slight modifications or additions to that password could still be vulnerable to attack in the future. In this regard, NIST also suggests that organizations should consider these aspects and implement access control systems that keep track of these leaked passwords to prevent the use of vulnerable passwords within their networks.

Following NIST guidelines typically leads companies to adopt more secure guidelines and users to establish strong passwords. However, most of the guidelines proposed by NIST are mainly focused on syntactic perspective, leaving out an equally important aspect, which is the semantic perspective. This is a crucial consideration in crafting robust and user-friendly password policies. In fact, the semantic perspective in the process of defining passwords requires incorporating meaningful elements that hold personal significance for users, making it easier for them to recall complex passwords. This aspect may result in users turning to easily guessable or common patterns, potentially compromising the overall security of the authentication process [17,28]. This aspect becomes more relevant when talking about users who use social network platforms, where personal information is often shared. In fact, it is often possible to gather information about users by cross-referencing the information available on multiple social networks, also bypassing privacy requirements set on a single social network [2]. Recently, different data reconstruction tools have been proposed that aim to provide users insights into how their data are unconsciously spread on different social networks [1]. Starting from this information, it is possible to create a semantic context around a user that can be used by a malicious user to infer passwords adopted on different platforms.

In this scenario, the recent spread of generative Large Language Models (LLMs) poses new challenges and risks, since these models have the potential to explore and understand the semantic context around users in order to generate text (or password) related to them. Although most LLMs do not have direct access to users' data, this can be specified in the prompts provided to them so that they can be analyzed and profiled. The combination of LLMs with data reconstruction tools amplifies the risk of inferring the passwords adopted by users, due to the suggestions that these LLMs can provide to potential malicious users. Therefore, in this study, we investigate how the combination of LLM and data reconstruction tools can lead to potential risks in the stages of defining user passwords.

In the following sections, we first introduce a new data reconstruction tool defined as an extension of the tool proposed in [1]. Then, we investigate the effectiveness of the combination of some of the most recent LLMs with the new data reconstruction tool in both generating and evaluating passwords based on data shared on social networks by means of three different interaction pipelines.

4. Overview of the new data reconstruction tool

As discussed in the previous section, users often overlook privacy and security when sharing information on different social networks, unknowingly exposing themselves to potential threats [2]. In this scenario, data reconstruction tools can help to identify shared information that might pose privacy risks and facilitate the process of securing sensitive data across multiple platforms [29,30].

Recently a new data reconstruction tool, namely SODA, has been proposed [1], aiming to provide users insights into how their data can be reconstructed from different social networks. SODA has been built on the top of Social Mapper,[1] which is a tool able to search users on social network platforms by using only a photo and information about the user, such as first name, surname, city, email, or occupation. SODA exploits the search engines of various social networks to perform an initial search for people based on the information provided as input. Starting from this, it compares the publicly available photos on the profiles and tries to find a match with the input photo using different facial recognition algorithms.

In this paper, we present an extension to the data reconstruction tool SODA, namely SODA ADVANCE, which aims to provide password evaluation based on public data extracted from social networks. SODA ADVANCE includes a new module for evaluating the strength of passwords based on the information reconstructed about users from social networks. This module integrates four different well-known methods proposed in recent literature and a new cumulative metric designed to provide a general evaluation of the strength of a password. Moreover, different from SODA, the *Web Crawler* and *Web Scraper* of SODA ADVANCE have been equipped with new functionalities for extracting a large set of public information from Facebook (Fb), LinkedIn (In), and Instagram (Ig).

Table 1
Information extracted by the data reconstruction tool SODA ADVANCE.

Features name	Fb	In	Ig	Features name	Fb	In	Ig
Name	√	√	√	Surname	√	√	√
Address	√	√	✗	Hometown	√	✗	✗
Biography	√	√	√	Languages	√	√	✗
Birthday	√	√	✗	Phone	√	√	✗
Current city	√	√	✗	Politic orientation	√	✗	✗
Curriculum Vitae	✗	√	✗	Relationship	√	✗	✗
Email	√	√	✗	Religious orientation	√	✗	✗
Employment	√	√	✗	Sexual orientation	√	✗	✗
Work experience	✗	√	✗	Skills	✗	√	✗
Family members	√	✗	✗	Education	√	√	✗
Gender	√	✗	✗	Web Site	√	√	√

Table 1 provides an overview of the information collected by SODA ADVANCE from the considered social networks. As we can see, much information can be reconstructed from all three social networks, such as *Name*, *Web Site*, *Biography*, and *Surnames*, other from at most two, such as *Address*, *Email*, and *Birthday*, whereas most of them from only a social network, such as *Curriculum Vitae*, *Politic orientation*, and *Relationship*.

Fig. 1 shows an overview of the components of SODA ADVANCE, which integrates the extension of some modules of SODA and introduces the new password *Evaluation modules*. As we can see, SODA ADVANCE starts considering some information about a user, such as his/her first name, surname, and photo, which have been provided as input to the *Web Crawler module* (Step ①). The latter is responsible for starting the search process of the user on the different platforms by using the search engines of the social networks to reduce the set of users to be analyzed. After that, the *Scraper module* extracts information about the social profiles of users identified by the crawler, including their public profile photos (Step ②). Starting from this, the photos extracted are compared with the image given as input to SODA ADVANCE using the *Face Recognition module* (Step ③). This module uses Face Distance [31], a function that calculates the Euclidean distance between the extracted photo and the input image, which is essential to minimize cases of homonymy. Then, the *Merging module* combines the information extracted from different social networks and removes duplicates in order to obtain a clear representation of the reconstructed data (Step ④). After that, the *Evaluation module* computes different metrics to evaluate the strength of inputted passwords based on the reconstructed data (Step ⑤). More specifically, the *Evaluation module* reads as input the data reconstructed from social networks and one or more passwords and computes four values for each of these, i.e., CUPP, LEET, COVERAGE, and FORCE, to evaluate the strength of passwords. These values are then combined into a single value by an aggregation module. The resulting value represents the cumulative percentage of the password strength in relation to the information reconstructed from the social networks.

In what follows, we provide a detailed explanation of each method used to evaluate password strength and the new cumulative metric.

4.1. Password strength evaluation module

Evaluating the strength of passwords defined by users based on some of their information is a challenging task due to the necessity to consider both syntactic and semantic aspects of the personal information specified in passwords. To this end, it is necessary to adopt different specific metrics and methodologies that account for the incorporation of personal details in order to evaluate the predictability and vulnerability of these passwords from different perspectives. The *Evaluation module* of SODA ADVANCE exploits a new cumulative password strength metric that relies on the results achieved by four of the best-known metrics and methodologies defined in the literature, i.e., CUPP [32], LEET [33], COVERAGE [18], and FORCE [17].

[1] www.github.com/Greenwolf/social_mapper.

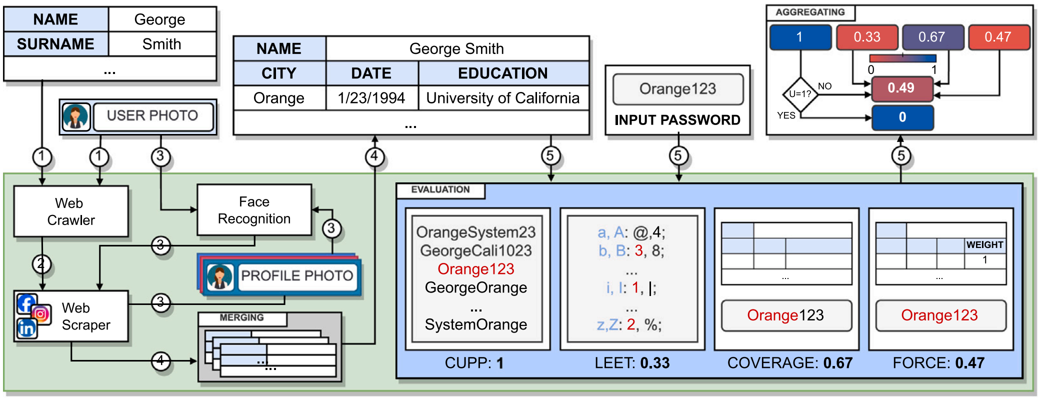


Fig. 1. Overview of the modules underlying SODA ADVANCE.

CUPP *method.* The Common User Password Profile (CUPP) is a password profiling tool capable of automating the process of generating lists of passwords considering a set of personal information of a user. The methodology behind CUPP starts by considering a set of information, such as names, birth dates, and cities of origin, combines and manipulates them to define a dictionary of customized words, which could represent common passwords. In fact, this methodology enables us to predict and identify potential passwords that users may choose based on their personal information. For example, let us consider the set of personal information, such as *Name*, *Surname*, *Birthday*, and *City*, e.g., *George, Smith, 1/23/1994, Orange*, some of the passwords generated by CUPP are *GeorgeOrange*, *Orange123*, *George!#?*, etc.

In SODA ADVANCE, this method is used to check whether a password specified by a user can be generated by CUPP, or in other words, whether it can be subject to a dictionary attack based on user-profiled data. In our approach, we associate a value U to each password based on whether the password can be generated by CUPP using data collected for a user.

Definition 4.1. Let $D = \{p_1, p_2, \ldots, p_n\}$ be the dictionary of passwords p_i created by manipulating sensitive user information, and let ρ be a password specified by a user to be evaluated with CUPP, we can associate a value U to ρ as follows:

$$U = \begin{cases} 1 & \text{if } \rho \in D \\ 0 & otherwise \end{cases} \tag{1}$$

More specifically, if a password can be found within the dictionary generated by CUPP, the associated value is set to 1; otherwise, it is set to 0. This approach is motivated by the fact that CUPP leverages personal information to create custom wordlists, which significantly increases the possibility of predicting passwords that users may choose. Specifically, by associating a binary value to each password, this approach allows SODA ADVANCE to precisely identify passwords vulnerable to dictionary attacks based on CUPP.

LEET *method.* Leetspeak (LEET) is a modified spelling system commonly used in online communication and gaming communities, characterized by replacing letters with visually similar characters or symbols to create a form of encryption or disguise. It represents an essential tool that allows us to identify passwords that incorporate Leetspeak variations, thus enhancing the accuracy of password strength evaluations. In fact, many users tend to replace letters with visually similar characters or symbols to increase the complexity of passwords and make them more resistant to dictionary-based attacks.

In SODA ADVANCE, we define a value L associated with each password representing the number of LEET characters within the password. These are the possible characters that can be replaced in a text with their visually similar counterparts, e.g., "*e*" replaced with "*3*" or "*a*" replaced with "@".

Definition 4.2. Let c be the number of possible characters that can be replaced in a password ρ, and let l be the length of ρ, we can compute the value L as follows:

$$L = \frac{c}{l} \tag{2}$$

The value L allows SODA ADVANCE to quantify the degree of Leetspeak usage in passwords, allowing for a more precise evaluation of their strength based on the password syntaxes. This value is between 0 and 1 and if the result tends to 1, the password is considered potentially weak, since it indicates a high reliance on Leetspeak substitutions.

COVERAGE *method.* The COVERAGE method evaluates the correlation between passwords and the personal information of users. In particular, it checks whether a password contains at least one of the personal information related to a user. The value of COVERAGE ranges from 0 to 1, where a value equal to 0 means that there is no personal information in a password, while 1 means that the entire password perfectly matches at least one with the considered personal information.

Starting from the value defined in [18], in SODA ADVANCE, we consider as C the number of characters in a password that exactly matches the personal information reconstructed from the Web. More specifically, this method calculates and verifies if there is any plain-text information in the password. For example, let us consider the password *George94* and the extracted information from the social networks of a generic user, such as *Name* and *Birthday* for *George Smith* born in *1994*. The COVERAGE method finds a perfect match between the name, i.e., *George*, but not with the reconstructed birthday, i.e., *1994*. Thus, the value of C in this case is equal to 0.75 since only the *Name* is contained in the password.

Definition 4.3. Let s_i be the length of ith information of the user that has a match within the password ρ, let l be the length of ρ, and let N be the number of information contained in ρ, thus C is calculated as follows:

$$C = \sum_{i=0}^{N} \frac{s_i}{l} \tag{3}$$

The value C determines the presence of extracted information in the password. It ranges from 0 to 1, with a higher value indicating a less secure password.

FORCE *method.* The FORCE method starts by considering the value of COVERAGE C, assigning a weight to each sensitive personal information reconstructed from the Web. These weights are assigned based on the frequency with which they can be found on social profiles with a value ranging from 0 to 1 [17]. Table 2 shows the weights associated with the features considered in SODA ADVANCE and reconstructed from the social network platforms. In SODA ADVANCE, we consider the value F, which is inversely proportional to the weighted amount of sensitive personal

Table 2
Weights associated with features extracted from SODA ADVANCE.

Features name	Weight	Features name	Weight
Name	1	Surname	1
Address	0	Hometown	0.78
Biography	0.022	Languages	0.05
Birthday	0.024	Phone	0
Current city	0.78	Politic orientation	0.001
Curriculum Vitae	0.1	Relationship	0.25
Email	0.3	Religious orientation	0.001
Employment	0.78	Sexual orientation	0.002
Work experience	0.12	Skills	0.04
Family members	0.05	Education	0.12
Gender	0.24	Web Site	0.09

information in a password. A password is considered weak if the value of F is less than or equal to 0.6. This value was obtained from the empirical evaluation presented in [17].

As an example, let us consider the previous scenario presented for the COVERAGE value, where the password is *George94* and the extracted information is *George* and *1994*. Thus, the weights k_i associated with the *Name* and the *Birthday* information of the user, are $k_1 = 1$ and $k_2 = 0.024$, respectively (see Table 2). The value F is equal to 0.24 meaning that the password *George94* is considered weak.

Definition 4.4. Let $Z = \{z_1, z_2, \dots, z_m\}$ a set of users analyzed by SODA ADVANCE, $G = \{i_1, i_2, \dots, i_n\}$ be the information that can be extracted for each user, and let e_j be the total number of information extracted by SODA ADVANCE for each information i_j, the weight k_j associated with i_j is defined as follows:

$$k_j = \frac{e_j}{m} \tag{4}$$

Starting from this, it is possible to compute the value F as formally defined in the following.

Definition 4.5. Let p_j be the length of the jth information in a password ρ, and l be the length of ρ, and let k_j the weight associated i_j, we can compute the value of F as follows:

$$F = 1 - \sum_{j=0}^{N} \frac{p_j}{l} k_j \tag{5}$$

The F value in SODA ADVANCE measures the strength of the password with respect to the information extracted from social networks and it ranges from 0 to 1. Consequently, the evaluation depends on the frequency of publication, which determines the approximate weight of how easy it is to find specific information online.

*Cumulative Password Strength (*CPS*).* Each of the methods introduced above provides a value that allows SODA ADVANCE to quantify the strength of the password based on different aspects. In particular, LEET computes the number of individual characters that could be replaced, while COVERAGE and FORCE evaluate the number of characters that correspond to extracted personal information. Instead, CUPP checks that all the characters in the password can form a string that can be easily generated. Starting from these methods, we have defined a new metric, namely Cumulative Password Strength (CPS) that combines the results of the previously described methods to provide a cumulative value $\mathcal{S}$ according to the results achieved by each of them.

Definition 4.6. Let ρ be a password specified by a user and U, L, C, and F be the values computed on ρ, by means of CUPP, LEET, COVERAGE, and FORCE methods, respectively. We can compute $\mathcal{S}$ as follows:

$$\mathcal{S} = 1 - \begin{cases} 1 & U = 1, \\ \frac{C+L+(1-F)}{3} & otherwise. \end{cases} \tag{6}$$

Table 3
CUPP, LEET, COVERAGE, FORCE, and $\mathcal{S}$ values for ρ_1, ρ_2, and ρ_3.

	ρ_1	ρ_2	ρ_3
U	1	0	0
L	–	0.6	0.4
C	–	0	0.5
F	–	0.5	0.5
$\mathcal{S}$	**0**	**0.64**	**0.54**

The result provides a value between 0 and 1, where 0 indicates a weak password and 1 indicates a strong password. It is important to notice that, if the value achieved by the CUPP method returns 1 means that the password ρ had an exact match with one of the passwords in the dictionary D generated by CUPP. If this is true, the password is considered weak by SODA ADVANCE.

Example 1. Let us consider a generic user namely *George Smith*[2] and suppose that after the analysis of the public information available on social network profiles, SODA ADVANCE reconstructs the data shown in Fig. 2.

If we consider the following passwords ρ_1 =“*Orange123*”, ρ_2 = “*$m1th90001*”, and ρ_3 =“*Smith90001*” the values achieved by methods in the evaluation module of SODA ADVANCE are shown in Table 3. As we can see, for the password ρ_1 the metric $\mathcal{S}_{\rho_1}$ is equal to 0. This is due to the fact that the value U_{ρ_1} is equal to 1, since the password ρ_1 matches with one of the passwords generated by CUPP, meaning that the password is extremely weak. In fact, the structure of ρ_1 consists of the city of the user and of part of the birthday, i.e., the day and month of birth. This information can be reconstructed through social networks, making the use of such a password highly vulnerable on any platform where it is used.

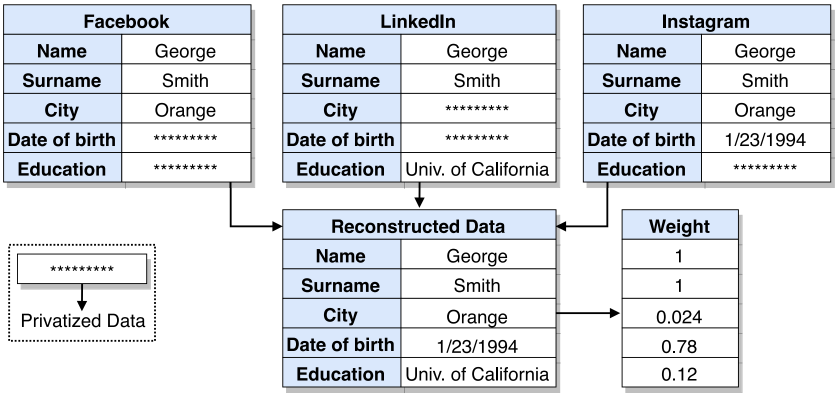
Concerning the passwords ρ_2 and ρ_3, we can see that they are written with different syntaxes, but contain the same information, i.e., the surname of the user. In fact, the password ρ_2 is composed of a series of substitutions of letters of the surname followed by a sequence of apparently meaningless numbers, since they do not seem to be related to the reconstructed data (Fig. 2). However, these numbers represent information that is semantically related to the user, but it has not been reconstructed on the social networks, i.e., the postal code of *Orange* city, which is not considered in the computation of the metrics.

$$\begin{cases} C_{\rho_2} = \frac{0}{10} = 0 & i = 0, \\ L_{\rho_2} = \frac{6}{10} = 0.6 & c = 6, \\ F_{\rho_2} = 1 - \frac{5}{10}1 = 0.5 & j = 1, p_1 = 5, k_1 = 1 \end{cases} \implies \mathcal{S}_{\rho_2} = 1 - \frac{0+0.6+(1-0.5)}{3} = 0.64$$

$$\begin{cases} C_{\rho_3} = \frac{5}{10} = 0.5 & i = 1, s_1 = 5 \\ L_{\rho_3} = \frac{4}{10} = 0.4 & c = 4 \\ F_{\rho_3} = 1 - \frac{5}{10}1 = 0.5 & j = 1, p_1 = 5, k_1 = 1 \end{cases} \implies \mathcal{S}_{\rho_3} = 1 - \frac{0.5+0.4+(1-0.5)}{3} = 0.54$$

As we can see, concerning the COVERAGE value C for ρ_2 is equal to 0 since ρ_2 does not contain a textual match between the password and the extracted data. Instead, in ρ_3 a part of a password matches with the surname of the user *Smith*, which has a length $s_i = 5$, leading it corresponds to half of the length of ρ_3. Therefore, for ρ_3 the COVERAGE value C_{ρ_3} is equal to 0.5. Concerning the LEET values they are L_{ρ_2} = 0.6 and L_{ρ_3} = 0.4 for ρ_2 and ρ_3, respectively. This is achieved by considering the number of characters that can be replaced with others, such as *$,1,0* with *s*, *i*, and *o*, which are $c = 6$ and $c = 4$ in ρ_2 and ρ_3, respectively. Concerning the FORCE values F, we have one match with

[2] The name and the data used in the example were chosen purely randomly and have no reference to a real person.



Facebook	
Name	George
Surname	Smith
City	Orange
Date of birth	*********
Education	*********

LinkedIn	
Name	George
Surname	Smith
City	*********
Date of birth	*********
Education	Univ. of California

Instagram	
Name	George
Surname	Smith
City	Orange
Date of birth	1/23/1994
Education	*********

Reconstructed Data		Weight
Name	George	1
Surname	Smith	1
City	Orange	0.024
Date of birth	1/23/1994	0.78
Education	Univ. of California	0.12

Fig. 2. Example of reconstructed data of *George Smith* with their relative weight.

the surname *Smith* (i.e., $j = 1$), which has a length $p_1 = 5$ and a weight $k_1 = 1$ (see Table 2). Starting from the values achieved for F_{ρ_2} and F_{ρ_3}, we have that the values S_{ρ_2} and S_{ρ_3} are equal to 0.64 and 0.54, respectively.

To summarize, the evaluation methods involved in SODA ADVANCE mainly focus on the password structure but lack the ability to evaluate the semantic meaning. To cover this gap, we introduce Large Language Models (LLMs) in order to evaluate the strength of passwords also from a semantic perspective using the reconstructed data. To this end, in the following sections, we first introduce the LLMs chosen for our study and then present three different pipelines that combine SODA ADVANCE and LLMs to explore the relationship between publicly available data on social networks and the possible passwords specified by a user.

5. Evaluating password with large language models

As discussed in the previous sections, many methods and metrics proposed in the literature are limited to evaluating the strength of passwords from a syntactic perspective, even when they start considering sensitive user information. SODA ADVANCE exploits some of these methods by introducing an evaluation module that is based on public data from social networks. However, to evaluate the semantics of passwords, we combine the effectiveness of SODA ADVANCE with some of the most recent LLMs, i.e., Google Gemini, ChatGPT, Claude, Dolly, Falcon, and LLaMa. In this section, we first provide an overview of the LLMs involved in our study and then we discuss the problem of interacting with them for both generating and evaluating the strength of passwords.

5.1. Overview of the large language models

The capabilities of LLMs enable the analysis and extraction of previously unknown patterns across diverse domains. These advanced pattern recognition and generative capabilities make LLMs a useful tool for evaluating password strength based on public information available online. Among the most recent LLMs, we choose to involve both open-source and proprietary LLMs in order to provide a comprehensive overview of the characteristics and performance of both types of LLMs in the password strength analysis. In what follows, we provide an overview of each LLM involved in our study.

ChatGPT is an LLM developed by OpenAi[3] and released in 2022. It relies on a transformer-based architecture trained on a vast amount of textual data from diverse sources on the Internet. This includes websites, books, articles, and other written content in multiple languages. ChatGPT excels in natural language understanding and generation tasks, allowing it to engage in conversations, answer questions, provide recommendations, and even generate creative content. It has been widely adopted for various applications including intelligent diagnoses, virtual assistants, and language translation. **Claude** is a non-open source LLM, created by Anthropic,[4] capable of managing more than 100.000 tokens in each prompt. Claude is highly competitive with respect to other LLMs and allows interaction using both textual and multimodal prompts. The latter represents one of the strengths of Claude, enabling it to directly process requests and prompts on textual files and datasets, further improving its applicability in different use cases.

Dolly is an open-source LLM, developed by Databricks,[5] built upon the EleutherAI Pythia architecture and fine-tuned on a 15,000-record instruction/response dataset. It is extremely efficient in different domains ranging from brainstorming and information extraction to content generation and question answering.

Falcon is an open-source LLM developed by the Technology Innovation Institute (TII)[6] in Abu Dhabi. It is trained using 40 billion parameters with the RefinedWeb dataset, which includes high-quality data extracts taken from websites, containing about 1 trillions of tokens [34]. Specifically, the pretraining data consisted mainly of public data, with some of them taken from scientific papers and conversations on social media platforms.

LLaMa is an open-source LLM developed by Meta AI,[7] trained on more than 2 trillion tokens, and fine-tuned on over 1 million human annotations. With its ability to comprehend and generate text, the LLM LLaMa 2 showcases remarkable proficiency in different tasks, such as text classification, sentiment analysis, named entity recognition, document summarization, question-answering, and machine translation.

Google Gemini is the most recent LLM developed by Google DeepMind.[8] It includes a sophisticated and multifaceted skill set, which encompasses a wide range of tasks. This advanced LLM stands out as a remarkable technological breakthrough, capable of composing various forms of creative content and providing comprehensive and informative responses to queries. Additionally, Gemini's multimodal processing capabilities enable it to seamlessly understand, operate across, and integrate various types of information, including text, code, audio, images, and videos.

In our study, we involve all described LLMs with the aim of investigating both their ability to generate passwords based on user information and evaluating the strength of passwords based on public information of users available on social networks.

[3] www.chat.openai.com.

[4] www.claude.ai.

[5] www.databricks.com.

[6] www.falconllm.tii.ae.

[7] www.llama.meta.com.

[8] www.deepmind.google.

5.2. Interacting with LLMs for evaluating password strength

Interacting with LLMs is an extremely challenging task, since it requires defining appropriate prompts to interact with these models. Their design can affect performance and the consistency of the user responses. A prompt consists of sentences written in natural language that should describe the context in which the LLM has to answer, and how it has to construct the response. The structure of prompts must be properly formed to clearly express the goals of the request and achieve an answer as close as related to the request itself.

There are two main categories of prompts, i.e., textual and multimodal. The first one is composed of only textual content and is considered the most natural way to interact with the LLMs; whereas a multimodal prompt can be composed by both textual content and files.

In the literature, several prompt engineering methodologies have been defined, also known as *Prompt Template Engineering* [35], which have been applied in different natural language processing tasks, such as machine translation, text summarization, and classification [36,37].

In the context of password generation and evaluation, It is worth noting that the capabilities of LLMs can be used to infer passwords of users who have publicly available data on social network platforms. Indeed, by combining the capabilities of LLMs with focused prompts, these models can be unconsciously used as malicious tools. Moreover, it is necessary to consider that many users are increasingly relying on LLMs in their daily activities, also involving these models in the processes of defining passwords to follow the updated security standards of different social network platforms [38]. However, most users do not care or do not have the technical knowledge to define precise prompts, completely relying on the answers provided by LLMs based on their requests. To this end, to interact with the LLMs in this study, we mainly rely on the *Automated Template Learning* approach which aims to require each LLM to generate a prompt to interact with itself in order to perform a specific task [35]. This approach enables each LLM to generate a prompt template based on their knowledge, which should be more suitable for it than a human-written prompt [39].

More specifically, we consider three different interaction pipelines aiming to combine SODA ADVANCE and LLMs to evaluate password strength based on publicly available information of a set of users. Furthermore, through these pipelines, we try to study if the combined usage of data reconstruction tools and LLMs could represent a risk for user passwords. In the following sections, we introduce the three pipelines defined for our study by also highlighting their components and the new ad-hoc prompting functions generated for interacting with LLMs. Moreover, we investigate the potential risks posed by integrating data reconstruction tools with LLMs, trying to highlight the implications for user password security in the digital landscape.

6. Combining SODA ADVANCE and LLMs for evaluating passwords

In this section, we describe the design and implementation of the three proposed pipelines that allow us to combine the capabilities of LLMs with those of SODA ADVANCE in order to address password strength generation and evaluation problems. The first pipeline focuses on using LLMs to generate strong and easy-to-remember passwords based on personal information provided by the user. Through ad-hoc prompt engineering strategies, the pipeline interacts with LLMs to generate passwords that prevent the insertion of sensitive data, while remaining easy to remember. The second pipeline aims to investigate the effectiveness of LLMs in evaluating password strength based on sensitive user information. Finally, the third pipeline aims to evaluate the impact of reconstructing and extracting public user data available on social networks on the capability of LLMs to generate and evaluate the strength of passwords.

6.1. Generation pipeline

Users who daily use different platforms that require an account, often struggle with the task of defining multiple strong passwords. This is often an extremely challenging task as it requires defining passwords that are easy to remember and have complex formats in order to comply with the different guidelines imposed by each platform. Moreover, since these guidelines evolve and become more stringent to combat emerging threats, users often find it challenging to generate and manage passwords that meet these requirements. To this end, the use of LLMs could be a promising solution, since they offer users the ability to effortlessly generate complex passwords easy to remember based on the user's information, and that meet established evolving security criteria of different platforms. In this scenario, the first pipeline is designed to investigate the capabilities of LLMs to generate strong passwords based on specific information provided by users.

Fig. 3 shows the process of generating passwords with LLMs and their evaluation using the evaluation module of SODA ADVANCE. The steps of the pipeline are discussed below:

- **Step ① - Password generation with LLMs**. Starting with the information provided by the user, we first ask each LLM to generate a set of strong passwords easy to remember based on the information provided as input.
- **Step ② - Password Evaluation with SODA ADVANCE**. For each generated password, we evaluate its strength using the evaluation module of SODA ADVANCE. This module reads the information of the user and computes the metrics discussed in Section 4 to check the strength of each password.
- **Step ③ - Password Labeling**. The results achieved by the evaluation module are then used to associate a label to each password, i.e., Weak or Strong, according to the resulting value S for each password.
- **Step ④ - Selection of an LLM**. After evaluating each password, we select the LLM with the greater number of strong passwords generated. This will be involved during the step of generating passwords in the next pipelines.

6.1.1. Prompt engineering approach for generating passwords

The process of generating passwords using LLMs has required the definition of an ad-hoc prompting function for interacting with LLMs. To this end, we have defined an ad-hoc function, namely $x' = f_{\text{password-generation}}(x_1, x_2, \ldots, x_n)$, which transforms a template into a prompt x' by replacing the input slots $[\mathrm{I}_1]$,$[\mathrm{I}_2]$,...,$[\mathrm{I}_n]$ with the input text $x_1, x_2, \ldots, x_n$ and by introducing the generated passwords as values of the additional slot [P]. The template of the prompting function has been defined as follows:

Template of $f_{\text{password-generation}}(i_1, \ldots, i_n)$

> **On the basis of the following personal information:** $[\mathrm{I}_1]$,$[\mathrm{I}_2]$,...,$[\mathrm{I}_n]$
> **Could you generate a set of passwords that do not have to directly contain personal data, but must be easy for the user to memorize?** [P]

where $[\mathrm{I}_1]$,$[\mathrm{I}_2]$,...,$[\mathrm{I}_n]$ are the slots containing the information of a user, such as first name, surname, city, and/or date of birth. The slot [P] is the slot related to the response of the LLM containing the set generated passwords $\mathrm{P} = \{\rho_1, \rho_2, \ldots, \rho_k\}$. In what follows, we provide an example of the prompt used for the generation of passwords:

> **On the basis of the following personal information: [Name: George], [Surname: Smith], [City: Orange, California], [Date: 10/23/1994].**
> **Could you generate a set of passwords that do not have to directly contain personal data, but must be easy for the user to memorise?**

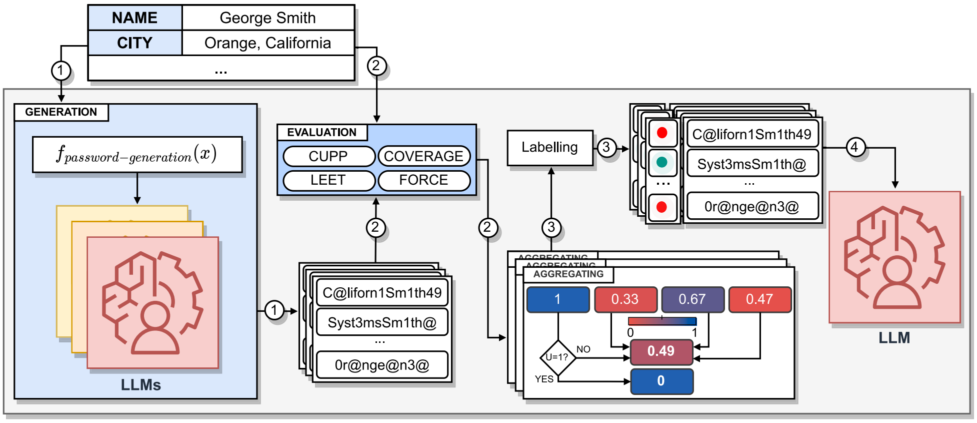


Fig. 3. Overview of the generation pipeline.

6.2. *Evaluation pipeline*

As discussed before, LLMs could represent a promising solution to generate strong passwords easy to remember based on a set of information related to users. However, an important consideration arises regarding the capabilities of LLMs to evaluate the actual strength of these passwords. In fact, although LLMs can generate passwords based on user-specific data that follow the security guidelines of different platforms, their effectiveness in evaluating password strength requires further investigation. To this end, the second pipeline aims to investigate the effectiveness of LLMs in assessing the strength of passwords based on sensitive user information. This process allows us to understand the ability of LLMs to perform classification tasks and to determine whether a password can be considered strong or not, also considering its semantics in relation to user data.

Fig. 4 provides an overview of the process of evaluating passwords based on information related to users. As we can see, we start by combining the strong passwords generated by the best LLM from the previous pipeline with those generated in a dictionary by means of the CUPP method, which examines the user information. The latter have been considered as not secure passwords since they are composed of common patterns or information easily accessible. The steps of the pipeline are discussed below:

- **Step ① - Passwords Generation with the LLM**. Starting from a set of user data, the LLM that generated the greatest number of strong passwords is used to define a new set of strong passwords adapted to user data.
- **Step ② - Password Generation with CUPP**. Starting from a set of user data, we generate the set of weak passwords using CUPP. In particular, CUPP reads user information and generates a dictionary of common words, containing easily guessable patterns combined with common words, phrases, and/or numerical values. The generated passwords follow standard patterns consisting of *[data] [special characters]*, *[data][calendar years]* or *[data][numeric value]*, which have been shown to be weak when applied in different scenarios [40–42].
- **Step ③ - Prompt Generation**. After the generation of weak and strong passwords, we evaluate the strength of passwords with the LLMs. To this end, we generate a new prompt for LLMs in order to enable each of them to evaluate the generated passwords.
- **Step ④ - Password Evaluation**. Each prompt is filled with the user data as input and it is submitted to an LLM together with the passwords to be evaluated and the corresponding user information. After this step, each password is associated with a short description that indicates whether the password is weak or strong according to the answers of each LLM.
- **Step ⑤ - Label Parsing**. Based on the textual rating provided by the LLMs, we ask each LLM to provide a strength score for each textual description associated with the passwords.
- **Step ⑥ - Evaluation Parsing**. To get a binary evaluation of the password strength, i.e., weak or strong, we associated each password with a strength score. Based on this score, the password is labeled as weak or strong.

Notice that, the steps ④ and ⑤ of the previous pipeline enable us to standardize the evaluations performed by LLMs. In fact, it is necessary to consider that either LLMs may associate different strength scores with similar labels or some labels are too general to consider a password as weak or strong, e.g., possibly secure or marginally secure. This heterogeneity in textual evaluation makes it difficult to directly compare the evaluations produced by different LLMs, requiring to introduce of a label parsing step to associate each label to a score and standardize the outputs provided by LLMs. To this end, we asked each LLM to associate a numerical value with the generated labels based on their inherent logic. After that, we are able to compare the results of different and identify if a password is considered weak or strong according to these scores. In what follows, we provide further details about these steps and on the scores associated with each label.

6.2.1. *Prompt engineering approach for evaluating password strength*

As discussed above, the process behind the password evaluation pipeline requires interacting with LLMs at several steps. In the first step, we rely on the prompt function defined in Section 6.1.1 for generating a set of passwords with the LLM. In the third step, we defined a new function $x' = f_{\text{prompt-generation}}(x)$ to ask each LLM to automatically generate prompts for password evaluation. The function replaces an input slot [M] with the input text x representing the name of the LLM for which we are generating the prompt. In the fifth step, we defined a new function $x' = f_{\text{parsing-prompt}}(x)$ to ask each LLM to provide a strength score for each textual description generated by each of them associated with the passwords. The template of this function has been defined as follows:

Template of $f_{\text{parsing-prompt}}(l_1,\ldots,l_g)$

I have the following security label for evaluate a password: $[\mathrm{L}_1],\ldots,[\mathrm{L}_g]$
Could rate each security label by giving each one a score level from 0 to 1, from most secure to least secure.

where $[\mathrm{L}_1],\ldots,[\mathrm{L}_g]$ are the slots containing the target labels achieved by each LLM as answers in the evaluation process. These labels have been considered independently for each LLM after requiring the evaluation

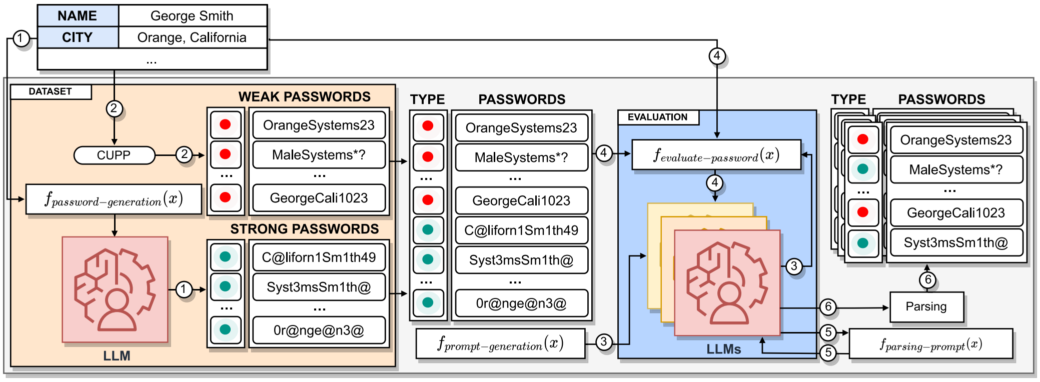


Fig. 4. Overview of the evaluation pipeline.

of password strength with automatically generated prompting functions using $x' = f_{\text{prompt-generation}}(x)$. The template of this function has been defined as follows:

Template of $f_{\text{prompt-generation}}(m)$

Write me a template for a prompt to be submitted to [M]**, which has as input a user's personal information and his password. The template has to be capable of evaluating whether the password provided is strong or otherwise.**
Consider that the template has to be written by an inexperienced user. The evaluation must be carried out exclusively on the basis of the input data.

where [M] is the slot of the LLM to be involved in the password strength evaluation. It is important to notice, that the slot [M] will be replaced with the name of the considered LLM for the generation of the prompt, i.e., with the LLM on we submitted the prompt.

The newly generated prompts enabled us to automatically create a new prompting function for each LLM involved in our study. All the new prompting functions transform a template into a prompt by replacing the slots [I$_1$],[I$_2$],...,[I$_n$] with the information of a user, [T$_1$], and [T$_2$] with the type of the results that we expect, i.e., *strong* or *week*, and [P$_1$],[P$_2$],...,[P$_k$] with the passwords to be evaluated. In what follows, we report the templates of the prompt automatically generated for each LLM. It is important to notice that, to avoid redundancy in discussions of prompt functions, we only provide an example and the explanation of slots for one function. The other functions will however be reported for the sake of clarity of the pipeline. The template of the prompt generated by ChatGPT has been defined as follows:

Template of $f_{\text{evaluate-password-GPT}}(i_1,\ldots,i_n,t_1,t_2,\rho_1,\ldots,\rho_k)$

User information: [I$_1$],[I$_2$],...,[I$_n$]
For each line containing a password that I could use for a social network account, give me an answer for each of them and write whether the password can be considered strong or not, giving [T$_1$], **or** [T$_2$]**. Assess the password's strength using the information supplied by the user, considering factors like its length and ability to resist guessing techniques.**
Passwords: [P$_1$],[P$_2$],...,[P$_k$]
[H$_1$],...,[H$_k$]

where [H$_1$],...,[H$_k$] are the slots containing the resulting target values $h_1, h_2, \ldots, h_k$, one for each password, such that $h_i \in \{$"*Strong*","*Weak*"$\}$ and $i = 1, 2, \ldots, k$. An example of this prompt is shown in the following:

User information: [Name: George], [Surname: Smith], [City: Orange, California], [Date: 10/23/1994], [Education: University of California].
For each line containing a password that I could use for a social network account, give me an answer for each of them and write whether the password can be considered secure or not, giving secure or not secure.
Assess the password's strength using the information supplied by the user, considering factors like its length and ability to resist guessing techniques.
Passwords: [OrangeSystems23], [MaleSystems*?], [GeorgeCali1023], [C@liforn1Sm1th49], [Syst3msSm1th@], [0r@nge@n3@]

The prompt template generated by Claude is shown below:

Template of $f_{\text{evaluate-password-Claude}}(i_1,\ldots,i_n,t_1,t_2,\rho_1,\ldots,\rho_k)$

User information: [I$_1$],[I$_2$],...,[I$_n$]
Please evaluate the following passwords for me: [P$_1$],[P$_2$],...,[P$_k$]
Using the user information provided, analyze the password strength based on factors such as length, complexity, incorporation of personal information, and resilience to brute force attacks or guessing. Then simply respond with [T$_1$]**, if you determine the password is strong and suitably protects the user information given or** [T$_2$] **if you determine the password could be vulnerable or guessed given the provided details.**
[H$_1$],...,[H$_k$]

The prompt template generated by Dolly is shown below:

Template of $f_{\text{evaluate-password-Dolly}}(i_1,\ldots,i_n,t_1,t_2,\rho_1,\ldots,\rho_k)$

With the following parameters:
User information: [I$_1$],[I$_2$],...,[I$_n$]
Passwords: [P$_1$],[P$_2$],...,[P$_k$]
Do you validate the password provided? Tell me if they are [T$_1$] **or** [T$_2$].
[H$_1$],...,[H$_k$]

The prompt template generated by Falcon is shown below:

Template of $f_{\text{evaluate-password-Falcon}}(i_1,\ldots,i_n,t_1,t_2,\rho_1,\ldots,\rho_k)$

Can you check this password for me? I'm still learning about making strong and secure passwords but want to make sure my accounts are protected.
Here is some information:
User information: [I$_1$],[I$_2$],...,[I$_n$]
Passwords: [P$_1$],[P$_2$],...,[P$_k$]
Can you please check if this password is [T$_1$] **or is** [T$_2$] **and if it needs to be stronger? Passwords are hard to remember so I want to make sure this one will keep my account safe without being too confusing.**
[H$_1$],...,[H$_k$]

The prompt template generated by LLaMa is shown below:

Template of $f_{\text{evaluate-password-LLaMa}}(i_1,\ldots,i_n,t_1,t_2,\rho_1,\ldots,\rho_k)$

I'm trying to make sure my password is secure, and I could use your help.
Could you please evaluate the strength of my password?
Here's my personal information: [I_1],[I_2],...,[I_n]
And here's my password: [P_1],[P_2],...,[P_k]
I'd really appreciate it if you could let me know if my password is [T_1] **or** [T_2].
[H_1],...,[H_k]

The prompt template generated by Google Gemini is shown below:

Template of $f_{\text{evaluate-password-Gemini}}(i_1,\ldots,i_n,t_1,t_2,\rho_1,\ldots,\rho_k)$

Input:
User's personal information: [I_1],[I_2],...,[I_n]
User's Passwords: [P_1],[P_2],...,[P_k]
Output: Examine the effectiveness of a password by evaluating its complexity, absence of personal information, and resilience against brute force attacks or guessing, using the data provided by the user. Then provide [T_1] **or** [T_2].
[H_1],...,[H_k]

The prompts generated by LLMs show a range of approaches to generate and evaluate password strength based on user information. Despite the similarity in their objectives, the prompts exhibit some differences in structure and language use. For instance, while prompts from ChatGPT and Claude begin with a direct request to evaluate passwords and emphasize factors like length, complexity, and resistance to guessing techniques, prompts from Dolly and Falcon adopt a more conversational tone, seeking validation of passwords in the context of user learning and account security concerns. Instead, the prompt provided by LLaMa expresses a request for generating text and passwords to ensure security. Concerning the prompt generated by Google Gemini, it explicitly focuses on technical aspects of password evaluation, emphasizing criteria such as complexity, absence of personal information, and resilience against brute force attacks or guessing. These types of automatically generated prompts reflect the interaction styles of each LLM and highlight the capabilities of these models in generating custom prompts for specific tasks and user needs.

6.3. *Data reconstruction and password evaluation pipeline*

After discussing the motivation behind the previous two pipelines, it is necessary to consider other two aspects related to password strength. First, it is necessary to address the fact that users registered on multiple social networks often have difficulty remembering which data they have shared publicly on different platforms and which is not privatized. Despite the option to privatize certain information, users frequently overlook or are unaware of the public accessibility of their data across multiple platforms. As discussed above, this publicly available data can be leveraged to infer passwords that are semantically linked to the user context, leading to a significant risk to password efficiency. In fact, when passwords are constructed based on personal information shared online, users could become vulnerable to targeted attacks exploiting the familiarity of the context to deduce the passwords. Thus, it becomes crucial to examine this problem within the context of password strength evaluation, highlighting the heightened susceptibility to breaches and emphasizing the necessity for robust password management practices. Secondly, it is necessary to consider that previous pipelines for password strength evaluation mainly focused on assessing either the semantics or syntax of passwords independently. However, the combined evaluation of both aspects can enhance the effectiveness of security measures, ensuring a higher level of protection against unauthorized access and data breaches. To this end, we combine the evaluation of password strength of the tool SODA ADVANCE with that of LLMs, using new automated prompting functions for evaluating passwords that consider both data reconstructed from the social networks and the results achieved by SODA ADVANCE within the prompt.

Fig. 5 shows an overview of the process underlying the third pipeline to generate a set of passwords in the context of a user and evaluate them with LLMs. Differently from previous processes, we do not completely rely on the knowledge underlying the LLMs in the evaluation phase, but we provide them an explanation of the metrics based on the definitions adopted in SODA ADVANCE. In fact, it has recently been shown that providing an explanation of some metrics or data to LLMs before performing a specific task can improve the understanding of LLMs in a conversation when used to address a problem in a specific domain [43]. As we can see, the process consists of the following steps:

- **Step ① - Data Reconstruction**: Starting from a small set of user information, we used SODA ADVANCE to reconstruct the information of a user publicly available from social network platforms.
- **Step ② - Password Database Creation**: The initial information with those reconstructed from SODA ADVANCE are used to create a dataset containing both strong and weak passwords associated with the user, following the strategy discussed in Section 6.2.
- **Step ③ - Passwords Evaluation with SODA ADVANCE**: The set of user passwords is provided to SODA ADVANCE that is responsible for the first evaluation of these passwords. For each password, SODA ADVANCE computes CUPP, LEET, COVERAGE, and FORCE values, and provides a final value of the password strength computing the $\mathcal{S}$.
- **Step ④ - Metrics Comprehension**: Before proceeding with the evaluation step, we provide an explanation for each of the metrics adopted by SODA ADVANCE to LLMs by means of a new prompt containing the explanation of each metrics.
- **Step ⑤ - Prompt Generation**: After providing the explanations of the metrics, we generate a new prompt for each LLM in order to enable each of them to evaluate the generated passwords. Differently from the previous pipeline, the automated generation of the evaluation prompts has considered both the resulting method values correlated to each password provided by SODA ADVANCE and data reconstructed from the social networks. It is important to notice that, the conversation in which we provide the explanation of the methods and the one in which we ask each LLM to evaluate the passwords are the same. This allows us to ensure that during the evaluation step, the explanation of the metrics is correctly assimilated by each LLM during the conversation.
- **Step ⑥ - Password Evaluation with LLMs**: Each prompt is filled with the user data reconstructed and the evaluation performed by SODA ADVANCE, and submitted to an LLM together with the passwords to be evaluated and the corresponding user information.
- **Steps ⑦ and ⑧ - Label and Evaluation Parsing**: Similarly to the steps ⑤ and ⑥ of the previous pipeline we associate each password with a strength score for each textual description in order to identify the type, strong or weak.

6.3.1. *Prompt engineering approach for password strength problem*

The interaction with LLMs for evaluating password strength has required to use of some of the previous prompting functions discussed in Sections 6.1.1 and 6.2.1, and the definition of new ones to explain the metrics to LLMs and evaluate the password also considering the results of SODA ADVANCE. In particular, we manually defined two new prompting functions following the *Manual Template Engineering* strategy [35], i.e., $f_{\text{understanding-metrics}}(x)$, $f_{\text{metrics-prompt-generation}}(x)$, and we automatically generated those to evaluate passwords for each LLM.

The prompting function $x' = f_{\text{understanding-metrics}}(x_1, x_2, x_3, x_4, x_5)$ aims to support LLMs in the comprehension of metrics used by SODA ADVANCE, i.e., CUPP, COVERAGE, LEET, FORCE, and $\mathcal{S}$. It transforms a template into a prompt x' by replacing the input slots [E_1], [E_2], [E_3], [E_4], and [E_5] with the textual explanation of the metrics x_1, x_2, x_3, x_4, and x_5. The template of the prompting function has been defined as follows:

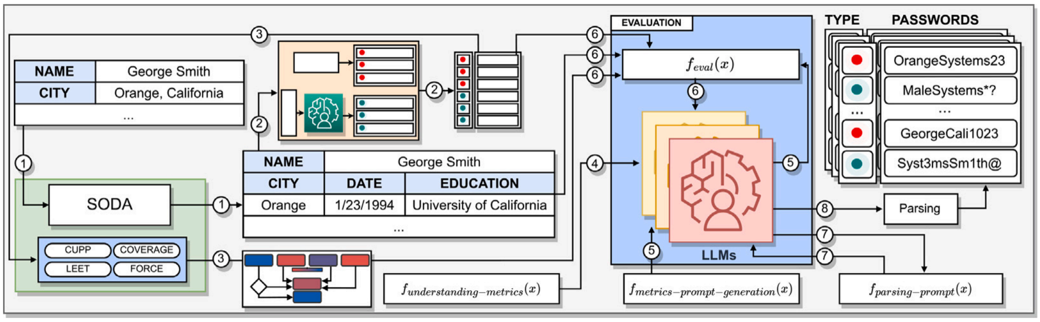


Fig. 5. Overview of the data reconstruction and password evaluation pipeline.

Template of $f_{\text{understanding-metrics}}(e_1, e_2, e_3, e_4, e_5)$

I will now provide you with an explanation of some of the methods used to evaluate Passwords Based on Personal Data from Social Networks Evaluation metrics used within a tool:
1. Leet: [E$_1$]
2. Coverage: [E$_2$]
3. CUPP (Common User Password Profiler): [E$_3$]
4. Force: [E$_4$]
5. Cumulative Password Strength: [E$_5$]

where the slots [E$_1$], [E$_2$], [E$_3$], [E$_4$], and [E$_5$] contains the detailed explanation of the metrics LEET, COVERAGE, CUPP, FORCE, and $\mathcal{S}$, respectively.

The second prompting function $x'' = f_{\text{metrics-prompt-generation}}(x)$ has been defined for the automatic generation of prompting functions to evaluate password strength. The function transforms a template into a prompt x'' by replacing the input slot [M] with the input text x representing the name of the LLM for which it must create a new prompt function. The template of the prompting function has been defined as follows:

Template of $f_{\text{metrics-prompt-generation}}(m)$

Write me a template for a prompt to be submitted to [M]**, which, having as input a user's personal information and his password and the corresponding evaluations metrics previously mentioned, i.e., Force, CUPP, Leet, Coverage, and Cumulative Password Strength.**
The template has to be capable of evaluating whether the password provided is strong or otherwise.
The evaluation must be carried out exclusively on the basis of the input data.

where [M] is the slot containing the name of the LLM used for generating the prompt, i.e. the same LLM on which the prompt has been submitted. Starting from the prompt, we automatically generate a new prompting function for each LLM. Similarly to the previous pipeline, we only provide an example of a generated prompting function with an explanation of the slots with the aim of avoiding redundancy in discussions. The template of the prompt generated by ChatGPT is shown below:

Template of $f_{\text{eval-GPT}}(i_1,\ldots,i_n,t_1,t_2,\rho_1,\ldots,\rho_k,f_1,\ldots,f_k,l_1,\ldots,l_k,c_1,\ldots,c_k,u_1,\ldots,u_k,s_1,\ldots,s_k)$

User information: [I$_1$], [I$_2$],..., [I$_n$]
Password, Force, Leet, Coverage, CUPP, CPS
[P$_1$], [F$_1$], [L$_1$], [C$_1$], [U$_1$], [$\mathcal{S}_1$]
...
[P$_k$], [F$_k$], [L$_k$], [C$_k$], [U$_k$], [$\mathcal{S}_k$]
Output:
Please assess the security of each password listed. Using the user information provided, analyze the password strength based on the following methods: Leet Coverage, Force, CUPP, and Cumulative Password Strength. Upon evaluation, please provide a response of [T$_1$] **if the password is deemed sufficiently strong and effectively safeguards the user's information based on the provided data, or** [T$_2$] **if the password could potentially be compromised or guessed based on the available details.**
[H$_1$],...,[H$_k$]

where [I$_1$],[I$_2$],...,[I$_n$] are the slots that will be replaced with the information of a user reconstructed from the social networks $i_1,\ldots,i_n$, and [P$_1$],[P$_2$],..., [P$_k$] represent the slots of generated passwords that will be replaced with $\rho_1,\ldots,\rho_k$. Each slot [P$_i$] is associated with the corresponding slots [F$_i$],[L$_i$],[C$_i$],[U$_i$], and [$\mathcal{S}_i$] containing the values of FORCE, LEET, COVERAGE, CUPP, and $\mathcal{S}$ related to the password, i.e., f_i, l_i, c_i, u_i, and s_i. The slots [T$_1$], [T$_2$] represent the type of results that we expect, i.e., *strong* or *weak*, whereas slots [H$_1$],...,[H$_k$] contains the resulting target values h$_1$,h$_2$,...,h$_k$, such that h$_i$ $\in$ {"*Strong*","*Weak*"} and $k = 1, 2, \ldots, k$. In what follows, we provide an example of the prompting function.

User information: [Name: George, Surname: Smith, City: Orange, California, Date: 10/23/1994, Education: University of California]
Passwords Evaluation Results:
Password; Force; Leet; Coverage; CUPP; CPS
OrangeSystems; 23; 57; 57; 0; 0.45
MaleSystems*?; 27; 2; 71; 1; 1
GeorgeCali1023; 63; 12; 76; 0; 0.50
C@liforn1Sm1th49; 65; 0; 83; 0; 0.49
Syst3msSm1th@; 54; 4; 57; 1; 1
0r@nge@n3@; 55; 25; 69; 0; 0.49
Please assess the security of each password listed. Using the user information provided, analyze the password strength based on the following methods: Leet Coverage, Force, CUPP, and Cumulative Password Strength. Upon evaluation, please provide a response of Strong **if the password is deemed sufficiently strong and effectively safeguards the user's information based on the provided data, or** Weak **if the password could potentially be compromised or guessed based on the available details.**

The template of the prompt generated by Google Gemini is shown below:

Template of $f_{\text{eval-Gemini}}(i_1,\dots,i_n,t_1,t_2,\rho_1,\dots,\rho_k,f_1,\dots,f_k,l_1,\dots,l_k,c_1,\dots,c_k,u_1,\dots,u_k,s_1,\dots,s_k)$

To evaluate the strength of a list of passwords, please provide the following information:
Personal information: [I_1],[I_2],...,[I_n]
Password List:
- Please provide a list of passwords that you would like to evaluate.
[P_1],[P_2],..., [P_k] **Evaluation Metrics:**
- Please provide the evaluation metrics for each password in the list. For each password, specify the corresponding values for Force, Leet, Coverage, CUPP, CPS:
[F_1], [L_1], [C_1], [U_1], [$\mathcal{S}_1$]
...
[F_k], [L_k], [C_k], [U_k], [$\mathcal{S}_k$]
Based on the provided data, we will assess the strength of each password in the list and determine if they adequately protect the user's information. Please note that the evaluation will be conducted solely based on the input data, and no external factors will be considered.
Based on the provided evaluation metrics for each password, please analyze the password strength and provide an output indicating whether each password is [T_1] **or** [T_2]. [H_1],...,[H_k]

The template of the prompt generated by Claude is shown below:

Template of $f_{\text{eval-Claude}}(i_1,\dots,i_n,t_1,t_2,\rho_1,\dots,\rho_k,f_1,\dots,f_k,l_1,\dots,l_k,c_1,\dots,c_k,u_1,\dots,u_k,s_1,\dots,s_k)$

User information: [I_1] , [I_2] ,..., [I_n]
Password, Force, Leet, Coverage, CUPP, CPS
[P_1], [F_1], [L_1], [C_1], [U_1], [$\mathcal{S}_1$]
...
[P_k], [F_k], [L_k], [C_k], [U_k], [$\mathcal{S}_k$]
For each password provided, assess its security based on the information provided by the user. Assess the strength of the password using Leet Coverage, Force, CUPP methods, and Cumulative Password Strength. Once assessed, classify each password as [T_1] **(if deemed sufficiently strong and effective in safeguarding user information) or** [T_2] **(if potentially vulnerable to compromise or guessing).** [H_1],...,[H_k]

The template of the prompt generated by Dolly is shown below:

Template of $f_{\text{eval-Dolly}}(i_1,\dots,i_n,t_1,t_2,\rho_1,\dots,\rho_k,f_1,\dots,f_k,l_1,\dots,l_k,c_1,\dots,c_k,u_1,\dots,u_k,s_1,\dots,s_k)$

User information: [I_1],[I_2],...,[I_n]
Password, Force, Leet, Coverage, CUPP, CPS
[P_1], [F_1], [L_1], [C_1], [U_1], [$\mathcal{S}_1$]
...
[P_k], [F_k], [L_k], [C_k], [U_k], [$\mathcal{S}_k$]
Output:
Please evaluate the security of each password in the list. Using the information provided, analyze the password strength based on the Leet Coverage, Force, CUPP methods, and Cumulative Password Strength. After evaluating, please respond with either strong or weak. If the password is strong and effectively protects the user's information based on the provided data, respond with [T_1]. **If the password could potentially be guessed or compromised based on the available details, respond with** [T_2].
[H_1],...,[H_k]

The template of the prompt generated by Falcon is shown below:

Template of $f_{\text{eval-Falcon}}(i_1,\dots,i_n,t_1,t_2,\rho_1,\dots,\rho_k,f_1,\dots,f_k,l_1,\dots,l_k,c_1,\dots,c_k,u_1,\dots,u_k,s_1,\dots,s_k)$

User, please provide your information below. Then list the passwords you would like evaluated along with the corresponding Force, Leet, Coverage, CUPP, and Cumulative Password Strength values for each password. I will assess the security of each password based on the details and metrics you supply.
User information: [I_1],[I_2],...,[I_n]
Password, Force, Leet, Coverage, CUPP, CPS
[P_1], [F_1], [L_1], [C_1], [U_1], [$\mathcal{S}_1$]
...
[P_k], [F_k], [L_k], [C_k], [U_k], [$\mathcal{S}_k$]
I will examine each password using the Force, Leet, Coverage and CUPP metrics provided to determine if the password is [T_1] **or** [T_2] **for protecting your personal information. A strong rating means the password is strong and complex based on the analysis methods and input data. A weak rating indicates the password may be vulnerable to guessing or cracking given the details supplied.**
[H_1],...,[H_k]

The template of the prompt generated by LLaMa is shown below:

Template of $f_{\text{eval-LLaMa}}(i_1,\dots,i_n,t_1,t_2,\rho_1,\dots,\rho_k,f_1,\dots,f_k,l_1,\dots,l_k,c_1,\dots,c_k,u_1,\dots,u_k,s_1,\dots,s_k)$

User, please provide your information below. Then list the passwords you would like evaluated along with the corresponding Force, Leet, Coverage, CUPP, and Cumulative Password Strength values for each password. I will assess the security of each password based on the details and metrics you supply. Provide [T_1] **if the password can be considered as strong or** [T_2] **if the password can be considered guessable.**
User information: [I_1],[I_2],...,[I_n]
1. **Password**: [P_1], **Force**: [F_1], **Leet**: [L_1], **Coverage**:[C_1], **CUPP**: [U_1], **CPS**: [$\mathcal{S}_1$]
...
k. **Password**: [P_k], **Force**: [F_k], **Leet**: [L_k], **Coverage**:[C_k], **CUPP**: [U_k], **CPS**: [$\mathcal{S}_k$]
[H_1],...,[H_k]

The prompts generated by LLMs for evaluating password strength show similarities in their structures but demonstrate differences in formatting and language style. Each prompt has been designed to consider a set of data reconstructed from the social networks, followed by a list of passwords to evaluate, along with corresponding metrics, such as FORCE, LEET, COVERAGE, CUPP, and CPS. The prompts enable each LLM to analyze password strength also considering these metrics as input to evaluate if a password is sufficiently strong or potentially weak. Nevertheless, although most of the prompting functions used in this study have been automatically generated by each LLM aiming to improve the capabilities of each to understand requests related to a specific problem, these functions are generalizable to any study concerning the generation and evaluation of passwords based on specific user data. In fact, it is possible to customize prompting functions by adding or removing users' information, passwords, and evaluation metrics. Moreover, these functions can be used considering different LLMs or a combination of them to evaluate password strength. In the following sections, we will show a case study involving real users that allows us to investigate the capabilities of SODA ADVANCE and LLMs to evaluate password strength.

7. Experimental evaluation

In this section, we try to answer the RQs discussed in Section 1 by performing different experimental sessions to evaluate the strength of the passwords specified by a set of users, based on personal data publicly available on social networks. As discussed in previous pipelines, for each session we perform comparative evaluations of several LLMs, namely Google Gemini, ChatGPT, Claude, Dolly, Falcon, and LLaMa, which represent some of the largest and most well-known proprietary and open-source LLMs.

7.1. Experimental settings and evaluation metrics

The experimental evaluations in this study aim to evaluate how password strength can be affected by the information publicly available on social network platforms from both syntactical and semantic perspectives. To this end, we investigate the behavior of SODA ADVANCE and generative LLMs following the three different pipelines discussed in the previous sections. The experimental evaluations conducted have required involving different users for gathering information on social network platforms. We involved 100 users, each of whom filled out an information survey and an authorization form for profiling their social network using SODA ADVANCE. The group of users was made up of 44 women and 56 men, of which 75 were in the range of 24 to 35, 25 were in the range of 36 to 47, and the remaining users were between 47 and 60. Users involved in our experiments come from European (56%), American (6%), Asian countries (4%), and the remaining part (34%) from other countries. All the users had different qualifications, such as bachelor's, master's, and Ph.D. degrees. Moreover, all of them had a LinkedIn profile, about 80% of them had a Facebook profile, and most of them had an Instagram account.

Experiment. Participants were provided a full explanation of the methodology and objectives of this work. The presentation included a comprehensive description of how the SODA ADVANCE can help users understand the severity of the risk to which passwords are exposed, along with a clear illustration of the main characteristics of the values calculated by the methods incorporated in the tool. In addition, a detailed explanation of the experiment sessions that involved LLMs was provided, specifying the data that they will evaluate. Among the questions submitted to users, we have required each of them to provide some personal information, such as name, surname, and photo. The data provided have been collected in a dataset and used as starting points in our experimental evaluations. It is important to notice that, the requests have been made to obtain the explicit consent of the users, in compliance with the data protection rules established by the GDPR. We have provided each user with complete information about how their data is used, guaranteeing their data protection rights. After processing the data necessary to generate the results, all the data collected from each user different from those provided in the initial step has been erased.

Concerning the usage of data with LLMs, we have informed users that their data were used only in a single conversation, and dropped at the end of the evaluation. However, we advised that LLMs should be regulated by European legislation on the processing of personal information and that we had no way to check that these regulations were actually complied with [44].

Technical settings. SODA ADVANCE was implemented using Python version 3.10.2 for the server side and using web programming frameworks for graphical interfaces. Concerning LLMs, we adopt ChatGPT in version 3.5.5, which is accessible through the official API provided by the OpenAI platform. This model employs the GPT-3.5 series based on GPT-3.5-turbo updated to March 2023. Instead, we involved Claude 2.0 using version 2.1 released in October 2023 which is available on the Anthropic platform. Regarding LLaMa, we used the model in version 2024.2.19.1 accessible via the official API available on the Meta Store platform. Falcon was used in its version at 40B, and Google Gemini in its version 1.0, which is accessible by interactive chat on the Google Bard platform. Concerning Dolly, we used the model in version Dolly-v2-12b built upon the EleutherAI Pythia architecture developed by Databricks. Moreover, for the analysis of the characteristics of the generated password we used *Passat*[9] and *Node-password-analyzer*[10] tools. Furthermore, to make a comparative evaluation with SODA ADVANCE, we use the *Zxcvbn library*[45] in its version 4.4.2, implemented with CoffeeScript,[11] Browserify [46], and Uglify-js,[12] the *CKL_PSM library*[47], and the *Semantic PCFG* [48] tool both written in Python language. The latter tool was trained on plaintext passwords extracted from the *Evite*[13] dataset. Finally, for generative password comparison, we use the *PassBERT model* [49], a generative model constructed on the top of Google BERT architecture.[14]

Communication costs with LLMs. The use of Large Language Models presents significant computational challenges. In fact, these models or require of advanced hardware infrastructures capable of handling the substantial computational load, or interacting with them by buying pay-as-you-go subscriptions to access the model APIs. In the latter case, users are charged based on their actual usage of the LLM, which can include costs per token generated, per API request, or per hour of usage. This approach provides flexibility and allows users to pay only for the resources they actually use.

The response speed of the APIs and the output generated by the LLMs vary depending on whether the models are run locally or via API services. When used locally, response times are directly affected by the performance of the available hardware. On the other hand, the use of external API services introduces variables such as network latency and computational process optimizations that are beyond the direct control of the user.

In our study, we used a local workstation for models that did not require high computational costs, with the following characteristics: 5 GHz Intel i9 CPU, 14 cores, and 64 GB of memory, equipped with an NVIDIA 3060 GPU with 6 GB of dedicated RAM memory. Instead, for larger models and proprietary models we subscribed a license that enabled us to interact with LLMs through API services. It is necessary to consider that the entire experimental evaluation required a large number of interactions with the LLMs to generate and evaluate passwords according to user data. The entire experimentation required an enormous effort due to the long time and continuous monitoring of the interaction processes.

Evaluation metrics. To evaluate the performances achieved by the models involved in our study, we considered Accuracy, Precision, Recall, and F1-score. These are defined in terms of the number of True Positives (TP), i.e., when a strong password was correctly identified; True Negatives (TN), i.e., when a weak password is correctly classified as not secure; False Positives (FP), i.e., when a weak password was identified as strong; False Negatives (FN), i.e., when a strong password was identified as weak. In what follows, we provide details about these metrics:

- **Accuracy**: Percentage of passwords successfully identified by the model so far: $Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$.
- **Precision**: The ratio of correctly identified positive passwords to all positive passwords in the positive class: $Precision = \frac{TP}{TP+FP}$.
- **Recall**: The ratio of correctly identified positive observations to all observations in the positive class: $Recall = \frac{TP}{TP+FN}$.
- **F1-score**: A weighted average of precision and recall, leading to take into account both FP and FN: $F1\text{-}Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$.

7.2. RQ1: Can we rely on LLMs to suggest strong and easy-to-remember passwords based on publicly available information on social networks?

As introduced above, the strength of passwords serves as a crucial point for protecting personal information and safeguarding digital data. The design of weak passwords can have far-reaching consequences, potentially jeopardizing user data on social media platforms and beyond. As cyber threats evolve and become more sophisticated, establishing strong passwords plays a crucial role in several areas. In this context, users must prioritize constructing strong and easy-to-remember passwords to protect their accounts. To investigate the capabilities of LLMs in password generation, we asked each LLM to generate 250 strong and easy-to-remember passwords for each user based on sensitive data, using the prompt engineering approach discussed in Section 6.1.1. To construct the generated password dataset we considered the basic set of information provided by users involved in our study.

Table 4 provides an overview of the characteristics of the generated passwords analyzed with *passat* and *node-password-analyzer* tools. The latter have analyzed different syntactic characteristics, such as length, types of characters used, and the types of starting and ending characters. For each of them, we report the percentage of passwords generated by LLMs starting from the initial set of user information. As we can see, most LLMs tend to consider strong passwords with the first uppercase letter, except for Gemini where the initial characters can also be a number. Moreover, only some passwords generated by ChatGPT, Claude, Dolly, and Falcon used special characters as the first characters. This is probably due to the fact that in several of these passwords the

[9] www.github.com/HynekPetrak/passat.

[10] www.github.com/T-PWK/node-password-analyzer.

[11] www.coffeescript.org.

[12] www.github.com/oyvindkinsey/UglifyJS.

[13] www.haveibeenpwned.com.

[14] www.github.com/google-research/bert.com.

Table 4
Statistics about the password generated by each LLM.

		ChatGPT (%)	Claude (%)	Dolly (%)	Falcon (%)	Gemini (%)	LLaMa (%)
Starting characters	Upper	99.04	95.35	98.3	95.35	75.5	98.8
	Lower	0.04	4.14	0.87	4.14	7.2	0.12
	Num	0.92	0.4	0.75	0.41	17.6	1.05
	Special characters	0.4	0.13	0.04	0.8	0	–
Ending characters	Letter	63.16	88.02	68.1	88.02	26.4	80.70
	Num	26.95	9.75	25.74	10.08	27.2	15.15
	Special characters	9.89	2.21	6.15	1.88	46.4	4.13
Word length	≤8	–	–	4.96	0.45	–	2.86
	>8 and <12	5.6	5.38	2.79	5.09	5.2	21.85
	≥12	94.2	94.4	92.25	94.4	94.8	75.26
Charsets and sequences	All Num	–	–	0.8	–	0.8	–
	All Lower	–	0.6	0.6	0.7	8.4	1.3
	Letter, Num	26.5	17.6	26.7	18	16.8	16.4
	Letter, Symbol	14.1	0.5	12.5	0.5	22.4	1.2
	Lower, Num, Symbol	–	0.2	4	0.2	4	–
	Upper, Lower	49.1	78.4	52.2	78.4	72.5	77.3
	Upper, Lower, Num	26.2	15.8	26.2	16.2	16.8	15
	Upper, Num, Symbol	0.6	–	0.6	–	–	–
	Upper, Lower, Symbol	27.2	0.5	11.3	0.4	22	1.2
	Upper, Lower, Num, Symbol	9.4	2.6	7.1	2.2	12.3	3.7
Most frequent symbol		! # $. _ @ + &	@ ! _ . - & % $ # ? *	_ # .	_ - ’ . *	_ @ $ # ’ !	@ ! $ # * & ’

actual letter is replicated using the leet method, e.g., S replaced with $, to increase the password strength.

Concerning the ending characters, we can see that most passwords generated by Claude, Falcon, and LLaMa end with a letter and among the passwords generated by those LLMs only a few of them end with special characters. Moreover, the word length analysis reveals interesting variations among the LLMs. As we can see, only LLaMa, Dolly, and Falcon have generated strong passwords with a length of less than 8, whereas most LLMs have generated passwords with a length greater than or equal to 12.

Regarding the charsets, we can see that among the passwords generated by Claude, Dolly, Gemini, and Falcon some of those considered as strong are composed of all numeric and lower characters. Instead, only a few passwords generated by LLMs contain all the different charsets, i.e., upper, lower, number, and symbol, except for passwords achieved by ChatGPT and Gemini.

The analysis has revealed that each LLM exhibits distinct patterns in the generation of strong passwords, with variations in syntactical complexity and the combination of letters and characters. Nevertheless, it is necessary to investigate their ability to evaluate passwords based on user information. To this end, we evaluate the strength of passwords using the evaluation metrics shown in Section 4. Table 5 shows the average $\mathcal{S}$ values achieved by each LLM on the passwords. Moreover, for each user, we report the data reconstructed from social networks that have been involved in the process of generating strong passwords. Claude, Google Gemini, and ChatGPT outperform the other LLMs achieving the highest number of strong passwords, i.e., 0.82, 0.75, and 0.74, respectively.

All three LLMs provide sufficiently strong and easy-to-remember passwords by outperforming all other LLMs. In particular, these models demonstrated their ability to generate strong passwords by incorporating personal references and interests into the password creation process, making it more meaningful and memorable for the user. The passwords generated by Google Gemini, Claude, and ChatGPT combine a wide range of characters, symbols, and numbers with some of the personal information of users.

On the other hand, Dolly, LLaMa, and Falcon have generated more weak passwords, achieving average values for $\mathcal{S}$ of 0.65, 0.66, and 0.66. These values were probably due to their tendency to generate repetitive or predictable passwords, using recurring and easily guessable patterns. In particular, the trend shown during generation is to generate passwords with sensitive information in a simple form, which can be easily traced back to users. Among these LLMs, Falcon and Dolly have shown a low ability to generate strong passwords related to the information of users, since in many cases passwords did not contain any information about the user, but only recurrent patterns, such as “*AdminPassWord492*” or “*SystemUser995*”.

The performance of LLMs in password generation seems to be affected by the size of the model and the corpora on which they are trained. ChatGPT, Claude, and Google Gemini were trained on large, diverse corpora, which seems to give them a deeper understanding of the language and enable them to generate more creative and original passwords. Instead, LLaMa, Falcon, and Dolly are trained on smaller, more specialized corpora, which may not be as representative of the full range of languages. These differences seem to affect the results of the generation tasks. Concerning Claude, it revealed good performances in generating sufficiently strong and easy-to-remember passwords and could be used as a user support tool with appropriate precautions.

7.3. RQ2: Can LLMs represent a valid tool to support users in evaluating the strength of passwords based on personal information?

LLMs have shown great potential for generating passwords, especially Claude which has achieved high performance in generating tasks. Therefore, to further investigate the capabilities of LLMs in the password evaluation task we conducted different assessments in identifying potential weaknesses in the passwords. Their extensive training using large textual corpora could enable LLMs to identify patterns and semantic correlations between passwords and personal user data. In particular, the evaluation was performed according to the processing pipeline shown in Sections Section 6.2 and the prompt engineering methodology discussed in Section 6.2.1. According to the procedure underlying the evaluation pipeline, we asked each LLM to provide a strength score for each textual evaluation (or target label) associated with passwords. Indeed, LLMs only associate a short description with each password that is often not very clear in terms of strength score, such as moderate secure, low secure, or possibly secure. This allowed us to clarify and standardize results between LLMs and ensure consistency in the evaluation process.

Fig. 6 shows the target labels extracted from the response of each LLM. The target labels range from 0 to 1, where 0 represents a weak password and 1 is a strong password. After a manual evaluation of the target labels, we consider passwords with a strength score greater than or equal to 0.55 as strong, i.e., *moderately secure* or *moderately secure*,

Table 5
Average S values achieved by LLMs on the passwords generated based on the reconstructed data.

UserID	#Password	Birthday	Current city	E-mail	Education	Family	Gender	Hometown	IG biography	Languages	Name	Surname	Web site	ChatGPT	Claude	Falcon	Dolly	LLaMa	Gemini	UserID	#Password	Birthday	Current city	E-mail	Education	Family	Gender	Hometown	IG biography	Languages	Name	Surname	Web site	ChatGPT	Claude	Falcon	Dolly	LLaMa	Gemini
0	250		🟦		🟦			🟦	🟦	🟦	🟦	🟦		± 0.72	± 0.77	± 0.65	± 0.63	± 0.63	± 0.72	50	250	🟦	🟦		🟦	🟦	🟦			🟦	🟦	🟦		± 0.84	± 0.92	± 0.68	± 0.74	± 0.74	± 0.85
1	250		🟦		🟦	🟦	🟦			🟦	🟦	🟦		± 0.69	± 0.79	± 0.66	± 0.6	± 0.62	± 0.74	51	250		🟦		🟦		🟦	🟦	🟦	🟦	🟦	🟦		± 0.74	± 0.8	± 0.64	± 0.64	± 0.64	± 0.74
2	250		🟦		🟦		🟦	🟦	🟦	🟦	🟦	🟦		± 0.69	± 0.76	± 0.6	± 0.6	± 0.61	± 0.7	52	250		🟦		🟦	🟦	🟦			🟦	🟦	🟦		± 0.69	± 0.8	± 0.63	± 0.6	± 0.61	± 0.74
3	250		🟦		🟦		🟦		🟦		🟦	🟦		± 0.75	± 0.81	± 0.67	± 0.65	± 0.66	± 0.74	53	250		🟦		🟦				🟦	🟦	🟦	🟦		± 0.81	± 0.83	± 0.67	± 0.71	± 0.71	± 0.8
4	250	🟦	🟦		🟦		🟦	🟦		🟦	🟦	🟦		± 0.73	± 0.8	± 0.65	± 0.63	± 0.64	± 0.74	54	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.76	± 0.84	± 0.67	± 0.66	± 0.67	± 0.78
5	250		🟦		🟦	🟦	🟦	🟦	🟦		🟦	🟦		± 0.68	± 0.74	± 0.6	± 0.6	± 0.59	± 0.68	55	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.78	± 0.91	± 0.69	± 0.68	± 0.72	± 0.82
6	250		🟦		🟦		🟦	🟦	🟦	🟦	🟦	🟦		± 0.84	± 0.93	± 0.74	± 0.74	± 0.74	± 0.85	56	250		🟦		🟦		🟦				🟦	🟦		± 0.78	± 0.83	± 0.71	± 0.68	± 0.7	± 0.76
7	250	🟦	🟦		🟦	🟦	🟦		🟦		🟦	🟦		± 0.76	± 0.85	± 0.68	± 0.66	± 0.67	± 0.77	57	250		🟦		🟦	🟦	🟦		🟦	🟦	🟦	🟦		± 0.75	± 0.76	± 0.65	± 0.65	± 0.65	± 0.75
8	250		🟦		🟦		🟦		🟦	🟦	🟦	🟦		± 0.66	± 0.73	± 0.58	± 0.67	± 0.58	± 0.67	58	250		🟦		🟦	🟦	🟦		🟦	🟦	🟦	🟦		± 0.73	± 0.82	± 0.66	± 0.63	± 0.64	± 0.75
9	250		🟦		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦		± 0.75	± 0.9	± 0.69	± 0.66	± 0.73	± 0.83	59	250	🟦	🟦		🟦		🟦	🟦	🟦	🟦	🟦	🟦		± 0.7	± 0.79	± 0.62	± 0.6	± 0.63	± 0.69
10	250		🟦		🟦		🟦		🟦	🟦	🟦	🟦		± 0.74	± 0.8	± 0.65	± 0.65	± 0.64	± 0.7	60	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.87	± 0.94	± 0.66	± 0.76	± 0.76	± 0.87
11	250		🟦		🟦		🟦				🟦	🟦		± 0.75	± 0.84	± 0.69	± 0.66	± 0.67	± 0.76	61	250	🟦	🟦			🟦	🟦		🟦		🟦	🟦		± 0.84	± 0.86	± 0.74	± 0.73	± 0.73	± 0.84
12	250	🟦	🟦		🟦		🟦				🟦	🟦	🟦	± 0.74	± 0.82	± 0.66	± 0.65	± 0.65	± 0.75	62	250	🟦	🟦		🟦	🟦	🟦		🟦		🟦	🟦		± 0.75	± 0.81	± 0.66	± 0.65	± 0.65	± 0.74
13	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.89	± 0.95	± 0.81	± 0.82	± 0.82	± 0.85	63	250		🟦		🟦		🟦		🟦		🟦	🟦		± 0.77	± 0.83	± 0.68	± 0.67	± 0.66	± 0.73
14	250		🟦		🟦		🟦		🟦	🟦	🟦	🟦		± 0.73	± 0.79	± 0.62	± 0.63	± 0.64	± 0.73	64	250		🟦		🟦		🟦	🟦	🟦	🟦	🟦	🟦		± 0.76	± 0.84	± 0.66	± 0.66	± 0.68	± 0.78
15	250		🟦		🟦		🟦	🟦			🟦	🟦		± 0.72	± 0.8	± 0.62	± 0.63	± 0.63	± 0.71	65	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.85	± 0.93	± 0.77	± 0.74	± 0.77	± 0.87
16	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.84	± 0.92	± 0.72	± 0.73	± 0.74	± 0.83	66	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.75	± 0.84	± 0.68	± 0.66	± 0.65	± 0.77
17	250		🟦			🟦	🟦				🟦	🟦		± 0.79	± 0.96	± 0.78	± 0.69	± 0.77	± 0.87	67	250		🟦		🟦		🟦				🟦	🟦		± 0.74	± 0.83	± 0.68	± 0.64	± 0.66	± 0.77
18	250		🟦		🟦		🟦		🟦	🟦	🟦	🟦		± 0.8	± 0.87	± 0.7	± 0.7	± 0.69	± 0.8	68	250		🟦		🟦		🟦	🟦	🟦		🟦	🟦		± 0.81	± 0.88	± 0.69	± 0.7	± 0.7	± 0.8
19	250		🟦		🟦		🟦				🟦	🟦		± 0.75	± 0.85	± 0.64	± 0.66	± 0.67	± 0.77	69	250	🟦	🟦				🟦				🟦	🟦		± 0.69	± 0.76	± 0.6	± 0.69	± 0.61	± 0.7
20	250		🟦		🟦	🟦	🟦		🟦	🟦	🟦	🟦		± 0.73	± 0.81	± 0.64	± 0.64	± 0.65	± 0.74	70	250		🟦		🟦		🟦		🟦	🟦	🟦	🟦		± 0.73	± 0.81	± 0.65	± 0.64	± 0.65	± 0.74
21	250	🟦	🟦		🟦		🟦			🟦	🟦	🟦		± 0.62	± 0.69	± 0.54	± 0.54	± 0.55	± 0.63	71	250		🟦				🟦				🟦	🟦		± 0.74	± 0.8	± 0.63	± 0.64	± 0.63	± 0.74
22	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.73	± 0.79	± 0.64	± 0.64	± 0.64	± 0.72	72	250		🟦		🟦		🟦		🟦	🟦	🟦	🟦		± 0.82	± 0.91	± 0.74	± 0.72	± 0.73	± 0.83
23	250		🟦		🟦	🟦	🟦		🟦	🟦	🟦	🟦		± 0.71	± 0.77	± 0.56	± 0.62	± 0.63	± 0.72	73	250		🟦		🟦		🟦		🟦	🟦	🟦	🟦		± 0.82	± 0.91	± 0.73	± 0.72	± 0.72	± 0.81
24	250	🟦	🟦			🟦	🟦	🟦			🟦	🟦		± 0.72	± 0.78	± 0.63	± 0.63	± 0.63	± 0.73	74	250		🟦		🟦		🟦	🟦		🟦	🟦	🟦		± 0.75	± 0.83	± 0.61	± 0.65	± 0.66	± 0.76
25	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.74	± 0.78	± 0.65	± 0.65	± 0.65	± 0.74	75	250		🟦		🟦		🟦	🟦		🟦	🟦	🟦		± 0.87	± 0.96	± 0.75	± 0.76	± 0.76	± 0.89
26	250		🟦		🟦	🟦	🟦			🟦	🟦	🟦		± 0.74	± 0.85	± 0.65	± 0.64	± 0.68	± 0.78	76	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.82	± 0.91	± 0.71	± 0.71	± 0.73	± 0.79
27	250		🟦			🟦	🟦	🟦			🟦	🟦		± 0.72	± 0.81	± 0.65	± 0.64	± 0.64	± 0.72	77	250		🟦		🟦		🟦		🟦		🟦	🟦		± 0.79	± 0.89	± 0.73	± 0.69	± 0.73	± 0.81
28	250		🟦		🟦		🟦	🟦		🟦	🟦	🟦		± 0.71	± 0.83	± 0.67	± 0.62	± 0.63	± 0.76	78	250		🟦		🟦		🟦				🟦	🟦		± 0.71	± 0.8	± 0.66	± 0.62	± 0.64	± 0.73
29	250		🟦		🟦		🟦	🟦	🟦		🟦	🟦		± 0.75	± 0.9	± 0.69	± 0.66	± 0.73	± 0.83	79	250		🟦		🟦		🟦				🟦	🟦		± 0.57	± 0.63	± 0.5	± 0.5	± 0.49	± 0.56
30	250		🟦		🟦		🟦		🟦	🟦	🟦	🟦		± 0.71	± 0.78	± 0.59	± 0.62	± 0.63	± 0.71	80	250		🟦		🟦		🟦	🟦		🟦	🟦	🟦		± 0.85	± 0.94	± 0.76	± 0.75	± 0.75	± 0.86
31	250		🟦		🟦		🟦		🟦	🟦	🟦	🟦		± 0.73	± 0.79	± 0.65	± 0.63	± 0.63	± 0.73	81	250		🟦		🟦		🟦				🟦	🟦		± 0.72	± 0.84	± 0.69	± 0.63	± 0.68	± 0.75
32	250		🟦		🟦				🟦	🟦	🟦	🟦		± 0.76	± 0.82	± 0.68	± 0.66	± 0.66	± 0.76	82	250		🟦		🟦		🟦		🟦		🟦	🟦		± 0.85	± 0.94	± 0.77	± 0.75	± 0.75	± 0.85
33	250		🟦		🟦				🟦	🟦	🟦	🟦		± 0.7	± 0.77	± 0.64	± 0.61	± 0.63	± 0.72	83	250		🟦		🟦		🟦		🟦	🟦	🟦	🟦		± 0.83	± 0.92	± 0.66	± 0.73	± 0.73	± 0.83
34	250		🟦		🟦				🟦	🟦	🟦	🟦		± 0.63	± 0.71	± 0.5	± 0.55	± 0.56	± 0.62	84	250		🟦		🟦	🟦	🟦			🟦	🟦	🟦		± 0.84	± 0.88	± 0.73	± 0.74	± 0.74	± 0.82
35	250		🟦		🟦		🟦		🟦	🟦	🟦	🟦		± 0.86	± 0.95	± 0.73	± 0.75	± 0.75	± 0.86	85	250		🟦		🟦		🟦	🟦		🟦	🟦	🟦		± 0.79	± 0.81	± 0.66	± 0.69	± 0.67	± 0.78
36	250		🟦		🟦		🟦		🟦	🟦	🟦	🟦		± 0.75	± 0.83	± 0.57	± 0.65	± 0.68	± 0.77	86	250		🟦			🟦	🟦	🟦		🟦	🟦	🟦		± 0.77	± 0.84	± 0.7	± 0.79	± 0.73	± 0.78
37	250		🟦				🟦	🟦	🟦		🟦	🟦		± 0.64	± 0.69	± 0.58	± 0.57	± 0.56	± 0.65	87	250	🟦	🟦				🟦				🟦	🟦		± 0.73	± 0.79	± 0.67	± 0.63	± 0.66	± 0.76
38	250		🟦		🟦		🟦	🟦		🟦	🟦	🟦		± 0.66	± 0.73	± 0.58	± 0.57	± 0.58	± 0.64	88	250		🟦			🟦	🟦	🟦		🟦	🟦	🟦		± 0.74	± 0.76	± 0.65	± 0.65	± 0.66	± 0.76
39	250		🟦		🟦	🟦	🟦		🟦	🟦	🟦	🟦		± 0.74	± 0.82	± 0.58	± 0.65	± 0.66	± 0.75	89	250		🟦			🟦	🟦	🟦		🟦	🟦	🟦		± 0.7	± 0.72	± 0.62	± 0.61	± 0.62	± 0.7
40	250		🟦		🟦						🟦	🟦		± 0.8	± 0.87	± 0.71	± 0.7	± 0.7	± 0.8	90	250	🟦	🟦		🟦		🟦		🟦		🟦	🟦		± 0.70	± 0.81	± 0.74	± 0.75	± 0.72	± 0.72
41	250		🟦		🟦		🟦		🟦	🟦	🟦	🟦		± 0.79	± 0.85	± 0.69	± 0.69	± 0.69	± 0.78	91	250		🟦		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	± 0.73	± 0.70	± 0.75	± 0.66	± 0.61	± 0.76
42	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.68	± 0.75	± 0.58	± 0.6	± 0.61	± 0.69	92	250		🟦		🟦		🟦	🟦		🟦	🟦	🟦		± 0.76	± 0.76	± 0.72	± 0.71	± 0.76	± 0.86
43	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.79	± 0.86	± 0.7	± 0.69	± 0.69	± 0.79	93	250	🟦		🟦					🟦	🟦	🟦	🟦		± 0.73	± 0.75	± 0.75	± 0.77	± 0.71	± 0.88
44	250	🟦	🟦		🟦		🟦		🟦	🟦	🟦	🟦		± 0.84	± 0.91	± 0.75	± 0.73	± 0.74	± 0.85	94	250		🟦		🟦	🟦	🟦		🟦		🟦	🟦		± 0.73	± 0.73	± 0.71	± 0.76	± 0.71	± 0.79
45	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.71	± 0.71	± 0.63	± 0.61	± 0.62	± 0.71	95	250	🟦				🟦				🟦		🟦	🟦	± 0.72	± 0.74	± 0.64	± 0.62	± 0.57	± 0.76
46	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.76	± 0.87	± 0.71	± 0.66	± 0.7	± 0.79	96	250		🟦				🟦	🟦	🟦			🟦		± 0.51	± 0.52	± 0.47	± 0.50	± 0.45	± 0.71
47	250		🟦		🟦		🟦			🟦	🟦	🟦		± 0.87	± 0.95	± 0.76	± 0.76	± 0.77	± 0.87	97	250				🟦	🟦	🟦	🟦	🟦	🟦		🟦		± 0.63	± 0.62	± 0.52	± 0.57	± 0.48	± 0.66
48	250		🟦		🟦		🟦		🟦		🟦	🟦		± 0.74	± 0.79	± 0.65	± 0.65	± 0.63	± 0.74	98	250	🟦	🟦		🟦	🟦	🟦		🟦	🟦		🟦		± 0.75	± 0.75	± 0.75	± 0.75	± 0.75	± 0.75
49	250		🟦		🟦	🟦	🟦				🟦	🟦		± 0.66	± 0.7	± 0.61	± 0.57	± 0.61	± 0.69	99	250		🟦				🟦	🟦		🟦		🟦	🟦	± 0.70	± 0.68	± 0.55	± 0.63	± 0.54	± 0.79

🟦 = The data has been reconstructed from social networks.

while the other as weak. Following this strategy, we are able to get a binary evaluation of passwords and compare the results achieved by LLMs with those achieved by methods proposed in the state-of-the-art (Section 4.1).

As we can see, Falcon and LLaMa provide fewer labels and strength scores, which means they tend to consider a coarse-grained password strength evaluation. Other LLMs consider a larger number of labels, meaning that they provide a more fine-grained assessment of password strength, which is probably due to the size of this LLM. It is important to notice that, although the number of labels differs among LLMs, their distribution is quite balanced between scores indicating strong passwords and those indicating weak passwords are similar. Regarding the types of labels identified by LLMs, as we can see, different labels often correspond to the same score. This is why we related the label types to a numerical value defined by the LLM itself.

Table 6 shows the performances achieved by the LLMs in terms of accuracy, precision, recall, and F1-score. In particular, we report the average value achieved by each LLM considering all users involved in our evaluation. As we can see, Claude achieves the highest values for accuracy, precision, and F1-score, with values of 0.75, 0.76, and 0.75, respectively. This indicates that Claude accurately identifies strong passwords while minimizing false positives and false negatives. The high precision score of 0.76 indicates that it has a low rate of false positives, meaning it correctly identifies strong passwords with a high degree of confidence.

Concerning the results of Google Gemini, it achieves performances of 0.60, 0.60, 0.62, and 0.63 for accuracy, precision, recall, and F1-score, respectively. These metrics indicate that Google Gemini effectively identifies strong passwords while maintaining a balance between precision and recall. Similarly to Claude, it is able to correctly evaluate the majority of passwords with a relatively low rate of false positives. However, the higher recall score indicates that Google Gemini better identified a high number of actual strong passwords among those considered.

Regarding ChatGPT, Dolly, LLaMa, and Falcon, they achieve lower performances than the other LLMs, with values for all metrics lower than 0.58. The worst results have been obtained for the recall metric, which shows the poor ability of these models to correctly identify strong passwords among all actual strong passwords in the dataset. These models exhibited a higher number of False Negatives (FN) compared to True Positives (TP), indicating that they often failed to recognize strong passwords based on user data, leading to a significant proportion of weak passwords being incorrectly classified as strong. These models show weaknesses, especially in the recall, which shows that they have some difficulties in detecting password strengths with respect to user data.

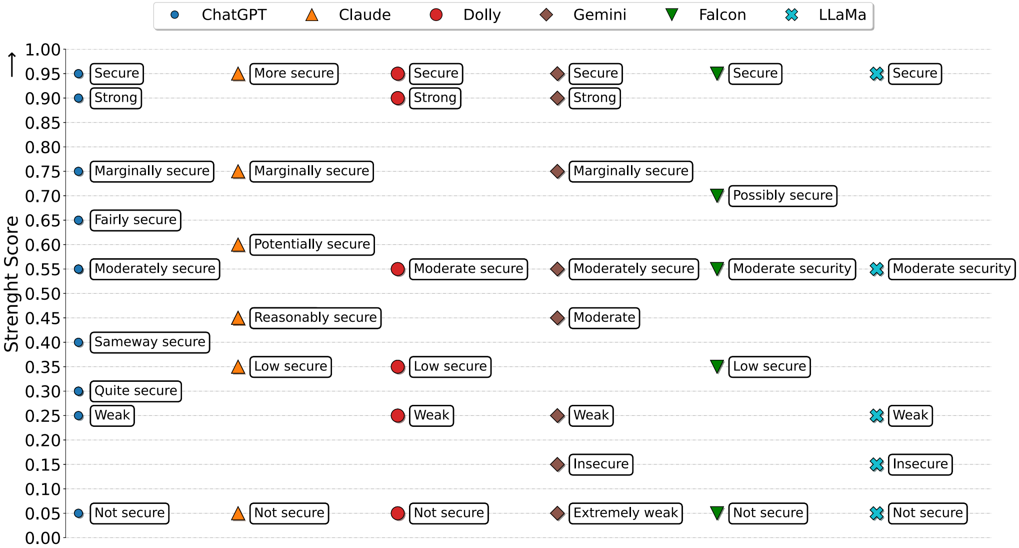


Fig. 6. Strength scores associated with the target labels provided by each LLM.

Table 6
Results achieved by LLMs in evaluating password strength.

Model	Accuracy	Precision	Recall	F1-score
ChatGPT	±0.58	±0.58	±0.56	±0.57
Claude	**±0.75**	**±0.76**	**±0.75**	**±0.75**
Dolly	±0.49	±0.49	±0.41	±0.45
Falcon	±0.49	±0.48	±0.41	±0.44
LLaMa	±0.52	±0.53	±0.41	±0.47
Google Gemini	±0.60	±0.60	±0.62	±0.61
All LLMs	±0.57	±0.57	±0.51	±0.54
Best 3 LLMs	**±0.63**	**±0.63**	**±0.62**	**±0.63**

Table 7
Results achieved by LLMs in evaluating password strength considering data reconstructed by SODA ADVANCE.

Model	Accuracy	Precision	Recall	F1-score
ChatGPT	±0.76	±0.88	±0.79	±0.83
Claude	**±0.77**	**±0.89**	**±0.79**	**±0.85**
Dolly	±0.55	±0.79	±0.46	±0.58
Falcon	±0.52	±0.77	±0.52	±0.62
LLaMa	±0.56	±0.77	±0.59	±0.67
Google Gemini	±0.68	±0.80	±0.73	±0.78
All LLMs	±0.60	±0.78	±0.65	±0.70
Best 3 LLMs	**±0.63**	**±0.82**	**±0.65**	**±0.72**

To further investigate if the ensemble of different LLMs improves the values of metrics, we consider two different ensembles: (*i*) considering all the LLMs and (*ii*) considering the three LLMs with higher scores. As we can see in Table 6, both ensembles performed lower than Claude, with values of 0.57, 0.51 and 0.55 for accuracy, recall, and F1-score, respectively, when considering the set of all LLMs, while 0.63, 0.62, and 0.63 for the three best LLMs. Only the precision values of both ensembles have increased with respect to the results of each LLM. This means that they are better able to identify strong passwords with high levels of strength, but if we consider the lower recall values, we can see that the ensembles fail to identify a number of truly strong passwords among those considered.

Generally speaking, when analyzing the performance of LLMs on classification tasks, it is essential to consider the wide variety of data on which they are trained and the resulting acquired capabilities. Unlike traditional machine learning models or evaluation metrics proposed in the literature that are conceived for specific tasks, LLMs, such as Claude, have developed an adaptive understanding of language models, semantics, and relationships due to their pre-training methodology. This allows them to better generalize and transfer their knowledge to new classification problems, such as evaluating password strength and leveraging their understanding of language, personal information and their correlations. Overall, based on the results achieved in the evaluation of password strength, Claude has demonstrated to be a valid tool to support users in evaluating the strength of passwords based on personal information. However, it is necessary to evaluate their effectiveness when they are used in combination with other metrics to evaluate password strength and a broader set of user information, to provide more accurate assessments considering both syntactic and semantic perspectives.

7.4. RQ3: How does the public availability of personal information across multiple social networks impact the capabilities of LLMs to generate and evaluate password strength?

After evaluating the capabilities of LLMs to both generate and evaluate password strength based on a set of personal information of real users, we investigate how data publicly available on social networks can affect these processes, following the strategy discussed in Section 6.3.

The SODA ADVANCE tool allows the identification of the profiles of users on different social networks, the extraction the public information, and the evaluation of password syntaxes with respect to reconstructed data, by using different evaluation methods, i.e., CUPP, LEET, COVERAGE, FORCE, and CPS. However, these evaluation approaches try to find a match between specific information and the content of passwords, without taking into account the semantic aspect in their evaluations. This aspect can be covered with the capabilities of LLMs, as shown from the results achieved in Section 7.3. To this end, starting from the initial data provided by users involved in our evaluation, we have first reconstructed the public information of each user available on social networks and then combined this information for generating and evaluating passwords.

Table 7 shows the results achieved by the evaluations performed combining SODA ADVANCE and LLMs in the password strength evaluations considering reconstructed data. Similarly to the previous evaluation, we report the average performances for all passwords generated for each user, i.e., 250 passwords. As we can see, the integration of new information reconstructed from social networks and the evaluation of password strength performed with the SODA ADVANCE evaluation module has led to a significant improvement in password evaluation performance in many LLMs, especially for Claude, ChatGPT, and Google Gemini. In particular, Claude, with an accuracy of 0.77 and a precision of 0.89, showed the highest overall performance among the evaluated LLMs. The other models obtain average accuracy values always greater than or equal to 0.56 and precision values ranging from 0.52 to 0.56. This can probably be due to the different capabilities in leveraging the additional information reconstructed from social networks and the evaluation metrics provided by SODA ADVANCE. In fact, while Claude may have been more efficient at incorporating and using the additional information to improve its password rating capabilities, other models may have had difficulty effectively leveraging this information or accurately interpreting the information and metrics provided by SODA ADVANCE.

Let us consider the results of the previous evaluation, when we consider a smaller set of information in the password strength evaluation (Table 6). Although the results are not directly comparable, since the passwords generated in this evaluation consider a larger set of information for each user, we try to analyze the behaviors of LLMs to evaluate password strength when considering different information for the evaluation steps. As we can see, Falcon showed improvements in accuracy and precision when considering additional data provided by SODA ADVANCE. The accuracy of Falcon increased from 0.49 to 0.52, and its precision increased from 0.48 to 0.77. Instead, LLaMa and Google Gemini showed improvements in their precision scores ranging from 20% to 24%, whereas ChatGPT showed the best improvement ranges from 0.18 to 0.30 points by achieving 0.76, 0.88, 0.79, and 0.83 for accuracy, precision, recall, and F1-score, respectively. Concerning Claude, which already had high accuracy and precision in the previous evaluation, it showed significant improvements in precision, recall, and F1-score in the second evaluation, suggesting that it may have been effectively leveraging the available information provided by SODA ADVANCE. A slight improvement has also been achieved by the ensembles of LLMs, both when considering all models and when considering only the best ones. This is probably due to the improvement of the models that most influence the ensemble results and the consequent reduction in incorrectly evaluated passwords.

These improvements can be attributed to the broader data extracted from social networks, which probably have provided additional context and personal information that allowed LLMs to make more accurate assessments of password strength. This has led to an increase in True Positives (TP) and True Negatives (TN), indicating that the models were better able to correctly identify both strong and weak passwords. Additionally, the evaluation metrics provided by SODA ADVANCE, such as CUPP, LEET, FORCE, COVERAGE, and CPS probably contributed to the improved performance by providing a more comprehensive assessment of password strength based on personal information.

Generally speaking, the public availability of personal information across multiple social networks tends to improve the capabilities of most LLMs in the considered scenario. As we have seen, by incorporating data publicly available on social networks and leveraging the evaluation metrics provided by the SODA ADVANCE tool, most LLMs demonstrated improved performance in generating and evaluating password strength compared to evaluations based on a smaller set of personal information. However, this raises new challenges for users. In fact, the problem of losing track of public and privatized information shared on various social networks can represent a threat. As personal information becomes more available across multiple platforms, individuals must be vigilant in managing their online presence and understanding the potential risks associated with sharing sensitive data. The integration of personal information into passwords highlights the importance of implementing robust security measures and privacy settings on social network accounts. Moreover, in this scenario, the growing diffusion of LLMs represents another threat to be considered since, as we have seen, they have the ability to analyze and generate potential passwords based on user data. This could lead to attackers using them as a tool to infer user passwords based on their publicly available information from social networks. Therefore, this highlights the need for stringent data privacy regulations and ethical guidelines in the dissemination of LLMs in order to avoid their misuse.

7.5. RQ4: How effective is the prompt-based methodology for password generation and evaluation compared to state-of-the-art models?

As discussed in the previous section, most of the tools and models available in the state-of-the-art tend to evaluate the strength of passwords from a semantic perspective. This enables them to consider three different levels of strength of the passwords, i.e., weak, strong, and medium.

Medium password strength evaluation. To deeply evaluate the capabilities of LLMs in both generation and evaluation tasks, and the effectiveness of SODA ADVANCE in evaluating passwords, we have investigated their performances also considering medium-level security passwords. To this end, starting from the initial data provided by the 100 users, we have generated a set of 10 passwords for each of the three categories and for each user, by using the prompt engineering approach discussed in Section 6.1.1. Table 8 shows the average $\mathcal{S}$ values on the medium strength passwords generated by LLMs and evaluated through SODA ADVANCE. As we can see, all the $\mathcal{S}$ take values in a range between 0.36 and 0.60. In particular, Claude, Google Gemini, and ChatGPT outperform all other LLMs showing good capabilities to generate medium password strength achieving the highest number of medium passwords. The combination of alphanumeric characters, symbols, and aspects of users' personal data has enabled these LLMs to demonstrate a remarkable ability to generate coherent passwords to the requested security level showing the high understanding capabilities of these models. Instead, Dolly, LLaMa, and Falcon demonstrate fewer capabilities to coherently generate this type of password than the other LLMs, reporting a $\mathcal{S}$ value in a range from 0.36 to 0.45. It is important to note that passwords generated by these models contain in many cases only a concatenation of personal information or recurrent patterns making them useless for the users, such as "*PolitecnicO*", "*MilanoMBA*", or "*NapoliSunset88*".

To assess the evaluation capabilities of LLMs and SODA ADVANCE when also considering medium-strength passwords, we asked each model to evaluate each password following the approach shown in Sections 6.2 and 6.3.

Table 9 shows the average values of the performances achieved by the LLMs according to the approach described in Section 6.2. As we can see, the classification task on multiple labels has significantly reduced the performances of all LLMs. These metrics indicate that these models have correctly classified a small part of passwords and that with a more complex problem, i.e., considering three types of strengths, their performances significantly decrease. From the analysis of the results. we have noticed that most of the passwords correctly evaluated were weak passwords containing recurrent patterns and combinations of user data. Instead, they were not able to discriminate between strong and medium levels of passwords.

Table 10 shows the performances achieved by the LLMs when combined with SODA ADVANCE according to the approach described in Section 6.3. As we can see, the overall performances are higher than those achieved in the previous evaluation (i.e., Table 9), showing that Claude outperforms all the other LLMs. From the analysis of the results, we have noticed that the initial evaluation provided by SODA ADVANCE has effectively supported LLMs in the discrimination of weak, medium, and strong passwords. Nevertheless, the overall performances are lower than those achieved in the evaluations shown in the previous Sections, i.e., when considering only two levels of strength. This is also due to the fact that they wrongly classified most of the strong passwords as medium level of security, and most of the medium passwords as weak or strong.

Table 8
Average S values achieved by LLMs on the medium passwords generated based on the reconstructed data.

UserID	#Password	ChatGPT	Claude	Falcon	Dolly	LLaMa	Gemini	UserID	#Password	ChatGPT	Claude	Falcon	Dolly	LLaMa	Gemini
0	10	± 0.46	± 0.52	± 0.38	± 0.44	± 0.39	± 0.60	50	10	± 0.52	± 0.49	± 0.44	± 0.42	± 0.44	± 0.59
1	10	± 0.56	± 0.47	± 0.45	± 0.42	± 0.45	± 0.53	51	10	± 0.54	± 0.50	± 0.43	± 0.40	± 0.40	± 0.56
2	10	± 0.57	± 0.48	± 0.38	± 0.40	± 0.42	± 0.55	52	10	± 0.57	± 0.47	± 0.45	± 0.45	± 0.44	± 0.55
3	10	± 0.58	± 0.57	± 0.42	± 0.43	± 0.45	± 0.60	53	10	± 0.55	± 0.51	± 0.40	± 0.44	± 0.40	± 0.51
4	10	± 0.52	± 0.55	± 0.45	± 0.39	± 0.43	± 0.48	54	10	± 0.60	± 0.49	± 0.41	± 0.41	± 0.41	± 0.58
5	10	± 0.48	± 0.50	± 0.42	± 0.41	± 0.43	± 0.55	55	10	± 0.52	± 0.52	± 0.44	± 0.38	± 0.39	± 0.60
6	10	± 0.52	± 0.45	± 0.45	± 0.42	± 0.37	± 0.54	56	10	± 0.51	± 0.55	± 0.44	± 0.39	± 0.42	± 0.60
7	10	± 0.52	± 0.58	± 0.42	± 0.45	± 0.42	± 0.55	57	10	± 0.60	± 0.54	± 0.41	± 0.40	± 0.45	± 0.60
8	10	± 0.55	± 0.58	± 0.43	± 0.41	± 0.39	± 0.52	58	10	± 0.52	± 0.49	± 0.41	± 0.42	± 0.44	± 0.58
9	10	± 0.55	± 0.51	± 0.38	± 0.41	± 0.41	± 0.58	59	10	± 0.49	± 0.49	± 0.43	± 0.42	± 0.44	± 0.55
10	10	± 0.55	± 0.49	± 0.39	± 0.42	± 0.37	± 0.58	60	10	± 0.50	± 0.55	± 0.41	± 0.41	± 0.41	± 0.45
11	10	± 0.55	± 0.50	± 0.44	± 0.44	± 0.45	± 0.60	61	10	± 0.50	± 0.49	± 0.41	± 0.42	± 0.42	± 0.57
12	10	± 0.49	± 0.53	± 0.42	± 0.44	± 0.45	± 0.51	62	10	± 0.58	± 0.51	± 0.44	± 0.44	± 0.38	± 0.46
13	10	± 0.50	± 0.50	± 0.41	± 0.42	± 0.45	± 0.56	63	10	± 0.55	± 0.55	± 0.42	± 0.41	± 0.44	± 0.59
14	10	± 0.50	± 0.50	± 0.43	± 0.36	± 0.39	± 0.57	64	10	± 0.59	± 0.46	± 0.42	± 0.41	± 0.45	± 0.57
15	10	± 0.60	± 0.53	± 0.40	± 0.41	± 0.45	± 0.59	65	10	± 0.60	± 0.49	± 0.44	± 0.38	± 0.44	± 0.59
16	10	± 0.56	± 0.47	± 0.43	± 0.42	± 0.43	± 0.54	66	10	± 0.52	± 0.49	± 0.42	± 0.44	± 0.45	± 0.58
17	10	± 0.52	± 0.55	± 0.44	± 0.40	± 0.38	± 0.57	67	10	± 0.47	± 0.50	± 0.43	± 0.38	± 0.45	± 0.58
18	10	± 0.50	± 0.53	± 0.41	± 0.44	± 0.42	± 0.51	68	10	± 0.53	± 0.53	± 0.45	± 0.39	± 0.37	± 0.45
19	10	± 0.49	± 0.55	± 0.42	± 0.38	± 0.42	± 0.60	69	10	± 0.59	± 0.57	± 0.45	± 0.36	± 0.40	± 0.58
20	10	± 0.55	± 0.55	± 0.44	± 0.44	± 0.41	± 0.46	70	10	± 0.49	± 0.52	± 0.38	± 0.42	± 0.45	± 0.59
21	10	± 0.52	± 0.51	± 0.37	± 0.44	± 0.44	± 0.55	71	10	± 0.55	± 0.46	± 0.45	± 0.44	± 0.40	± 0.52
22	10	± 0.51	± 0.50	± 0.42	± 0.42	± 0.42	± 0.58	72	10	± 0.52	± 0.58	± 0.45	± 0.39	± 0.39	± 0.50
23	10	± 0.49	± 0.48	± 0.39	± 0.39	± 0.39	± 0.55	73	10	± 0.48	± 0.46	± 0.42	± 0.43	± 0.40	± 0.54
24	10	± 0.51	± 0.50	± 0.38	± 0.41	± 0.40	± 0.57	74	10	± 0.50	± 0.55	± 0.42	± 0.40	± 0.43	± 0.59
25	10	± 0.59	± 0.53	± 0.43	± 0.37	± 0.40	± 0.58	75	10	± 0.55	± 0.51	± 0.40	± 0.41	± 0.45	± 0.58
26	10	± 0.48	± 0.47	± 0.39	± 0.41	± 0.44	± 0.54	76	10	± 0.52	± 0.54	± 0.45	± 0.43	± 0.39	± 0.56
27	10	± 0.48	± 0.50	± 0.38	± 0.43	± 0.38	± 0.51	77	10	± 0.58	± 0.47	± 0.43	± 0.43	± 0.43	± 0.59
28	10	± 0.45	± 0.55	± 0.39	± 0.42	± 0.43	± 0.45	78	10	± 0.52	± 0.54	± 0.41	± 0.42	± 0.36	± 0.56
29	10	± 0.53	± 0.60	± 0.43	± 0.42	± 0.39	± 0.51	79	10	± 0.53	± 0.49	± 0.38	± 0.45	± 0.45	± 0.54
30	10	± 0.50	± 0.51	± 0.45	± 0.39	± 0.42	± 0.57	80	10	± 0.59	± 0.48	± 0.44	± 0.40	± 0.44	± 0.56
31	10	± 0.50	± 0.49	± 0.42	± 0.43	± 0.38	± 0.57	81	10	± 0.50	± 0.49	± 0.42	± 0.41	± 0.36	± 0.56
32	10	± 0.49	± 0.46	± 0.41	± 0.39	± 0.38	± 0.58	82	10	± 0.54	± 0.57	± 0.45	± 0.44	± 0.43	± 0.48
33	10	± 0.47	± 0.46	± 0.40	± 0.43	± 0.42	± 0.49	83	10	± 0.47	± 0.49	± 0.42	± 0.42	± 0.44	± 0.51
34	10	± 0.57	± 0.48	± 0.41	± 0.45	± 0.43	± 0.48	84	10	± 0.50	± 0.57	± 0.43	± 0.40	± 0.43	± 0.52
35	10	± 0.58	± 0.55	± 0.42	± 0.42	± 0.44	± 0.60	85	10	± 0.60	± 0.54	± 0.44	± 0.41	± 0.45	± 0.49
36	10	± 0.56	± 0.45	± 0.41	± 0.44	± 0.45	± 0.56	86	10	± 0.54	± 0.48	± 0.45	± 0.42	± 0.36	± 0.50
37	10	± 0.55	± 0.51	± 0.45	± 0.42	± 0.45	± 0.54	87	10	± 0.52	± 0.50	± 0.36	± 0.42	± 0.41	± 0.56
38	10	± 0.53	± 0.52	± 0.42	± 0.36	± 0.44	± 0.60	88	10	± 0.52	± 0.47	± 0.38	± 0.39	± 0.41	± 0.53
39	10	± 0.59	± 0.47	± 0.38	± 0.43	± 0.38	± 0.59	89	10	± 0.49	± 0.50	± 0.43	± 0.39	± 0.41	± 0.58
40	10	± 0.48	± 0.55	± 0.44	± 0.44	± 0.43	± 0.53	90	10	± 0.52	± 0.53	± 0.43	± 0.45	± 0.36	± 0.46
41	10	± 0.48	± 0.51	± 0.40	± 0.38	± 0.41	± 0.59	91	10	± 0.52	± 0.53	± 0.41	± 0.42	± 0.44	± 0.54
42	10	± 0.51	± 0.52	± 0.45	± 0.45	± 0.41	± 0.60	92	10	± 0.57	± 0.49	± 0.37	± 0.43	± 0.44	± 0.54
43	10	± 0.50	± 0.54	± 0.36	± 0.40	± 0.42	± 0.60	93	10	± 0.46	± 0.45	± 0.45	± 0.44	± 0.41	± 0.52
44	10	± 0.47	± 0.45	± 0.45	± 0.42	± 0.44	± 0.54	94	10	± 0.51	± 0.55	± 0.43	± 0.42	± 0.36	± 0.54
45	10	± 0.56	± 0.51	± 0.42	± 0.43	± 0.40	± 0.55	95	10	± 0.51	± 0.48	± 0.44	± 0.39	± 0.45	± 0.56
46	10	± 0.59	± 0.52	± 0.41	± 0.41	± 0.40	± 0.59	96	10	± 0.55	± 0.51	± 0.39	± 0.42	± 0.44	± 0.56
47	10	± 0.59	± 0.51	± 0.39	± 0.41	± 0.44	± 0.60	97	10	± 0.46	± 0.48	± 0.39	± 0.42	± 0.43	± 0.55
48	10	± 0.52	± 0.52	± 0.40	± 0.41	± 0.39	± 0.55	98	10	± 0.52	± 0.48	± 0.41	± 0.41	± 0.36	± 0.52
49	10	± 0.54	± 0.51	± 0.43	± 0.43	± 0.43	± 0.53	99	10	± 0.59	± 0.52	± 0.40	± 0.44	± 0.38	± 0.58

Table 9
Results achieved by LLMs in evaluating medium password strength considering data reconstructed by SODA ADVANCE.

Model	Accuracy	Precision	Recall	F1-score
ChatGPT	±0.33	±0.33	±0.33	±0.32
Claude	**±0.34**	**±0.34**	**±0.34**	**±0.34**
Dolly	±0.29	±0.30	±0.29	±0.30
Falcon	±0.30	±0.31	±0.30	±0.31
LLaMa	±0.31	±0.31	±0.31	±0.31
Google Gemini	±0.32	±0.32	±0.32	±0.32
All LLMs	±0.32	±0.17	±0.32	±0.18
Best 3 LLMs	**±0.33**	**±0.21**	**±0.33**	**±0.22**

Table 10
Results achieved by LLMs in evaluating password strength considering data reconstructed by SODA ADVANCE.

Model	Accuracy	Precision	Recall	F1-score
ChatGPT	±0.43	±0.43	±0.44	±0.44
Claude	**±0.57**	**±0.58**	**±0.57**	**±0.58**
Dolly	±0.31	±0.31	±0.31	±0.31
Falcon	±0.32	±0.32	±0.32	±0.32
LLaMa	±0.33	±0.33	±0.32	±0.33
Google Gemini	±0.41	±0.41	±0.42	±0.42
All LLMs	±0.35	±0.29	±0.35	±0.23
Best 3 LLMs	±0.41	±0.36	±0.41	±0.32

Comparative evaluation with state-of-the-art tools. To further investigate the capabilities of SODA ADVANCE with respect to other tools, we perform a comparison with some of the most recent tools for password evaluation available in the state-of-the-art, i.e., Zxcvbn, CKL_PSM, and Semantic PCFG.

The semantic PCFG tool creates a grammar based on the set of passwords provided during the training phase and evaluates passwords

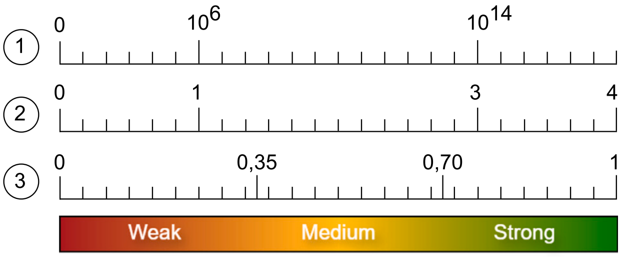


Fig. 7. Range split for each password strength meter.

according to this. This tool is especially useful when more interpretability is needed, as it allows for a more in-depth analysis of a password's structure. The strength of a password is calculated using Monte Carlo strength evaluation [50], which estimates the number of attempts needed to guess the password correctly.

CKL_PSM model has been designed and implemented to mitigate the risks of chunk-level attacks. The method, based on Probabilistic Context-Free Grammar (PCFG) [51], converts passwords into patterns by grouping consecutive characters of the same category. This approach allows to estimate the probability of a password being correctly guessed based on its composition and structure.

Zxcvbn is an open-source password security analyzer library. It adopts a machine learning approach to estimate the strength of passwords and identify potential vulnerabilities. It provides a numerical score accompanied by suggestions for improving password security, taking into account the complexity and predictability of passwords. The score ranges from 0 to 4, where 0 is weak and 4 is strong.

In order to be able to compare the values obtained from the library and tools with those of the evaluation module of SODA ADVANCE, we uniform the ranges to fit the strength of the passwords in three categories, *weak*, *medium*, and *strong*, were defined to evaluate the password. Fig. 7 shows the resulting categorization of values obtained from the password strength meters used, where ① denotes the range split for CKL_PSM and the Semantic PCFG tools; ② denotes the range split for Zxcvbn, and ③ denotes the range split for the SODA ADVANCE tool.

To compare evaluation methods of SODA ADVANCE with the other tools, we use the Evite dataset, which contains a set of leaked passwords with some information related to them, such as e-mails, first and last names, phone numbers, physical addresses, birth dates, and genders. For the purposes of our evaluation, we extracted a random sample of 250 passwords, ranging in length from 8 to 25 characters.

Fig. 8 shows the results of SODA ADVANCE, CKL_PSM, Zxcvbn, and Semantic PCFG on the considered set of passwords. As we can see, most of the passwords have been classified as medium by all tools, and only a few of them as strong. Concerning CKL_PSM, Zxcvbn, and Semantic PCFG these results highlight that they mainly consider syntactical characteristics of passwords, evaluating as medium or strong passwords containing information associated with users, such as those in the format *[last name], [first letter of first name][last name], [last name/first name][year of birth]*, or *[first letters of last name][birthday]*. Conversely, SODA ADVANCE has demonstrated good capabilities of evaluation for the passwords containing these types of information, classifying them as weak. Moreover, SODA ADVANCE classified as medium some passwords consisting of simple dictionary words not semantically linked to users, such as "*Chocolate.1973!*" or "*OfficeUS57*". These types of passwords have been considered strong by the methods that evaluate these attempts, i.e., CKL_PSM, Zxcvbn, and Semantic PCFG, since they have a medium-complex syntax that requires a large number of attempts to crack. This is probably due to the metrics for the analysis of syntaxes included in the CPS. Generally speaking, we have noticed that no model excels at evaluating password strength. As we expected, SODA ADVANCE demonstrated good evaluation capabilities for passwords that contain some user information but overestimates the complexity of passwords when they contain words not semantically linked to the user. On the other hand, tools that evaluate passwords based on crack attempts often underestimate the strength of passwords with complex syntax if they contain information related to the user. However, as also demonstrated for LLMs, considering the problem of evaluating password strength based on semantics with three levels of strength is extremely more difficult and the evaluations are less accurate.

Evaluating passwords with a state-of-the-art model. To further investigate the password-generation capabilities of LLMs, we evaluated the strength of the passwords with PassBERT, which is one of the most recent models in the literature for making focused attacks on passwords. PassBERT uses the fine-tuning paradigm for password-guessing attacks, with a pre-trained password model and different fine-tuning approaches for making focused attacks. Among them, we consider Targeted Password Guessing (TPG) which aims to estimate the number of guesses of cracking the input password given a set of leaked passwords [49].

For the purposes of our evaluation, we consider the 100 users and their 250 strong passwords generated by LLMs adopted in the previous evaluation (see Section 7.2). Moreover, we consider the weak passwords inferred by CUPP as leaked passwords. For each strong password, we evaluate its strength with the PassBERT model and the TPG approach. In particular, for each leaked password, the TPG model makes syntactical transformations on it by keeping, deleting, and/or replacing one or more characters in the passwords. Then, it first computes the minimal edit path, i.e., the shortest amount to change a password into a leaked password, and then evaluates the probability of edit paths by multiplying the edit operations' probabilities [49].

The evaluation considered 250 passwords for each user, with a total of 25.000 strong passwords. The results showed that among the strong passwords, only the passwords of a small set of users were inferred by PassBERT. Specifically, PassBERT was able to identify only 22 passwords out of the 25.000 evaluated, probably due to the complexity of the syntaxes of these passwords. In fact, although the passwords generated by LLMs are based on personal information about the user and are therefore easy to remember, they are also syntactically complex and difficult for models such as TPG to crack. These results, together with those achieved from the previous evaluation, underscore the robustness of using LLMs for generating secure passwords semantically related to the information of the users and highlight the limited effectiveness of an advanced targeted guessing model, i.e., PassBERT.

8. Discussion

This work focused on evaluating and improving the strength of user passwords. However, the development of mechanisms to ensure strong user-chosen passwords is not the only way to approach the problem, as we are detailing next in this Section. Nevertheless, as we will demonstrate, this issue needs to be addressed since alternative authentication mechanisms alone are not sufficient to cover all possible scenarios in a more secure manner.

8.1. Multi-factor authentication

Multi-Factor Authentication (MFA), in particular the 2-Factor Authentication (2FA), is a method to enhance security in electronic authentication that verify authentication by verifying authentication using at least two methods before granting access to the protected resource. The basic idea is that if one method is breached by an attacker, the second method provides additional security.

While MFA certainly improves security, it comes at the cost of increased usability issues from the user's point of view. For this reason, it is usually applied in limited contexts (e.g., for bank accounts,

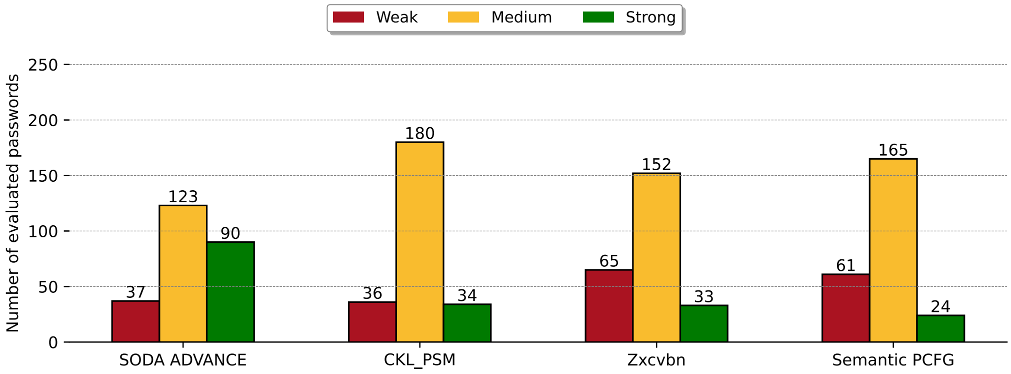


Fig. 8. Result obtained from SODA ADVANCE, CKL_PSM, Zxcvbn, and Semantic PCFG on a sample of *Evite* dataset.

sometimes only when monetary operations are required by the user, or on other websites when significant personal identification editing is needed). Due to the burden of a second authentication, the second mechanism is often quite trivial, relying, for example, on a simple button click (on a different device).

Two important facts must be considered regarding MFA as a potential solution for weak user passwords:

1. MFA still requires authentication mechanisms such as user-defined passwords discussed in our work, so it cannot be considered a true alternative but an additional security mechanism to bolster password authentication;
2. Second factors in 2FA are proven to be vulnerable, as a user can accidentally approve access to a request issued by a hacker without realizing it. Due to implementation or usability issues mentioned earlier, MFA may not be available at all as a mechanism on some platforms.[15]

In conclusion, our work has applications and potential impact in contexts where MFA is not applicable as well as when it is available.

8.2. Server-side limits on the number of login attempts

Users may forget or mistype their passwords; therefore, the server should allow a certain number of wrong attempts to enable them to authenticate as usual. While limiting the number of login attempts undoubtedly reduces a hacker's chances of guessing credentials through brute force, dictionary-based attacks, or hybrid attacks, it is important to understand that strict limitations may also lock out legitimate users.

Blocking the authentication mechanism due to an ongoing attack could therefore result in a Denial of Service Attack (DoS), limiting, slowing down, or completely preventing the user from accessing with the correct password. In some cases, this may be part of a more complex attack involving phishing, increasing the chances that the user would believe on phishing emails since the authentication mechanism in the original site was actually malfunctioning. Therefore, similar to 2FA, we should consider server-side protections as an additional security mechanism that alone cannot solve the problem of weak user passwords. We believe, therefore, that the outcomes of our work will also apply in contexts where server-side detection of attacks is in place.

9. Conclusion and future directions

In this paper, we have investigated the threats related to the definition of password when users publicly share their data on social network platforms. To this end, we have first proposed a new data reconstruction tool, namely SODA ADVANCE, capable of identifying users' social profiles starting from a few pieces of information, reconstructing their public data, and evaluating a password according to these. The evaluation module of SODA ADVANCE combines some of the well-known methodologies proposed in the literature for evaluating password strength, i.e., CUPP, LEET, COVERAGE, and FORCE, and integrates a new metric, namely CPS, that combines each of these values in order to provide a cumulative value that describes the strength of a password. Moreover, we have designed three different pipelines aiming to evaluate the performance of emerging generative LLMs, i.e., ChatGPT, Claude, Dolly, Falcon, LLaMa, and Google Gemini, in the generation of strong passwords and the evaluation of their strength. To interact with LLMs, we designed new ad-hoc prompting functions based on automatic and manual prompt engineering approaches, which allowed us to generate and evaluate passwords using both generic users' data and the public information reconstructed by SODA ADVANCE. The experimental evaluations with real users have shown that LLMs revealed good capabilities in generating strong passwords and evaluating password strength based on user data. Among the LLMs, Claude has proven to be the model with the greatest ability to provide meaningful and detailed analysis of password strength, as well as generate more secure passwords based on the reconstructed data. Moreover, the combination of LLMs with the SODA ADVANCE tool has led to significant improvements in the password evaluation process with LLMs. To further investigate the effectiveness of LLMs and SODA ADVANCE in the generation and evaluation of passwords we compared it with recent state-of-the-art approaches for both tasks. Finally, a specific evaluation including an intermediate strength level for passwords has been performed. Results highlight that although, in general, LLMs do not perform well in the generation of medium-level passwords, the SODA ADVANCE tool allows the effectiveness of the approach in the evaluation task not to be lowered by too much. Instead, concerning the comparison with the state-of-the-art approaches, the evaluation methods included in SODA ADVANCE perform better in this task. In fact, although the compared approaches obtained good performances in the evaluation of passwords containing common words, they failed to classify well more complex passwords that are semantically related to the user's information. Finally, it has been shown that only a very small percentage of strong passwords generated by LLMs succeed in being leaked by PassBERT's TPG model.

The methodologies and results obtained in this study open research in several new directions. Firstly, the study revealed different threats to users stemming from the public sharing of personal data on social network platforms, particularly regarding the security of their passwords. Future research could delve deeper into understanding and

[15] For further information of issues related to 2FA, see for example *5 Myths of Two-Factor Authentication* at www.wired.com/insights/2013/04/five-myths-of-two-factor-authentication-and-the-reality/.

mitigating these threats, including exploring alternative approaches to password management and authentication in the context of widespread public data availability. Moreover, further investigation could focus on enhancing the capabilities of the data reconstruction tool to extract a large set of public information from other web platforms. Furthermore, password strength assessment can be explored further using LLM by investigating the effectiveness of models trained specifically for this problem. Finally, emerging trends related to LLMs require further investigation for a better understanding of how these models treat personal information and whether they comply with European and global regulations.

CRediT authorship contribution statement

Maurizio Atzori: Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **Eleonora Calò:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Conceptualization. **Loredana Caruccio:** Writing – review & editing, Supervision, Methodology, Formal analysis, Conceptualization. **Stefano Cirillo:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Methodology, Data curation, Conceptualization. **Giuseppe Polese:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Giandomenico Solimando:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Data curation, Conceptualization.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work was partially supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

References

[1] F. Cerruto, S. Cirillo, D. Desiato, S. Gambardella, G. Polese, Social network data analysis to highlight privacy threats in sharing data, J. Big Data (2022).

[2] S. Cirillo, D. Desiato, M. Scalera, G. Solimando, A visual privacy tool to help users in preserving social network data, in: Proceedings of the Workshops, Work in Progress Demos and Doctoral Consortium at the IS-EUD 2023 Co-Located with the 9th International Symposium on End-User Development (IS-EUD 2023), Cagliari, Italy, June 6-8, 2023, in: CEUR Workshop Proceedings, vol. 3408, 2023, pp. 1–8.

[3] M. Teresa Baldassarre, V. Santa Barletta, D. Caivano, A. Piccinno, Integrating security and privacy in HCD-scrum, in: Proceedings of the CHItaly 2021: 14th Biannual Conference of the Italian SIGCHI Chapter, 2021, pp. 1–5.

[4] L. Caruccio, D. Desiato, G. Polese, Fake account identification in social networks, in: Proceedings of the IEEE International Conference on Big Data (Big Data) 2018, IEEE, 2018, pp. 5078–5085.

[5] J.N. Paredes, J.C.L. Teze, M.V. Martinez, G.I. Simari, The HEIC application framework for implementing XAI-based socio-technical systems, Online Soc. Netw. Media 32 (2022) 100239.

[6] J.H. Abawajy, M.I.H. Ninggal, T. Herawan, Privacy preserving social network data publication, IEEE Commun. Surv. Tutor. 18 (2016) 1974–1997.

[7] L. Luceri, D. Andreoletti, M. Tornatore, T. Braun, S. Giordano, Measurement and control of geo-location privacy on Twitter, Online Soc. Netw. Media 17 (2020) 100078.

[8] V.S. Barletta, G. Desolda, D. Gigante, R. Lanzilotti, M. Saltarella, From GDPR to privacy design patterns: The MATERIALIST framework, in: S.D.C. di Vimercati, P. Samarati (Eds.), Proceedings of the 19th International Conference on Security and Cryptography, SECRYPT 2022, Lisbon, Portugal, July 11-13, 2022, SCITEPRESS, 2022, pp. 642–648.

[9] M. Tulek, M. Kuskon, I. Sezgin, A. Levi, Disclosure of personal information in passwords on social media, in: Proceedings of the 2020 28th Signal Processing and Communications Applications Conference, SIU, 2020, pp. 1–4.

[10] L. Bošnjak, J. Sreš, B. Brumen, Brute-force and dictionary attack on hashed real-world passwords, in: Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) 2018, 2018, pp. 1161–1166.

[11] I. Kayes, A. Iamnitchi, Privacy and security in online social networks: A survey, Online Soc. Netw. Media 3–4 (2017) 1–21.

[12] N. Voloch, N. Gal-Oz, E. Gudes, A trust based privacy providing model for online social networks, Online Soc. Netw. Media 24 (2021) 100138.

[13] D. Antypas, A. Preece, J. Camacho-Collados, Negativity spreads faster: A large-scale multilingual twitter analysis on the role of sentiment in political communication, Online Soc. Netw. Media 33 (2023) 100242.

[14] J. Rando, F. Perez-Cruz, B. Hitaj, Passgpt: Password modeling and (guided) generation with large language models, 2023, arXiv preprint arXiv:2306.01545.

[15] S. Egelman, A. Sotirakopoulos, I. Muslukhov, K. Beznosov, C. Herley, Does my password go up to eleven? The impact of password meters on password selection, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13, Association for Computing Machinery, 2013, pp. 2379–2388.

[16] C. Kuo, S. Romanosky, L.F. Cranor, Human selection of mnemonic phrase-based passwords, in: Proceedings of the Second Symposium on Usable Privacy and Security, SOUPS '06, Association for Computing Machinery, 2006, pp. 67–78.

[17] X. Cui, C. Li, Y. Qin, Y. Ding, A password strength evaluation algorithm based on sensitive personal information, in: Proceedings of the IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) 2020, IEEE, 2020, pp. 1542–1545.

[18] Y. Li, H. Wang, K. Sun, Personal information in passwords and its security implications, IEEE Trans. Inf. Forensics Secur. 12 (10) (2017) 2320–2333.

[19] P. Jourdan, E. Stavrou, Towards designing advanced password cracking toolkits: optimizing the password cracking process, in: Proceedings of the Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization, 2019, pp. 203–208.

[20] B. Ur, F. Alfieri, M. Aung, L. Bauer, N. Christin, J. Colnago, L.F. Cranor, H. Dixon, P. Emami Naeini, H. Habib, N. Johnson, W. Melicher, Design and evaluation of a data-driven password meter, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17, Association for Computing Machinery, 2017, pp. 3775–3786.

[21] T. Zhang, Z. Cheng, Y. Qin, Q. Li, L. Shi, Deep learning for password guessing and password strength evaluation, a survey, in: Proceedings of the IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) 2020, IEEE, 2020, pp. 1162–1166.

[22] A. Nosenko, Y. Cheng, H. Chen, Password and passphrase guessing with recurrent neural networks, Inf. Syst. Front. 25 (2) (2023) 549–565.

[23] S. Aboukadri, A. Ouaddah, A. Mezrioui, Machine learning in identity and access management systems: Survey and deep dive, Comput. Secur. (2024) 103729.

[24] G. Deng, X. Yu, H. Guo, Efficient password guessing based on a password segmentation approach, in: Proceedings of the IEEE Global Communications Conference (GLOBECOM) 2019, 2019, pp. 1–6.

[25] B. Hitaj, P. Gasti, G. Ateniese, F. Perez-Cruz, Passgan: A deep learning approach for password guessing, in: Proceedings of the Applied Cryptography and Network Security: 17th International Conference, ACNS 2019, Bogota, Colombia, June 5–7, 2019, Proceedings 17, Springer, 2019, pp. 217–237.

[26] D. Pasquini, A. Gangwal, G. Ateniese, M. Bernaschi, M. Conti, Improving password guessing via representation learning, 2020, arXiv:1910.04232.

[27] D. Temoshok, J. Fenton, Y.-Y. Choong, N. Lefkovitz, A. Regenscheid, J. Richer, Digital Identity Guidelines: Authentication and Lifecycle Management, Tech. Rep., National Institute of Standards and Technology, 2022.

[28] C. Castelluccia, A. Chaabane, M. Dürmuth, D. Perito, When privacy meets security: Leveraging personal information for password cracking, 2013, arXiv:1304.6584.

[29] S. Mishra, Information extraction from digital social trace data with applications to social media and scholarly communication data, SIGWEB Newsl. 2021 (Spring) (2021).

[30] O. Alonso, T. Sellam, Quantitative information extraction from social data, in: Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018, SIGIR '18, Association for Computing Machinery, 2018, pp. 1005–1008.

[31] M.D. Malkauthekar, Analysis of euclidean distance and manhattan distance measure in face recognition, in: Proceedings of the Third International Conference on Computational Intelligence and Information Technology, CIIT 2013, 2013, pp. 503–507.

[32] Mebus, Common user password profiler, 2019, URL https://github.com/Mebus/cupp. (Accessed 20 March 2019).

[33] W. Li, J. Zeng, Leet usage and its effect on password security, IEEE Trans. Inf. Forensics Secur. 16 (2021) 2130–2143.

[34] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, J. Launay, The RefinedWeb dataset for falcon LLM: outperforming curated corpora with web data, and web data only, 2023, arXiv preprint arXiv:2306.01116.

[35] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, ACM Comput. Surv. 55 (9) (2023) 1–35.

[36] L. Caruccio, S. Cirillo, G. Polese, G. Solimando, S. Sundaramurthy, G. Tortora, Can ChatGPT provide intelligent diagnoses? A comparative study between predictive models and ChatGPT to define a new medical diagnostic bot, Expert Syst. Appl. 235 (2024) 121186.

[37] Z.J. Wang, A. Chakravarthy, D. Munechika, D.H. Chau, Wordflow: Social prompt engineering for large language models, 2024, arXiv:2401.14447.

[38] D. Biesner, K. Cvejoski, R. Sifa, Combining variational autoencoders and transformer language models for improved password generation, in: Proceedings of the 17th International Conference on Availability, Reliability and Security, ARES '22, Association for Computing Machinery, 2022, pp. 1–6.

[39] T. Gao, A. Fisch, D. Chen, Making pre-trained language models better few-shot learners, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, 2021, pp. 3816–3830.

[40] B. Ur, F. Noma, J. Bees, S.M. Segreti, R. Shay, L. Bauer, N. Christin, L.F. Cranor, "I added '!' at the end to make it secure": Observing password creation in the lab, in: Proceedings of the Eleventh Symposium on Usable Privacy and Security, SOUPS 2015, 2015, pp. 123–140.

[41] M.L. Mazurek, S. Komanduri, T. Vidas, L. Bauer, N. Christin, L.F. Cranor, P.G. Kelley, R. Shay, B. Ur, Measuring password guessability for an entire university, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, 2013, pp. 173–186.

[42] R. Shay, S. Komanduri, P.G. Kelley, P.G. Leon, M.L. Mazurek, L. Bauer, N. Christin, L.F. Cranor, Encountering stronger password requirements: User attitudes and behaviors, in: Proceedings of the Sixth Symposium on Usable Privacy and Security, Association for Computing Machinery, 2010, pp. 1–20.

[43] L. Caruccio, S. Cirillo, G. Polese, G. Solimando, S. Sundaramurthy, G. Tortora, Claude 2.0 large language model: tackling a real-world classification problem with a new iterative prompt engineering approach, Intell. Syst. Appl. (2024) 200336.

[44] Parlamento europeo e del Consiglio, Regolamento (UE) 2016/679 relativo alla protezione delle persone fisiche con riguardo al trattamento dei dati personali, 2016.

[45] D.L. Wheeler, Zxcvbn: Low-budget password strength estimation, in: Proceedings of the 25th USENIX Security Symposium, USENIX Security 16, USENIX Association, Austin, TX, 2016, pp. 157–173.

[46] T. Ambler, N. Cloud, Browserify, in: Proceedings of the JavaScript Frameworks for Modern Web Dev, A Press, Berkeley, CA, 2015, pp. 101–120, Ch. 1.

[47] M. Xu, C. Wang, J. Yu, J. Zhang, K. Zhang, W. Han, Chunk-level password guessing: Towards modeling refined password composition representations, in: Y. Kim, J. Kim, G. Vigna, E. Shi (Eds.), Proceedings of the CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021, ACM, 2021, pp. 5–20, http://dx.doi.org/10.1145/3460120.3484743.

[48] R. Veras, C. Collins, J. Thorpe, A large-scale analysis of the semantic password model and linguistic patterns in passwords, ACM Trans. Priv. Secur. 24 (2021) 1–21, http://dx.doi.org/10.1145/3448608.

[49] M. Xu, J. Yu, X. Zhang, C. Wang, S. Zhang, H. Wu, W. Han, Improving real-world password guessing attacks via bi-directional transformers, in: Proceedings of the 32nd USENIX Security Symposium, USENIX Security 23, USENIX Association, Anaheim, CA, 2023, pp. 1001–1018.

[50] M. Dell'Amico, M. Filippone, Monte Carlo strength evaluation: Fast and reliable password checking, in: I. Ray, N. Li, C. Kruegel (Eds.), Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015, ACM, 2015, pp. 158–169, http://dx.doi.org/10.1145/2810103.2813631.

[51] M. Weir, S. Aggarwal, B. de Medeiros, B. Glodek, Password cracking using probabilistic context-free grammars, in: Proceedings of the 30th IEEE Symposium on Security and Privacy (SP 2009), 17-20 May 2009, Oakland, California, USA, IEEE Computer Society, 2009, pp. 391–405, http://dx.doi.org/10.1109/SP.2009.8.