
IMPROVING USERS' PASSWORDS WITH DPAR: A DATA-DRIVEN PASSWORD RECOMMENDATION SYSTEM

Assaf Morag

Department of Industrial Engineering
Tel Aviv University
assafmorag@post.tau.ac.il

Liron David

Department of Computer Science
Weizmann Institute of Science
liron.david@weizmann.ac.il

Eran Toch

Department of Industrial Engineering
Tel Aviv University
erant@tauex.tau.ac.il

Avishai Wool

School of Electrical Engineering
Tel Aviv University
yash@eng.tau.ac.il

ABSTRACT

Passwords are the primary authentication method online, but even with password policies and meters, users still find it hard to create strong and memorable passwords. In this paper, we propose DPAR: a Data-driven PAssword Recommendation system based on a dataset of 905 million leaked passwords. DPAR generates password recommendations by analyzing the user's given password and suggesting specific tweaks that would make it stronger while still keeping it memorable and similar to the original password. We conducted two studies to evaluate our approach: verifying the memorability of generated passwords ($n = 317$), and evaluating the strength and recall of DPAR recommendations against password meters ($n = 441$). In a randomized experiment, we show that DPAR increased password strength by 34.8 bits on average and did not significantly affect the ability to recall their password. Furthermore, 36.6% of users accepted DPAR's recommendations verbatim. We discuss our findings and their implications for enhancing password management with recommendation systems.

1 Introduction

Passwords are currently the most widely used method for authentication to digital services [22]. Despite their many limitations, They are extensively used for protecting sensitive information across numerous services, including web services, online banking, social media, and organizational IT access. Creating robust passwords is essential to the safeguarding of these services. According to industry studies, nearly half of data breaches involved authentication failures [65, 33]. Passwords remain vulnerable because people tend to choose easy-to-remember passwords, which are also commonly used [54], and tend to reuse the same password in different services [70]. Adversaries then exploit these patterns to craft attacks on systems and individual accounts [55, 15, 8, 64]. While the use of password managers is increasing, users eschew generated passwords because system-generated passwords are challenging to enter and remember when the manager is unavailable [46, 42]. Due to their flexibility, simplicity, and wide deployment, passwords are still the preferred method of authentication by users [81]. Therefore, this study is motivated by the need to help users create stronger passwords, which remains a constant concern for users, organizations, and policy-makers.

Organizations use two primary types of interventions to encourage stronger passwords during password creation: password composition policies and password meters [39]. Password composition policies generally rely on blocklists and heuristics and are often found to be less apparent to users [34]. Moreover, they do not necessarily produce stronger passwords [38, 62, 39] or memorable ones [38]. Password meters have more success in encouraging users to create stronger passwords [61, 17, 59, 23]. However, the effect of password meters on users' final passwords was shown to be weak [13]. Furthermore, service providers often misuse password meters, implementing them in a way that weakens the final password [39]. One possible explanation for password-meter limitations is that they fall short in nudging

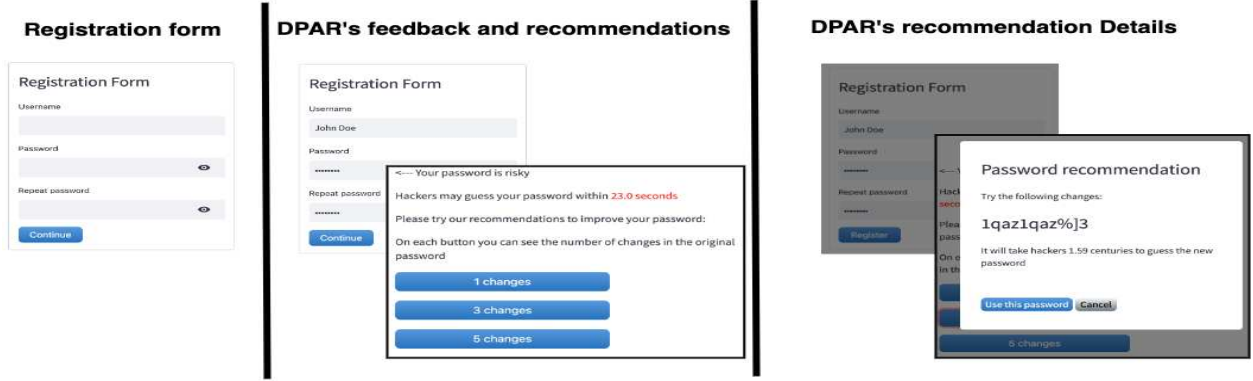


Figure 1: A step-by-step flow of the DPAR user experience, assuming the user initially chose the password “1qaz1qaz”

users to change their passwords to stronger ones [52, 35]. More interventional approaches, such as providing guidance and detailed explanations to users [21, 59], had accumulative positive effects on password strengths. These results point to the potential of more concrete and detailed approaches to nudging users in password creation processes, an approach which we take here a step further in generating suggestions to users based on their given password. These processes can be useful not only when asking users to enter new passwords but also in systems that require users to change their password every few weeks or months, a process that users find exceptionally cumbersome [80].

In this paper, we introduce **DPAR – a Data-driven Password Recommendation system**. DPAR takes as input the user’s given password and aims to improve it by suggesting specific tweaks to the password that make it more robust but still memorable to the user¹. To make the recommendations easy to recall, DPAR’s recommendations are based on the user’s actual given password, in contrast to suggesting completely new random passwords [45]. To do so, DPAR uses a dataset of 905 million leaked passwords [12], and suggests modified passwords that maximize the similarity to the given password while considerably improving the password strength. The DPAR user experience is embedded in the password creation process, providing the user with a small set of optimized suggestions and a traditional password meter that reflects the password strength to the user (see Figure 1).

We have conducted two studies to evaluate the DPAR recommendation system: (1) A study ($n = 317$) designed to measure the correlation between password similarity and perceived usability. our objective was to ascertain whether participants also perceive that a password with fewer modifications is easier to remember. (2) Our main online study ($n = 441$) assessed how our recommendations impacted the security and recall of passwords created by participants, and compared it against the state-of-the-art password meter approach. We show that DPAR’s recommendations led users to create more secure passwords than just providing feedback on their initial password choice. However, this extra strength did not significantly impact recall metrics: Participants could recall the improved password and reported that the recommendation system was easy to use and helped them create stronger passwords. The key contributions of this work are:

- The design of a password recommendation algorithm that uses a dual analysis of password strength and similarity. This algorithm is intended to synthesize assessments of both security and memorability to proffer optimized password proposals.
- We introduce and evaluate a configurable user interaction for multiple password recommendations based on the user’s given password and show it can significantly improve the strength of users’ passwords with comparable recall.
- We show that users perceive password suggestions most similar to their original password as more usable and thus consistently choose it over other alternatives.
- We propose an algorithm for generating password recommendations based on a given password, which maximizes both the similarity of the password and its strength.

¹Our source code is publicly available at <https://github.com/iWitLab/DPAR/tree/main>.

2 Background

2.1 Password policies

Typical password policies are designed to make users’ passwords more difficult to predict by imposing strict guidelines on the password content. Policies appear in many incarnations: some sites require digits, others require the presence of at least three character classes, some banish certain symbols, and most set varying length minimums and maximums [5, 67, 19]. A common approach is the LUDS approach, which stands for *Lowercase Uppercase Digits Special Characters*, detailing how many of these classes should appear in the password [72].

Password policies often pose cognitive burdens on users, which make it harder to create and memorize a password [34]. The tension between password security and usability creates competing challenges that influence how passwords are created and used [69]. Therefore, users tend to employ simple heuristics to cope with the demands of the policy, such as adding special characters at the end of the password [62]. These heuristics might paradoxically result in weaker passwords because of their regularity [38, 26], reducing the barrier for cyber attacks.

2.2 Nudging users

Nudging approaches do not explicitly impose strict policies on users but rather reflect the concept of nudging in behavioral economics to help users make better decisions when creating passwords. Renaud and Zimmerman [52] found that a simple textual feedback nudge did not significantly impact password strength. However, adding an incentive in the form of a more extended expiration date for a stronger password resulted in more robust passwords. Kankane et al. [35] found that the warning messages, which point to the ease with which an adversary can compromise the password, were the most effective ways to encourage the creation of stronger passwords.

Not all measures will work equally well on all users, and giving users several password creation methods might result in more robust and more memorable passwords [79]. Detecting behavioral patterns, such as password reuse [1], might require special equipment and infrastructure. Moreover, not every nudging method may result in stronger passwords. When users were asked to change computer-generated passwords, modifying more characters improved memorability but reduced the strength of passwords, as modifications made them easier to crack [32].

Password meters are a common form of nudging in which the users are presented with information about the password’s strength [82]. Password meters often extend the LUDS approach by including dictionaries, considering 133t speak transformations, keyboard walks, and other heuristics [72, 25]. Ur et al. developed a data-driven password meter that measures strength and detailed feedback [59]. The meter uses neural network models to calculate the strength of a password and 21 heuristics to generate feedback. While the strength estimation relies on neural network models, which have shown strong estimation capabilities, the feedback and suggestions are ultimately generated by heuristics, which suffer from the same drawbacks as password composition policies.

2.3 Measuring passwords

Data-driven approaches to measure password strength aim to simulate how easy it is to crack it, based on heuristics [24, 28, 47] or Markov Models [7, 16, 71]. Neural networks provide a robust way to measure password strength, taking into account the large datasets of leaked passwords that adversaries can easily access [44, 12]. Neural networks are trained to generate the next character of a password given the preceding characters of a password. The Password Guessability Service (PGS) [63] provides easy access to the Neural method and some of its predecessors.

The PESrank model [12] is based on the observation that when people create a password, they often modify a base word by applying various transformations such as adding a prefix and/or a suffix of digits and symbols before or after the base word, capitalizing some letters of the base word, or substituting letters with visually similar digits or symbols using “133t” translations. PESrank models the probability distributions of these classes (dimensions) of transformations based on a massive dataset of 905 million leaked passwords.

While analyzing the strength of a password is a relatively straightforward task, given a threat model, understanding what makes a password usable requires understanding how users use passwords in the real world. Passwords are not created in isolation, and users take into account their ability to remember and manage multiple passwords for multiple services [75]. Password managers somewhat reduce the cognitive burden of remembering hundreds of passwords, but they do not remove the burden altogether: users still want a memorable password that will work even if they do not have access to their password manager [48].

Memorability is, therefore, considered the most crucial aspect of password usability. How can we measure memorability? That depends on the usage scenario. Mukherjee et al. [45] developed a model for generating memorable

passwords from scratch based on passphrases [76]. Levenshtein distance, also known as edit distance, was used in several password-related scenarios, including measuring the impact of privacy policies and meters on modified passwords [66], determining the similarity between a given password and a dictionary word [31], or measuring the similarity between a correct and incorrect password during a recall exercise [40]. However, it is yet unclear whether the Levenshtein distance could serve as a viable approach for generating a usable suggestion based on the user’s original password.

2.4 Password creation systems

In reviewing the existing literature, we identified two common strategies for password creation mechanisms. The first involves system-generated passwords, while the second accepts an initial user-provided password and strengthens it. In Table 3 in the Appendix we summarize the similarities and differences between DPAR and the aforementioned approaches.

System-generated passwords suggest completely new passwords to users and are used in password managers and browsers. For example, Yang et al. [77] conducted an evaluation of various models that generate passwords for users, specifically focusing on those that employ mnemonic strategies. The study illuminates the advantages of mnemonic-based password generation methods while also providing a comparative analysis of these strategies in terms of their effectiveness and the ease with which passwords can be remembered. Huh et al. [32] developed the “Surpass” system, which generates computer-based passwords while permitting users to modify certain characters to enhance memorability. However, their laboratory studies revealed a trade-off: an increase in character modifications improved password recall but simultaneously decreased security levels, as these passwords were more susceptible to cracking. Mukherjee et al. [45] introduced the “MASCARA” model, aimed at enhancing the recall rate of system-generated passphrases. By assessing the character error rate (CER) of an extensive passphrase dataset, the authors identified three statistically significant signals correlated with memorability. This enabled them to compute the recall rate of passphrases and modify them for improved memorability, albeit with a potential reduction in security.

Forget et al. [20] introduced the concept of Persuasive Text Passwords (PTP) as a way to encourage users to create passwords that are both more secure and usable. This system functions by accepting a password input from the user, and then randomly injecting additional characters into the original password to enhance its strength. Users may select the modified password or to further inject random characters into their original choice. The system was evaluated only in a small-scale lab experiment. Ur et al. [59] developed a data-driven password meter (hereafter referred to as the “CUPS password meter”) that uses a neural network model to assess password strength. It not only provides textual feedback but also adjusts the original password by offering suggestions for enhancement. These suggestions and feedback are derived from password policies. This approach bears the closest resemblance to DPAR.

Breaking up with the interaction model of password meters, DPAR gives users several password suggestions, mimicking the interaction of classic recommendation systems [27]. This seemingly subtle change in the user interface leads to a considerable shift in how the system operates, as it requires implementing a new method to choose the best passwords that will be presented to the user based on their existing ones. This selection method demands we establish and apply specific criteria to optimize how passwords are picked. In turn, it led to research questions about how the requirements are set and required to rigorously assess the effectiveness of password strength and recall, which were not covered in the existing literature.

3 DPAR Design and Implementation

3.1 Password strength modeling

In the design of DPAR, we have employed the PESrank model [12] to analyze each password component’s strength and the complete password’s strength based on a-priori analysis of 905 million leaked passwords. As part of the password strength estimation process, a given password is divided into five components (dimensions). If the password begins or ends with a sequence of symbols and digits, the beginning sequence is separated from the password and identified as the prefix. Similarly, the ending digit/symbol sequence is identified as the suffix. The remaining middle part is the mutated base word, which allows for two mutations: specific letters can transform symbols or digits that resemble them (l33t transformations), and lowercase letters can be capitalized. The capitalization pattern and combination of l33t transformations are dimensions 3 and 4. PESrank’s base word (after converting to lower case and undoing l33t transformations) is the 5th dimension. For instance, the password “!1P@ssw0rD2#” has a prefix (“!1”), a suffix (“2#”), a base word (“password”), a l33t list [“@”, “0”], and a capitalization pattern [0, -1].

Empirical data shows that this 5-dimension decomposition is very well aligned with statistics observed in leaked password corpora. According to an analysis of large leaked password corpora [12], 50.24% of leaked passwords end

with digits/symbols and 8.95% start with digits/symbols—i.e., have a non-empty suffix or prefix. Moreover, 7.66% use capital letters and 9.86% use l33t transformations. Finally, while the PESrank model has special case handling for all-numeric or all-symbol passwords (like “123456”, the #1 most popular password), these cases cannot occur in DPAR: our password policy requires having at least one letter in the original password.

In the PESrank model, each dimension has its a-priori probability distribution. The global password probability is calculated as a product of the per-dimension probabilities, converted into its rank in the list of possible passwords in decreasing order of likelihood [11]. The rank of the password is its strength since it directly correlates to the time it would take an attacker to crack the password by testing passwords one by one according to the order. PESrank returns $\log_2(rank)$ which measures the password strength in *bits*.

Our primary goal was to enhance the strength of any given password by following the intuitive human manipulations: improving one or more of the non-base dimensions, including prefix, suffix, l33t transformations, and capitalization. The PESrank model provides both strength estimation to the entire password and specific strength estimation to each dimension. This can be leveraged to both explain the password’s weaknesses and pinpoint modified dimensions of the password to increase its entire strength.

3.2 Password Enhancement

The DPAR password enhancement process starts with a user-selected password. The user’s input is divided into five dimensions, as explained above. We treat each dimension specifically. We create a list of per-dimension random candidate mutations and then assemble all their possible combinations as a list of password candidates. The base word dimension is left unchanged by our recommendation system.

A password prefix (or suffix) can be any string composed of 10 digits and 32 keyboard symbols. Our function (GeneratePrefixSuffix in Algorithm 1 in the Appendix) always returns three choices for a modified prefix/suffix. It distinguishes between an empty prefix/suffix: the user did not use digits or symbols in the password’s beginning (or the end), and a non-empty one. If the original prefix (or suffix) is empty GeneratePrefixSuffix produces three random digit/symbol strings of lengths 1,2,3, respectively. If the original prefix/suffix *str* is not empty, we distinguish between the following scenarios:

1. If *str* is one character long, we randomly replace the character as the length-1 string and randomly append 1 or 2 more digits/symbols to create the length-2 and length-3 strings.
2. If *str* is two characters long, we randomly replace one character or the other character, and we randomly add one digit or symbol, producing 2 strings of length 2 and one of length 3.
3. If *str* is three or more characters long, we randomly select and replace a character three times, producing three modifications and a string of length 3 or more.

Ultimately, we end up with three sets of prefixes/suffixes that were either replaced or created. We repeat this step 4 times and end up with an array of 12 modified prefixes/suffixes. Note that we only insert between 1-3 changes each time. We do this to control the number of tweaks. Lastly, we add the original prefix (or suffix) to the list of recommendations for a total of 13 options for each dimension.

L33t transformations (replacing letters by digits and symbols that are visually similar) were observed by [72] and shown to be commonly used by [12]. We used the 14 l33t transformations from Table 2 in [12]. The l33t rank improvement function (GenerateL33t in Algorithm 1) receives the base word and returns one random possible l33t transformation, i.e., when one of a list of 11 lowercase letters [a, e, i, o, s, x, o, p, z, t, g] is located in the base word (these are letters commonly replaced via l33t). Note that some letters have more than one l33t substitution (e.g., ‘a’ can be transformed into ‘@’ or ‘4’), so there are 14 options. The returned l33t transformation is added to the original l33t transformation for a total of up to 2 options. If the base word doesn’t contain any of the letters mentioned above the l33t transformation dimension is left empty. Note that this transformation only replaces letters with digits or symbols, and not the reverse.

The capitalization rank improvement function receives the base word and returns one random capitalization out of all the possible lower-to-upper-case transformations. The returned capitalization is added to the original capitalization transformation (which may be empty) for a total of up to 2 options.

Recommendation Generation Process In Algorithm 1, we describe the steps taken to create recommendations per each of the non-base dimensions. As described above, each dimensional modification function is activated to create a randomized set of non-base word dimensions. Next, a power set function creates all possible combinations, which are all variations of the original password: up to $13 * 13 * 2 * 2 = 676$ combinations.

This algorithm produces a dataset of 676 recommendations, where each recommendation consists of the user’s base password and at least one modification made by DPAR. Some recommendations in the dataset are modified only in one dimension (prefix, suffix, l33t, capitalization), while others are modified in multiple positions in multiple dimensions. For instance, if the base password of the user was “amsterdam”, a valid DPAR recommendation could be “am5terDam&#”, in which the user’s original choice is modified in 4 positions (“*5*D*#”) spanning 3 dimensions (l33t, capitalization, and suffix).

We calculate the Edit (Levenshtein) distance [41] to estimate password similarity between the original password and a given recommendation. The returned value is the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a character. The Levenshtein distance can be, at most, the sum of the lengths of the original and suggested password.

For each recommendation, DPAR calculates the password strength and Levenshtein distance. We end up with a list of candidate passwords annotated by their password strength (denoted as PS) and Levenshtein Distance (denoted as LD). We exclude from the list of candidates any passwords whose strength is less than that of the original password.

We sort all passwords based on their Levenshtein Distance, with ties broken arbitrarily. For each LD value, we retain the password with the highest strength in $Cand[LD]$ and discard the rest. DPAR displays a choice of 3 password replacement recommendations (recall Figure 1), which are selected by pressing a button. The upper button leads to a random selection between $Cand[1]$ and $Cand[2]$. The middle button leads to a random selection between $Cand[3]$ and $Cand[4]$, and the bottom button leads to a random selection between $Cand[5]$ and $Cand[6]$.

3.3 User Experience

As illustrated in Figure 1, our user interface consists of three screens: the initial landing screen, the feedback and recommendation selection screen, and the specific recommendation screen.

The initial landing screen is divided into two sections. Our system instructs users to create a password with at least eight characters, one letter, and one digit, mimicking a basic password complexity policy. After the user has entered the password, DPAR presents textual feedback based on the password strength, and optionally also three buttons representing our recommendations. The feedback and its text color are determined by the password strength (PS) calculated by the PESrank model. Passwords with $0 < PS \leq 29$ bits receive the textual feedback “weak” colored in red, those with $29 < PS \leq 37$ bits receive the textual feedback “fair” colored in yellow, and those $PS > 37$ bits receive the textual feedback “strong” colored in green. For instance, “Your password is weak”, when the word weak is colored in red, as depicted in Figure 1.

We also estimated how long it would take an attacker to guess the password and placed the color-coded estimate in the feedback. The color coding of the time estimates is aligned with the “weak/fair/strong” categorization described above; namely, if the password is found to be “weak”, the user interface displays the text “Hackers may guess your password within 4 minutes”, where “4 minutes” is colored in red. We estimated the time (denoted as T) it will take a hacker to guess the password by $T = \frac{2^{PS}}{CR}$, where as before PS is the Password Strength measured in units of bits, and CR is the estimated crack rate. As CR we used the value of 3.6 million guesses per second, which is aligned with various values other studies used, including 1/2 million guesses per second [14], 3 million guesses per second, but also reports of systems that can guess up to a few billion guesses per second [63].

We aimed to avoid burdening the users with too many recommendations on the screen, so we chose to display three buttons with improvement recommendations. When a recommendation button is pressed, a window appears and presents an explanation of the recommendation, as shown on the right side of Figure 1. The description includes the recommended password and an estimation of how long it will take a hacker to guess it. The recommendation details screen also includes two buttons. One button says ‘Use our recommendation’—clicking this button injects the recommendation into the form and closes the details window. The second button says ‘cancel’, and it simply closes the recommendation window.

3.4 DPAR inputs and configuration

DPAR is designed to generate a strengthening recommendation to any input password and accepts an input that consists of letters, digits, and symbols of any length. In case the input password consists of letters only, DPAR will generate recommendations for all the dimensions (prefix, suffix, l33t, and capitalization). For the case of an all-digits or all-symbol input password, DPAR’s back end will recognize the whole password as a base word and generate prefix and suffix dimensions only. However, this situation is forbidden by the front-end code that requires at least one letter in the password (Recall Figure 1). DPAR can be easily configured to impose limitations and requirements on the user input, such as minimum number of characters, usage of at least one letter or one digit or one symbol or a combination of the above. DPAR optimization algorithm can also be configured to require minimum strength of a password and preference

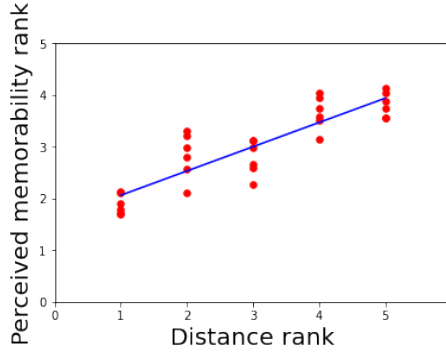


Figure 2: Password Levenshtein distance vs. ranks of perceived memorability, linear regression equation: $y = 0.46X + 1.59$.

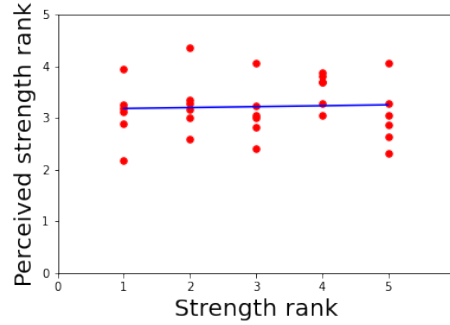


Figure 3: Password strength vs. users' ranks of passwords' strength.

of various dimensions, for instance the prefix dimension strength will be prioritized over the suffix dimension strength, this will lead to more options with strong prefix in the final recommendations array.

4 Perceived memorability study

Our first study tested the relationship between a tweaked password and its perceived memorability. We hypothesized that the closer a password recommendation is to the original password created by the user, it will be perceived as easier to remember. To measure the similarity, we used Levenshtein distance (or edit distance). We expected users to rank passwords as more memorable when the Levenshtein distance from the original password is low. To control for the password's strength, we also asked participants to rank the password strength. However, we expected to find that users' perceptions of password strength are poor, as appears in the literature [60].

4.1 Methodology

A total of $n = 317$ participants successfully completed the study. We recruited participants from Amazon's Mechanical Turk crowdsourcing service, often used in usable security studies [51]. To participate, individuals had to be in the U.S. (to control for cultural and language differences), at least 18 years old, have a Hit approval rate (%) greater than 98 for all participants' hits, and have approved at least 1,000 hits. The institutional ethics review committee authorized the study. Participants were compensated \$0.55 for completing the task, which is the equivalent of living per hour.

We presented an online questionnaire to participants. Each participant was presented with three passwords (randomly selected from a pool of 20 passwords). We created five modifications for each password using Algorithm 1, picking one random mutation per distance for distances 1-5. For each variant of the three passwords, we asked participants to rank their strengths and how easy it is to remember. To do so, we asked the participants to order each of the five modified options per password between 1-5: when one is the weakest, five is the strongest, or one is the most difficult to remember while five is the easiest to remember. The questions and answers were presented randomly to avoid biasing the participants' answers with order effects.

To assess the quality of the participants' data, we performed multiple tests. Initially, 350 participants participated in this study, and 33 (9.4%) were subsequently excluded from the results due to failures in one or more tests, as follows: 17 participants failed straight-line tests [37], 13 submitted incomplete questionnaires, and 3 took an unusually short time to answer the questions. Additionally, we scrutinized the participants' IP addresses to identify and disqualify any addresses listed on blocklists as potential bots, but none of the IP addresses were found on such lists.

We ran a linear regression to analyze how the participants' ranks of password strength and distance correlate with the rank of the password strength based on PESrank and the actual password distance. All statistical tests use a significance level of $\alpha = 0.05$.

4.2 Results

We have measured how users' perceptions of password strength and memorability are related to the actual strength and distance (see Figure 2). The ranks of passwords' strength were fitted to the PESrank strength according to a generalized

linear regression and yielded a poor fit ($R^2=-0.13$). I.e., perceptions of password strength were uncorrelated with the actual strength.

Conversely, the results of ranks of passwords’ memorability that were fitted to the Levenshtein distance according to a generalized linear regression yield a good fit ($R^2=0.68$), see linear regression equation in 2. As is clearly illustrated in the right side of Figure 2, users feel that passwords with a low distance and few mutations are more likely to be used.

5 DPAR user study

Our primary user study evaluated the DPAR system in a scenario resembling a real-world password creation process. We have employed a standard design in password usability studies, similar to several other studies [36, 53, 59]: we created an online study and invited users to register to our online system, guiding them to imagine registering to an important account such as an online banking application or a healthcare account: see the presented instructions in Figure 8 in the appendix.

5.1 Methodology

We recruited participants from Amazon’s Mechanical Turk crowdsourcing service in the U.S.. To participate, individuals had to be at least 18 years old, have a Hit approval rate (%) greater than 98 for all participants’ hits, and have approved at least 1,000 hits. Similar to the memorability study, we paid the equivalent of a living wage to participants, which is \$0.60 for completing Stage 1 and \$0.40 for Stage 2. The Institutional Ethics Committee approved the study.

The study was divided into two stages: (1) creating a password by registering to our system and (2) measuring password recall by logging in again 48 hours later. In addition to the task at hand, at each stage, we also asked the users to fill out a questionnaire. During Stage 1, participants completed a password creation and recall task and were randomly assigned to a password recommendation feedback variant. Stage 2 occurred at least 48 hours after Stage 1 and included a password recall task and questionnaire.

We set an upper bound of password strength: Participants whose initial password strength was very strong (exceeded 54 bits) were automatically allocated to the “feedback only” group since tweaking such passwords typically yields marginal strength improvements. In the questionnaires, we observed that users with strong initial passwords typically used a password generator. Note that the CUPS Password Meter [49] has similar behavior. To distinguish between them and the original feedback-only group, they were assigned to group number 5. No lower bound of password strength was set.

Stage 1 Participants were instructed to create a password for a significant account carrying sensitive information. After creating a username and password, they received feedback on the password strength and a recommendation based on their assigned condition (see below). Participants could choose to register without changing their password or to change it. As an immediate “recall” step, users logged into the application using their possibly modified password and answered a survey about the password creation process.

Stage 2 After 48 hours, participants accessed the second stage and logged in to the application. This step was identical to the Stage 1 “recall” step: Participants needed to remember their password and username and were then directed to the survey presented in the first stage.

Conditions We controlled the study using four independent conditions, as shown in Figure 4:

1. Feedback-only: only the password feedback was displayed, with no recommendations.
2. Feedback+recommendation (asterisks): the buttons displayed the original password masked with asterisks (*), and only the changes suggested by DPAR were in clear text.
3. Feedback+recommendation (number of changes): the buttons displayed the number of changes from the original password.
4. Feedback + recommendation (hack time): the buttons displayed the new estimated hack time of the recommendation.²

DPAR output was configured to provide three recommendations. The display screen was set to show each recommendation on a button. We did not clearly display the recommendations to avoid shoulder surfing risk.

²An error in our code caused the “hack time” that was displayed for the recommendations to be over-estimated. This visual over-estimation was marginal for weak passwords (strength up to ≈ 40 bits) and more pronounced for stronger password. The final strength values that we use in the analysis are unaffected by the visual error.



Figure 4: The graphical user interface for the three different feedback+recommendation button variants.

Participants assigned to the feedback+recommendations conditions could press any of the three buttons and were shown the recommendation details as in Figure 1 (right side). The buttons were ordered according to the number of changes, from the smallest to the largest. They could inject the password into the form by pressing the ‘*use the recommended password*’ button. We randomly assigned the participants to any of the four conditions.

The application was designed to be compatible with both desktop and mobile displays. We chose not to disable any browser features, such as password managers or password storage, to prevent impacting the participants’ browsers. This approach also ensured that we did not provide specific guidance that could influence the password creation process.

Survey Demographic questions were asked, followed by questions on password creation aids and recall methods. Participants were asked about their experiences during the registration process and to respond on a 5-point scale with questions about the process. They were also asked to share their thoughts and feelings about the process and our recommendations. The questionnaire was adapted to be compatible with both desktop and mobile displays. The questions appear in Appendix 1.

Participants The study included 441 participants who completed Stage 1 of the study, and 46.5% of them finished Stage 2, which is in equivalent proportions to [59]. Of the participants, 36% identified as female, 62% identified as male, and the remaining 2% identified as another gender or preferred not to answer. The age range of the participants was 19 to 72 years old, with a median age of 36 (mean 37.67, standard deviation 10.82).

Regarding formal education, 27.9% of the participants received a high school diploma, 62.4% graduated with a bachelor’s degree or trade school certificate, 9.7% graduated with a master’s or Ph.D. degree, and the remaining 0.7% preferred not to disclose their educational background. Additionally, we requested that participants evaluate their level of self-reported computer proficiency. The results indicate that 34% of participants ranked their skills as “very high,” 41.7% ranked as “high,” 22% ranked as “medium,” 2% ranked as “low,” and 0.3% ranked as “very low.” Notably, there were no significant differences in demographic characteristics between conditions.

We used a random allocation process embedded in the user interface; whenever a new participant accessed the web page, a number between 1 to 4 represented the group affiliation. The distribution of the 441 participants was evenly allocated between the four groups in our study as follows: 24% were assigned to group 1, 27% to group 2, and 25% each to groups 3 and 4.

To assess the quality of our data, we conducted several tests. Initially, 500 participants participated in this study, and 59 (11.8%) were subsequently excluded from the results as they all failed to complete the registration task. Specifically, none of them could complete the registration form, which may indicate bot-like behavior. Participants with a very strong initial password, whose strength exceeded 54 bits, were excluded from the statistical analysis ($n = 120$, 27.2%). This was done because there is a less practical need to improve users’ passwords with very strong passwords (Mean=64.63, Std=9.07, Median=61.09). The participants were removed with similar proportions from each condition (26 participants from the feedback-only condition and 31 on average from the feedback + recommendation conditions, Chi-Square=3.09, $p = 0.377$).

Additionally, the order of the questions and answers in the questionnaire were randomly allocated. We conducted straight-line tests [37], reviewed incomplete questionnaires, assessed the time it took to answer the questions, and compared the participants’ IP addresses against blocklists to identify potential bots. All the participants, however, passed the aforementioned tests.

Statistical Analysis To calculate if there is a statistical difference between the number of participants in each group before and after we extracted the participants with strong initial passwords, we used Chi-Square test for proportions.

To assess statistical significance among the conditions in the study, we analyzed the results using two-way mixed factorial analysis of variance (ANOVA) with the password initial and final password as the within-subjects factor and the two main condition groups, feedback-only and feedback + recommendations as the between-subjects factor. A simple before and after effect test with Bonferroni correction was used to analyze the significance further. All statistical tests use a significance level of $\alpha = 0.05$.

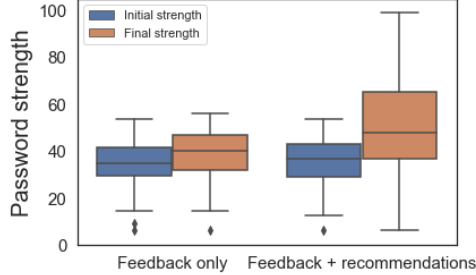


Figure 5: Password strength of feedback-only versus feedback+recommendations participants.

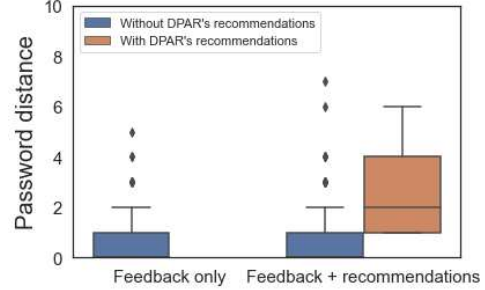


Figure 6: Password similarity between participants who used DPAR's recommendations and those who did not.

5.2 Recommendations versus Feedback-Only

Our first observation from the user study is that among the 238 participants to whom DPAR recommendations were shown, 87 (36.6%) chose to use the recommendations verbatim (by clicking the 'Use our recommendation' button). This is compared to only 7% of the participants that adopted the suggestions of the CUPS password meter as reported in [59]. We consider this to be encouraging: it seems that making the recommendations easily accessible through action buttons drives users to strengthen their passwords at much higher adoption rates than when they only receive textual feedback.

5.3 Password strength and similarity

We found that choosing a recommended password led users to create a stronger password, while it had no significant impact on the usability measurements in the study. As shown in Figure 5, the strengths of the initial passwords created by participants from the feedback-only group and the feedback+recommendations group were almost identical. However, the final passwords chosen by the participants who were shown both feedback and DPAR's recommendations were significantly stronger than their initial passwords (from a mean of 36.3 to 51.8, 42.6%, $p < .001$). In contrast, the final passwords chosen by the participants in the feedback-only condition improved less (from a mean of 34.9 to 38.3, 9.7%, $p < .001$).

No significant differences were found between different variations of the feedback + recommendation conditions. The final password strength among displaying asterisks ($t = -0.89$, $p = 0.38$), number of changes ($t = 1.57$, $p = 0.12$), and hack time ($t = 0.60$, $p = 0.55$) all exhibited no significant difference. In other words, it seems that the choice of text displayed on the buttons has little influence on the strength of the final password.

However, a surprising result is that users have a preference for clicking the upper button. Our analysis shows that the upper button (associated with the fewest changes from the original password) was pressed much more frequently (77 times) than the middle (32 times) and bottom (41 times) buttons, even if the types of recommendations were randomized.

As shown in Figure 6, displaying DPAR recommendation buttons significantly drove users to change their passwords. Participants in the feedback + recommendations condition had a Levenshtein distance average of 2.52, with a minimum of at least one change. On the other hand, participants in the feedback-only condition made only 0.59 changes on average, and participants in the feedback+recommendation condition who did not choose to use the DPAR recommendations exhibited an average distance of 0.52. A t-test for two independent samples between the group that used DPAR's recommendations and the group that did not use them shows a significant effect ($t = -11.095$, $p < 0.001$).

To test the impact of recommendations on usability, we also measured the time participants took to create their accounts. No notable differences were found among the participant groups' password creation time measurements. The individuals who did not utilize password recommendations had comparable creation time measurements. Those who employed recommendations experienced a minor increase in creation time, but this was not statistically significant. Participants who relied on recommendations and manually typed their passwords into the form exhibited a higher increase in password creation time than the other two groups. However, this increase was still not considered significant.

Login Failure	Condition Group	Feedback only	Feedback + recommendations
Immediate	Password kept	0.0%	2.6%
	Password modified	1.9%	2.1%
48 hours	Password kept	22.7%	21.6%
	Password modified	33.3%	19.2%

Table 1: Percentage of participants who failed to login after the registration process

5.4 Password Recall

To test the recall of the passwords, we asked participants to log in immediately after setting the passwords and again in Stage 2, 48 hours after the creation of the passwords.

In the second login attempt, made 48 hours after the password creation, 51.2% of the feedback + recommendation groups returned to the second part of the experiment, whereas only 41.6% of the feedback-only group had returned. As appears in Table 1, the percentage of failed second login among the participants in the feedback-only group who modified their passwords (33.3%) was higher than those in the feedback + recommendation groups (19.2%). While the difference between the proportions was not significant ($p = 0.357$), they allow us to conclude that the feedback + recommendation group did not underperform in comparison to password feedback methods.

We control for demographic variables such as age, gender, and education, as well as computer literacy and found no significant difference in login failure rates, see the logistic regression analysis in Table 4 in the Appendix.

It is also important to stress that the percentage of failed logins on the second attempt, observed after a 48-hour interval, is comparable to the rates of second-attempt login failures reported in other studies: 21.8% [59], 19% [74], 19%–28% (across 10 experiment groups) [56], 20% [78] and 28.6%–79% (across 7 experiment groups) [73].

We observed that participants who created their password using a DPAR recommendation and successfully logged in to the system after 48 hours had an average Levenshtein distance score of 2.86, significantly lower than that of those who failed to log in (Mean=3.17), ($F = 4.296$, $p < 0.001$). This supports our assertion that making fewer changes to the password (i.e., a lower Levenshtein distance) improves password recall beyond the (same) perception reported in Section 4.

Furthermore, in the questionnaire presented immediately after the password was registered, we inquired about the participants’ self-assessed confidence in their ability to recall their passwords, offering the options “Easy to remember” and “Difficult to remember”. We found that among those who reported “Easy”, only 5% of those who used one of DPAR’s recommendations failed to remember the password 48 after creation, whereas 20% of those who didn’t use DPAR recommendations failed the second login despite claiming it was “Easy”. In addition, those who reported “Difficult to remember” in fact, found the passwords difficult to remember and had a similarly high fail ratio regardless of whether they used one of DPAR’s recommendations (42%) or not (38%). Thus, we argue that users’ self-assessment of the memorability of DPAR recommendations is aligned with their actual ability to recall the passwords.

5.5 User interface and attitudes

To better understand the experience the participants had with DPAR, we used an online questionnaire. Table 2 summarizes the replies to the questionnaire of the participants in the feedback+recommendations conditions. More than 80% of the participants reported they understood the task and found it easy to create the password. On the other hand, only between 28% and 34% replied that the experience of working with DPAR was negative (i.e., difficult, confusing, annoying, or redundant). In general, the qualitative feedback and sentiment of the participants are positive and may show users can adopt that password recommendation system. We also noticed that 36.4% of participants reported they did not see the recommendations³, although they were shown. We mentioned above that 36.6% of the users used DPAR’s recommendations, so those participants saw the recommendations. Therefore, the finding that a significant fraction of participants did not see the recommendations may indicate that there is room to improve the user interface and increase the adoption rate of the recommendations even higher.

For those who reported the password was difficult to remember, we found two main types of explanations: (i) the password was too long, and (ii) the password contained digits or symbols which the users were not accustomed to. Typical feedback was, for instance, “*It was difficult to remember because it added random letters and symbols to my password*”, and “*The password was difficult to remember since I had never used it before and the addition of*

³The fraction of those who claimed “I did not see the recommendations” is perfectly aligned with the number participants who were allocated to groups 4 and 5 and didn’t see any recommendations.

Feedback	Percentage
I understood the given task	89.7%
I felt it was easy to create the password	80.9%
The recommendations were helpful	63.2%
Creating the password was fun	46.5%
I did not see the recommendations	36.4%
Creating the password was difficult	34.6%
The recommendations were annoying	30.9%
The recommendations were redundant	30.6%
The recommendations were confusing	28.3%

Table 2: The percentage of participants who agreed or very much agree or agree with the statement

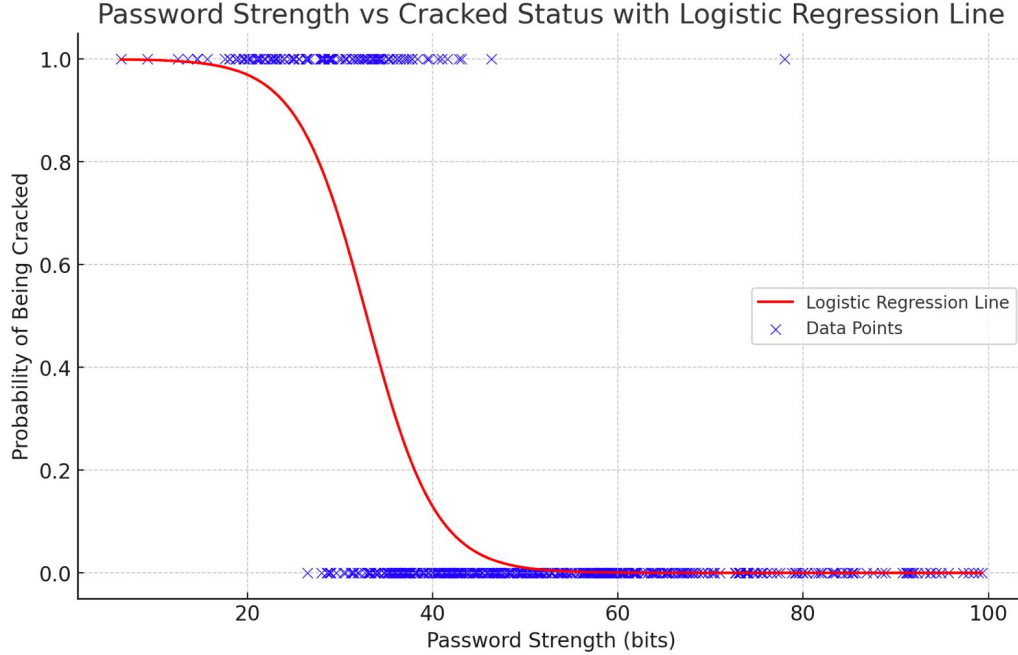


Figure 7: Scatter plot and logistic regression curve of the passwords being cracked by an online hash lookup service as a function of password strength. The regression curve’s log-odds of a password being cracked are $\log(p/(1-p)) = 8.813 - 0.268X$.

various numbers and symbols made it hard to remember”. In most cases, those who reported the password was easy to remember did not add further explanations. But some of the users did recognize that the DPAR only offered minor changes, “*easy because I used a system to make it easy for me to remember and because the suggested change was a small one*”, “*Easy, because it was only changed by a little bit*”, and “*It was easy because it was just a slight adjustment to the password that I had chosen.*”.

5.6 Validation of the PESrank strength

Our data allowed us to validate that the PESrank strength is correlated with stronger passwords. Following common authentication system designs, we did not store the passwords themselves and only kept their SHA256 hash values to let the users log in a second time. Thus, we collected a total of 598 hashes of user-selected passwords. Having the password hashes allows us to validate the correlation between password strengths (as calculated by PESrank) and real password “crackability”. For this purpose, we attempted to crack the passwords, using 3 popular online hash lookup services [29, 9, 43] to evaluate if the password was compromised or not. The results are shown in Figure 7. The graph shows the expected inverse correlation: passwords with higher PESrank scores are less likely to be crackable.

We also performed a logistic regression analysis to examine the influence of password strength, measured in bits, on the likelihood of a password being compromised. The model’s goodness-of-fit was substantial, with a pseudo R^2 value of 0.9064, indicating that PESrank strength can significantly predict the probability of cracking. The coefficient for

password strength was found to be -0.268 , underscoring an inverse relationship; as password strength increases, the likelihood of being cracked decreases.

6 Discussion

In this paper, we described the design and evaluation of DPAR, a Data-driven Password Recommendation system that helps users pick up stronger passwords while being able to recall them. We found that users who had used DPAR’s password recommendations had ended up with stronger passwords without significantly impacting their ability to recall the passwords. This may suggest that minor modifications to a password that the user initially chose can be recalled similarly to passwords that the user modified. DPAR differs significantly from the approach adopted by password meters [25, 59, 20], as it spares users the need to navigate the complexities of password security guidelines or best practices.

Our user study points to several promising aspects of using recommendation systems in the domain of passwords. DPAR recommendations were chosen by 36.6% of the participants in the user study, compared to only 7% of the participants that adopted the suggestions of the CUPS password meter as reported in [59]. Most participants who viewed DPAR’s recommendations chose the slightest possible change over their original password and the other recommendations. This may indicate that users prefer to make as few changes as possible.

Our findings also point to further complexities that interface designers must face when designing password recommendation systems. For example, users are biased toward clicking on the upper button, even if the order of the buttons is randomized. This phenomenon is a classic example of position effects, where the probability of clicking on a button is influenced by its position [10]. Each recommendation button represents a trade-off between security and usability, or in behavioral economics terms, a trade-off between cost and benefit; when presented with three recommendation buttons, the upper button is the most immediate to find, and it, therefore, represents higher usability and lower security, the bottom button represents higher security and lower usability, and the one in the middle represents intermediate usability and security. Thus, we speculated that most users would choose the middle ground of usability and security in the cost and benefit [2]—and this was not the case. We found that most users preferred usability over security. This can emphasize the lack of understanding of security, or perhaps users don’t see a lot of benefit in security, thus favoring usability.

Our findings point to new ways to improve password creation processes. Passwords remain important despite their shortcomings. There is ongoing debate regarding passwords’ continued relevance in the face of the growing adoption of alternative authentication solutions [3], and the declaration of the death of passwords seems premature [50, 30, 81]. However, the usefulness of passwords remains high. Laboratory studies have shown that users highly rate passwords regarding preference, usability, and intention to use, and lowest in terms of expected problems and effort [81, 6]. At the same time, while password managers and biometric authentications are becoming more prevalent, they still require passwords as a backup method. Therefore, we argue that the need to help users create more robust and better passwords is a paramount concern that research should address.

6.1 Limitations

When considering the generalizability of our findings, it is essential to understand several limitations of our study. First, we should be aware of some limitations to the usage of m-turk in online studies [57]. In the DPAR study, we redirected users to an external application, asking them to complete registration to a website. We also conducted various tests to ensure the quality of our data. Second, the participants in this study were asked to create a password, but unlike a real-life scenario where they would protect a valuable asset, they were not tasked with securing an actual account. This may have led to a reduced motivation to create a secure password compared to those who create passwords to protect important accounts. To mitigate this effect, we instructed the participants to create a password for their high-value online bank accounts or an essential email account, following best practices from password creation studies [36, 53]. In addition, only U.S.-based participants and ASCII-based recommendations were tested in our study, thus the ecological validity is limited to these boundaries, as password composition heavily differs around the world [4, 68].

Another limitation of our study is the relatively short time in measuring password recall. We measured the recall twice, right after the participants created the passwords and 48 hours later, as was done in a few recent studies [59, 36], but in real-life scenarios, users must remember passwords for long periods of time: days, weeks, and even months after creating them. Many other factors may affect the recall of these passwords, such as frequency of use and the importance of the account.

To assess the overall contribution of our approach compared to password meters, we note that the mean strength of passwords created without any help was 36.3 bits and improved via DPAR recommendations to a mean of 51.8 bits,

while the strength of random and unwieldy passwords generated by password managers was significantly stronger (mean=64.63 bits). Therefore, it is important to note that passwords that are memorable in any way will most probably never reach the strength of fully generated passwords and that the improvements we suggest are inherently limited.

6.2 Deploying DPAR and Future Work

While DPAR helps users improve their passwords and better protect their accounts from hackers, it may also introduce new privacy risks. If not implemented carefully, there’s a risk that passwords might be inadvertently exposed. While the risk is not different than the risk imposed by existing password management systems [58], adding another layer that has access to password content may introduce new risks. These may include the interception of passwords during transmission, unauthorized access due to insufficient security controls, and the possibility of re-identification from anonymized password data. Moreover, the aggregation of password data could inadvertently create a single point of failure, significantly amplifying the impact of any security breach. Given these risks, there is a pressing need for future research focused on privacy-enhancing technologies in the context of password recommendations.

Existing works on similar problems may point to possible directions. For example, Emmadi et al. implemented a password meter using Fully Homomorphic Encryption (FHE), which performs end-to-end query privacy for the users’ passwords [18]. Additionally, developing decentralized approaches that allow password strength evaluation to occur locally on the user’s device could eliminate the need to transmit sensitive information, further preserving privacy. The objective of future research would be to provide robust password recommendations while adhering to the standards of user privacy and data protection.

Finally, a potential concern with implementing DPAR as a password recommendation system is that it might prompt attackers to modify their strategies, potentially negating some of the benefits of increased password strength. However, we note that password cracking tools, like hashcat [28] and John the Ripper [47], already employ techniques to tweak passwords, and the PESrank measure is designed to simulate such cracking behavior. If DPAR recommendations become popular and crackers adapt, this will change the observed distributions of prefixes, suffixes, l33t, and capitalizations – and one of PESrank’s advantages is its ability to re-train it with additional corpora.

7 Conclusions

Our study provides a crucial first step in building and implementing password recommendation systems. We have shown that recommendations can add an effective interaction for helping people create stronger passwords while preserving similar recall to password meters. We hope our method can open new avenues for work in making password recommendations useful in real-world systems. Future research should extend the ecological validity of the study, especially when users interact with password recommendations in more diverse contexts.

References

- [1] Yasmeen Abdrabou, Johannes Schütte, Ahmed Shams, Ken Pfeuffer, Daniel Buschek, Mohamed Khamis, and Florian Alt. “your eyes tell you have used this password before”: Identifying password reuse from gaze and keystroke dynamics. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2022.
- [2] Dan Ariely and Thomas S. Wallsten. Seeking subjective dominance in multidimensional space: An explanation of the asymmetric dominance effect. *Organizational Behavior and Human Decision Processes*, 63(3):223–232, 1995.
- [3] Michael Bachmann. Passwords are dead: Alternative authentication methods. In *2014 IEEE Joint Intelligence and Security Informatics Conference*, pages 322–322, 2014.
- [4] Joseph Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *2012 IEEE symposium on security and privacy*, pages 538–552. IEEE, 2012.
- [5] Joseph Bonneau and Sören Preibusch. The password thicket: Technical and market failures in human authentication on the web. In *The 9th Workshop on the Economics of Information Security (WEIS)*, 2010.
- [6] Leon Bošnjak and Boštjan Brumen. Rejecting the death of passwords: Advice for the future. *Computer Science and Information Systems*, 16(1):313–332, 2019.
- [7] Claude Castelluccia, Markus Dürmuth, and Daniele Perito. Adaptive password-strength meters from markov models. In *NDSS*, 2012. https://www.ndss-symposium.org/wp-content/uploads/2017/09/06_3.pdf.
- [8] Sudhanshu Chauhan and Nutan Kumar Panda. Chapter 11 - online security. In *Hacking Web Intelligence*, pages 203–216. Syngress, Boston, 2015.

- [9] Crackstation. Crackstation, 2023. <https://crackstation.net/>.
- [10] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*, pages 87–94, 2008.
- [11] L. David and A. Wool. Rank estimation with bounded error via exponential sampling. *Journal of Cryptographic Engineering*, 12:151–168, 2022.
- [12] Liron David and Avishai Wool. Pesrank: An explainable online password strength estimator. *J. Comput. Secur.*, 30(6):877–901, 2022.
- [13] Xavier de Carné de Carnavalet and Mohammad Mannan. From very weak to very strong: Analyzing password-strength meters. In *Network and Distributed System Security Symposium (NDSS 2014)*. Internet Society, 2014.
- [14] Alessia Michela Di Campi. Password guessing: Learn the nature of passwords by studying the human behavior, 2021. M.Sc. thesis, Università Ca’Foscari Venezia.
- [15] Jeff Dunsavage. Despite warnings, weak password policies still invite cybercrime. Technical report, January 2023.
- [16] Markus Dürmuth, Fabian Angelstorf, Claude Castelluccia, Daniele Perito, and Abdelberi Chaabane. Omen: Faster password guessing using an ordered Markov enumerator. In *International Symposium on Engineering Secure Software and Systems*, pages 119–132. Springer, 2015.
- [17] Serge Egelman, Andreas Sotirakopoulos, Ildar Muslukhov, Konstantin Beznosov, and Cormac 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*, pages 2379–2388, 2013.
- [18] Nitesh Emmadi, Imtiyazuddin Shaik, Harshal Tupsamudre, Harika Narumanchi, Rajan Mindigal Alasingara Bhattachar, and Sachin Premasukh Lodha. Privacy-preserving password strength meters with fhe. In *Cyber Security Cryptography and Machine Learning: 5th International Symposium, CSCML 2021, Be’er Sheva, Israel, July 8–9, 2021, Proceedings 5*, pages 94–103. Springer, 2021.
- [19] Dinei Florencio and Cormac Herley. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web*, pages 657–666, 2007.
- [20] Alain Forget, Sonia Chiasson, Paul C Van Oorschot, and Robert Biddle. Improving text passwords through persuasion. In *Proceedings of the 4th Symposium on Usable Privacy and Security*, pages 1–12, 2008.
- [21] Steven Furnell, Warut Khern am nuai, Rawan Esmael, Weining Yang, and Ninghui Li. Enhancing security behaviour by supporting the user. *Computers and Security*, 75:1–9, 2018.
- [22] Lori Glavin. The 2023 workforce authentication report: Embracing the passwordless future. Technical report, FIDO Alliance, Oct 2023.
- [23] Maximilian Golla and Markus Dürmuth. On the accuracy of password strength meters. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1567–1582, 2018.
- [24] Dan Goodin. Anatomy of a hack: How crackers ransack passwords like “qeadzcxwrsfxv1331”. *Ars Technica*, 2013.
- [25] Yimin Guo and Zhenfeng Zhang. LPSE: Lightweight password-strength estimation for password meters. *Computers & Security*, 73:507–518, 2018.
- [26] Ebu Yusuf Güven, Ali Boyaci, and Muhammed Ali Aydin. A novel password policy focusing on altering user password selection habits: A statistical analysis on breached data. *Computers & Security*, 113:102560, 2022.
- [27] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.
- [28] hashcat. Hashcat advanced password recovery, 2019. <http://hashcat.net>.
- [29] Hashes. Hashes, 2023. <https://hashes.com/en/decrypt/hash>.
- [30] Cormac Herley and Paul Van Oorschot. A research agenda acknowledging the persistence of passwords. *IEEE Security and privacy*, 10(1):28–36, 2011.
- [31] Gongzhu Hu. On password strength: A survey and analysis. *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 165–186, 2018.
- [32] Jun Ho Huh, Seongyeol Oh, Hyoungshick Kim, Konstantin Beznosov, Apurva Mohan, and S Raj Rajagopalan. Surpass: System-initiated user-replaceable passwords. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 170–181, 2015.

- [33] Checkpoint Inc. Check point 2022 cyber security report. Technical report, 2022. <https://resources.checkpoint.com/cyber-security-resources/2022-cyber-security-report>.
- [34] Philip G. Inglesant and M. Angela Sasse. The true cost of unusable password policies: Password use in the wild. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 383–392, New York, NY, USA, 2010. Association for Computing Machinery.
- [35] Shipi Kankane, Carlina DiRusso, and Christen Buckley. Can we nudge users toward better password management? an initial study. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2018.
- [36] Patrick Gage Kelley, Saranga Komanduri, Michelle L Mazurek, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *2012 IEEE symposium on security and privacy*, pages 523–537. IEEE, 2012.
- [37] Yujin Kim, Jennifer Dykema, John Stevenson, Penny Black, and D Paul Moberg. Straightlining: Overview of measurement, comparison of indicators, and effects in mail–web mixed-mode surveys. *Social Science Computer Review*, 37(2):214–233, 2019.
- [38] Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L Mazurek, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Serge Egelman. Of passwords and people: Measuring the effect of password-composition policies. In *Proceedings of the sigchi conference on human factors in computing systems*, pages 2595–2604, 2011.
- [39] Kevin Lee, Sten Sjöberg, and Arvind Narayanan. Password policies of most top websites fail to follow best practices. In *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*, pages 561–580, 2022.
- [40] Michael D Leonhard and VN Venkatakrishnan. A comparative study of three random password generators. In *2007 IEEE International Conference on Electro/Information Technology*, pages 227–232. IEEE, 2007.
- [41] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.
- [42] Peter Mayer, Collins W Munyendo, Michelle L Mazurek, and Adam J Aviv. Why users (don’t) use password managers at a large educational institution. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1849–1866, 2022.
- [43] md5decrypt. Md5decrypt, 2023. <https://md5decrypt.net/en/Sha256/#answer>.
- [44] William Melicher, Blase Ur, Sean M Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Fast, lean, and accurate: Modeling password guessability using neural networks. In *Proc. 25th USENIX Security Symposium*, pages 175–191, 2016.
- [45] Avirup Mukherjee, Kousshik Murali, Shivam Kumar Jha, Niloy Ganguly, Rahul Chatterjee, and Mainack Mondal. Mascara: Systematically generating memorable and secure passphrases. *arXiv preprint arXiv:2303.09150*, 2023.
- [46] Sean Oesch, Scott Ruoti, James Simmons, and Anuj Gautam. “it basically started using me:” an observational study of password manager usage. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–23, 2022.
- [47] OpenWall. John the ripper password cracker, 2019. <http://www.openwall.com/john>.
- [48] Sarah Pearman, Shikun Aerin Zhang, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Why people (don’t) use password managers effectively. In *Fifteenth Symposium On Usable Privacy and Security (SOUPS 2019)*. USENIX Association, Santa Clara, CA, pages 319–338, 2019.
- [49] CyLab Usable Privacy and Security Laboratory. CUPS password meter, 2023. <https://cups.cs.cmu.edu/meter/>.
- [50] Nils Quermann, Marian Harbach, and Markus Dürmuth. The state of user authentication in the wild. *WAY*, 18, 2018.
- [51] Elissa M Redmiles, Sean Kross, and Michelle L Mazurek. How well do my results generalize? comparing security and privacy survey results from mturk, web, and telephone samples. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1326–1343. IEEE, 2019.
- [52] Karen Renaud and Verena Zimmermann. Nudging folks towards stronger password choices: Providing certainty is the key. *Behavioural Public Policy*, 3(2):228–258, 2019.

- [53] Richard Shay, Saranga Komanduri, Adam L Durity, Phillip Huh, Michelle L Mazurek, Sean M Segreti, Blase Ur, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Can long passwords be secure and usable? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2927–2936, 2014.
- [54] Chao Shen, Tianwen Yu, Haodi Xu, Gengshan Yang, and Xiaohong Guan. User practice in password security: An empirical study of real-life passwords in the wild. *Computers and Security*, 61:130–141, 2016.
- [55] Ahmet Ali Süzen. A risk-assessment of cyber attacks and defense strategies in industry 4.0 ecosystem. *International Journal of Computer Network and Information Security*, 12(1), 2020.
- [56] Joshua Tan, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Practical recommendations for stronger, more usable passwords combining minimum-strength, minimum-length, and blocklist requirements. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1407–1426, 2020.
- [57] Jenny Tang, Eleanor Birrell, and Ada Lerner. Replication: How well do my results generalize now? the external validity of online privacy and security surveys. In *Eighteenth symposium on usable privacy and security (SOUPS 2022)*, pages 367–385, 2022.
- [58] Eran Toch, Claudio Bettini, Erez Shmueli, Laura Radaelli, Andrea Lanzi, Daniele Riboni, and Bruno Lepri. The privacy implications of cyber security systems: A technological survey. *ACM Computing Surveys (CSUR)*, 51(2):1–27, 2018.
- [59] Blase Ur, Felicia Alfieri, Maung Aung, Lujo Bauer, Nicolas Christin, Jessica Colnago, Lorrie Faith Cranor, Henry Dixon, Pardis Emami Naeini, Hana Habib, et al. Design and evaluation of a data-driven password meter. In *Proceedings of the 2017 chi conference on human factors in computing systems*, pages 3775–3786, 2017.
- [60] Blase Ur, Jonathan Bees, Sean M Segreti, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Do users’ perceptions of password security match reality? In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 3748–3760, 2016.
- [61] Blase Ur, Patrick Gage Kelley, Saranga Komanduri, Joel Lee, Michael Maass, Michelle L Mazurek, Timothy Passaro, Richard Shay, Timothy Vidas, Lujo Bauer, et al. How does your password measure up? the effect of strength meters on password creation. In *USENIX Security Symposium*, pages 65–80, 2012.
- [62] Blase Ur, Fumiko Noma, Jonathan Bees, Sean M Segreti, Richard Shay, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. I added ‘!’ at the end to make it secure”: Observing password creation in the lab. In *Proc. SOUPS*, 2015.
- [63] Blase Ur, Sean M Segreti, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Saranga Komanduri, Darya Kurilova, Michelle L Mazurek, William Melicher, and Richard Shay. Measuring real-world accuracies and biases in modeling password guessability. In *24th USENIX Security Symposium*, pages 463–481, 2015.
- [64] US Department of Interior. P@s\$w0rds at the U.S. Department of the Interior: Easily cracked passwords, lack of multifactor authentication, and other failures put critical DOI systems at risk, January 2023.
- [65] Verizon Security Team. Verizon 2022 data breach investigations report. Technical report, 2022. <https://www.verizon.com/business/resources/reports/dbir/>.
- [66] Emanuel Von Zezschwitz, Alexander De Luca, and Heinrich Hussmann. Survival of the shortest: A retrospective analysis of influencing factors on password composition. In *Human-Computer Interaction–INTERACT 2013: 14th IFIP TC 13 International Conference, Cape Town, South Africa, September 2-6, 2013, Proceedings, Part III 14*, pages 460–467. Springer, 2013.
- [67] Ding Wang and Ping Wang. The emperor’s new password creation policies: An evaluation of leading web services and the effect of role in resisting against online guessing. In *Computer Security–ESORICS 2015: 20th European Symposium on Research in Computer Security, Vienna, Austria, September 21-25, 2015, Proceedings, Part II 20*, pages 456–477. Springer, 2015.
- [68] Ding Wang, Ping Wang, Debiao He, and Yuan Tian. Birthday, name and bifacial-security: Understanding passwords of chinese web users. In *28th USENIX security symposium (USENIX security 19)*, pages 1537–1555, 2019.
- [69] Rick Wash and Emilee Rader. Prioritizing security over usability: Strategies for how people choose passwords. *Journal of Cybersecurity*, 7(1):tyab012, 2021.
- [70] Rick Wash, Emilee Rader, Ruthie Berman, and Zac Wellmer. Understanding password choices: How frequently entered passwords are re-used across websites. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 175–188, 2016.
- [71] Matt Weir, Sudhir Aggarwal, Breno De Medeiros, and Bill Glodek. Password cracking using probabilistic context-free grammars. In *2009 30th IEEE Symposium on Security and Privacy*, pages 391–405. IEEE, 2009.

- [72] Dan Wheeler. zxcvbn: Realistic password strength estimation. *Dropbox TechBlog*, 2012. <https://blogs.dropbox.com/tech/2012/04/zxcvbn-realistic-password-strength-estimation>.
- [73] Simon S Woo and Jelena Mirkovic. Improving recall and security of passphrases through use of mnemonics. In *Proceedings of the 10th International Conference on Passwords (Passwords)*, 2016.
- [74] Simon S Woo and Jelena Mirkovic. Guidedpass: Helping users to create strong and memorable passwords. In *Research in Attacks, Intrusions, and Defenses: 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings 21*, pages 250–270. Springer, 2018.
- [75] Naomi Woods. Frequently using passwords increases their memorability-a false assumption or reality? In *AMCIS 2017: Proceedings of the Twenty-third Americas Conference on Information Systems, ISBN 978-0-9966831-4-2*. AIS Electronic Library (AISeL), 2017.
- [76] Xiaoyuan Wu, Collins W Munyendo, Eddie Cosic, Genevieve A Flynn, Olivia Legault, and Adam J Aviv. User perceptions of five-word passwords. In *Proceedings of the 38th Annual Computer Security Applications Conference*, pages 605–618, 2022.
- [77] Weining Yang, Ninghui Li, Omar Chowdhury, Aiping Xiong, and Robert W Proctor. An empirical study of mnemonic sentence-based password generation strategies. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1216–1229, 2016.
- [78] Bei Ye, Yajun Guo, Lei Zhang, and Xiaowei Guo. An empirical study of mnemonic password creation tips. *Computers & Security*, 85:41–50, 2019.
- [79] M Yıldırım and Ian Mackie. Encouraging users to improve password security and memorability. *International Journal of Information Security*, 18:741–759, 2019.
- [80] Leah Zhang-Kennedy, Sonia Chiasson, and Paul van Oorschot. Revisiting password rules: Facilitating human management of passwords. In *2016 APWG symposium on electronic crime research (eCrime)*, pages 1–10. IEEE, 2016.
- [81] Verena Zimmermann and Nina Gerber. The password is dead, long live the password—a laboratory study on user perceptions of authentication schemes. *International Journal of Human-Computer Studies*, 133:26–44, 2020.
- [82] Verena Zimmermann, Karola Marky, and Karen Renaud. Hybrid password meters for more secure passwords—a comprehensive study of password meters including nudges and password information. *Behaviour & Information Technology*, 42(6):700–743, 2023.

A Appendix

Criteria	DPAR	CUPS [59]	PTP [20]	MASCARA [45]
Based on user’s password	Yes	Yes	Yes	No
Password strength model	PESrank	Neural network	Password cracking	Password cracking
Strength is estimated during creation	Yes	Yes	No	No
Password usability model	Edit distance, recall tests	Recall tests	Recall tests	Character error rate
Usability is estimated during creation	Yes	No	No	Yes
Number of recommendations	3	1	1	3
Recommendation creation mechanism	Data-driven	Heuristics and policies	Random	Random
What in the password is optimized?	Strength and usability	None	None	Strength and usability
Both meter and textual feedback	Yes	Yes	No	No
May choose original password	Yes	Yes	No	No

Table 3: A comparison between DPAR and state-of-the-art password creation systems

A.1 Questionnaire

1. What is your age?
2. What is the highest degree or level of school you have completed? If currently enrolled, highest degree received.
 - Some High School
 - High School
 - Bachelor’s Degree
 - Master’s Degree

Variable	Coefficient	Std. Error	z-value	P _z —z—	[0.025	0.975]
const	2.6197	1.106	2.369	0.018	0.452	4.787
Experiment Group	-0.0114	0.191	-0.060	0.952	-0.386	0.363
Age	-0.0120	0.021	-0.570	0.569	-0.053	0.029
Gender	-0.5259	0.417	-1.261	0.207	-1.343	0.291
Computer Literacy	-0.2471	0.274	-0.902	0.367	-0.784	0.290
Level of Education	0.0539	0.196	0.275	0.783	-0.330	0.438

Table 4: Logistic Regression Results

This is an experiment on people reactions to website registration processes.

In this experiment you will be asked to create a username and a password for yourself.

Please imagine that this is a significant account that carries sensitive information (for instance bank account or health information).

Next, you'll be asked to login to our system and fill out a basic survey.

Two days later you'll be asked to login again and complete the survey.

Please note that the survey is confidential and anonymized according to guidelines from our ethics review board.



Figure 8: DPAR user study instructions

- Ph.D. or higher
 - Trade School
 - I prefer not to answer
3. What gender do you identify as?
- Female
 - Male
 - Other
 - I prefer not to answer
4. How do you rate your computer literacy and skills?
- Very high
 - High
 - Medium
 - Low
 - Very low
5. I remembered the username and password I chose:
- Yes
 - No
 - Only after several attempts
6. I used a password manager, password generator, or browser feature to create my password
- Yes
 - No
7. I wrote down my password
- Yes
 - No
8. Please respond per each statement below with the most suitable answer (Strongly agree, agree, natural, disagree, strongly disagree):
- Creating the password was challenging
 - Creating the password was fun

- Creating the password was annoying
 - Creating the password was easy
 - Creating the password was difficult
 - Creating the password was confusing
 - The instructions on the registration page were clear
 - The recommendations buttons were visible
 - I understood the task I was given
9. Please respond per each statement below with the most suitable answer (Strongly agree, agree, natural, disagree, strongly disagree):
- The recommendations were helpful
 - The recommendations were redundant
 - I didn't see the recommendations
 - The recommendations were annoying
 - The recommendations were confusing
10. The initial password I suggested:
- I reused a previous password
 - I modified a password I used in the past
 - I created an entirely new password
 - I used a password manager or a password generator to create the password
11. The final password that I used:
- I used one of the passwords that were suggested by the recommendation system
 - I modified my password but didn't use any of the passwords suggested by the recommendation system
 - I didn't use the recommendation system
 - I didn't change the original password I created
12. Was it easy or difficult to remember the password, and why?
13. What did you think about the password recommendations we suggested?
14. Please write down any comments you might have regarding this study (if you had difficulty understanding the questions, any issues related to the content or the study format, etc.).

Algorithm 1 Recommendations Generation Process

```

1: Input data from the password the user created:
2: baseword, prefix, suffix, l33t, capitalization  $\leftarrow$  password
3: digSymList  $\leftarrow$  [0...9] + [!..?]
4:
5: procedure GENERATEPREFIXSUFFIX(str)
6:   if str is empty then
7:     return 3 lists of 1, 2 and 3 random digits and symbols
8:   else
9:     return 3 lists, randomly replace existing digits and symbols
10:
11: procedure GENERATEL33T(l33t)
12:   l33tcandidate  $\leftarrow$  randomly choose one l33t candidate
13:   if l33tcandidate is not empty then
14:     return l33t transformation identifier
15:
16: procedure GENERATECAPITALIZATION(capitalization)
17:   capitalizationPos  $\leftarrow$  random position in the base word
18:   if capitalizationPos is not empty then
19:     return capitalization position
20:
21: procedure POWERSET(prefix', suffix', l33t', capitalization')
22:   combinationlist  $\leftarrow$  a powerset of all the groups
23:
24: for  $i = 1 \dots n_1$  do
25:   prefix'  $\leftarrow$  prefix'  $\cup$  GeneratePrefixSuffix(prefix)
26:   suffix'  $\leftarrow$  suffix'  $\cup$  GeneratePrefixSuffix(suffix)
27:
28: prefix'  $\leftarrow$  original  $\cup$  prefix'
29: suffix'  $\leftarrow$  original  $\cup$  suffix'
30: l33t'  $\leftarrow$  original  $\cup$  GenerateL33t(l33t)
31: capitalization'  $\leftarrow$ 
32:   original  $\cup$  GenerateCapitalization(capitalization)
33:
34: DPAR recommendations  $\leftarrow$  powerset(prefix', suffix', l33t', capitalization')

```
