# A New Data-Driven Password Strength Meter for Enhancing Cybersecurity Using Attention Tracking

Moath M. Algharibeh
*Department of Electrical and Computer Engineering*
*Oakland University*
Rochester, USA
Algharibeh@oakland.edu

Ghaith Husari
*Department of Computer Science*
*East Tennessee State University*
Johnson City, USA
husari@etsu.edu

Sardar Jaf
*Department of Computer Science*
*University of Sunderland*
Sunderland, U.K.
sardar.jaf@sunderland.ac.uk

***Abstract*—Password-based authentication is the most popular authentication mechanism over insecure networks due its simplicity and convenience. To ensure the security of this authentication mechanism, measuring the strength of users' passwords becomes a crucial task to guide users to create stronger passwords. However, password strength meters are only helpful if they are accurate. Passwords meters that do not provide accurate scores that reflect the actual passwords strengths, e.g., providing a high score for a weak password, may misinform users and hinder the overall security of password-based authentication mechanisms. While many password strength meters were proposed in the literature, the lack of a standardized method to measure password strengths and comparing the accuracy of different password meters, selecting the most appropriate password meter will remain a difficult and unclear process.**

**In this paper, we propose and implement a data-driven password meter that scrapes and collects large datasets to be used by the proposed password strength meter to help provide more accurate scores. Also, we measured the influence of the proposed meter at guiding users to create stronger passwords by tracking their eye movements. To do this, we conducted a user study on a testing web service and monitored the eye movements of our users using an eye tracking tool. Our results exhibited a significant improvement by influencing 88% of users to create an average of 150 years for password cracking-time.**

***Index Terms*—password meter, authentication security, eye-tracker, time-to-crack.**

## I. Introduction

Passwords are ubiquitous in all system aspects. It is protecting the system that contains the slightest information to the most critical information systems. However, due to the importance of this protection, attackers are continuously developing new techniques in guessing and cracking users' passwords (both offline and online). In many cases, users were forced and guided to choose a stronger password complying with the password policies.

Furthermore, password policies alienate users at the moment of password creation; either as a consequence of a non-obvious meter where users are dealing with it as a black box or by the different results of each meter that influence users to create a stronger password, so the need of an influencer password-meter that genuinely guides the end-user to create a stronger password is now essential. This paper aims to solve the lack of user awareness regarding password creation cybersecurity problem. As well, we have developed a new accurate, time-efficient and dynamic technique to measure password strength depending on the password cracking-time.

One of the leading cybersecurity problems is the password strength; the more password is strong; the more system is cyber secured. This paper solved one of the main problems raised in user's awareness in password strength creation that could affect the user practice in all different systems as a password acceptance criterion. Additionally, we shed light on the lack of a user's awareness of password creation by the literatures. Then we used different methods to solve this lack by influencing the end-user to create a stronger password using a contributed and an accurate novel password-meter. The method used to prove the effectiveness of our novel meter was a questionnaire, the test of our contributed meter through more than 245 participants and the test of the eye-fixation through an eye tracker lab, then we analysed the created participant's passwords, and crosstab it with their questionnaire answers. As well, we presented the key contribution out of this paper.

Furthermore, we discussed the analysis in the discussion section, then we included the limitation of this work and how can we mitigate these limitations in the future work section.

### A. KEY FINDINGS

Reviewing the literatures in this space, apparently, reveals some knowledge gaps both on the basic research and applied research sides. For instance, using cracking-time as a visual meter. Likewise, the use of eye-tracker from Human-Computer Interaction (HCI) perspective whilst creating the password. Finally, the use of static algorithms instead of dynamism and automation. The following sections will address these knowledge gaps that we have found in the literatures so far.

*1) AIMS & OBJECTIVES:* **AIMS**

- Adjust and solve the lack of user awareness on the password creation cybersecurity problem.
- Prove a technical solution for the main problem on the cybersecurity field (Password Creation).

**OBJECTIVES**

- Develop a new accurate, time-efficient and dynamic technique to measure password strength depending on the password cracking-time.

- Study the impact of the proposed technique on how it is influencing the end-user to create a stronger password.

The rest of the paper is organized as follows. In Section 2, we discuss the related work. In Section 3, we describe the design of our approach, Thor. We evaluate the effectiveness and the computational cost in Section 6. presents a brief description of the datasets and classification Finally, we discuss future directions in Section 6. Finally, Section 7 presents a discussion and future research directions.

## II. Related works

Many service providers and password meters enforce users to follow password policies in order to nudge users to create a hard-to-crack password (Golla & Dürmuth, 2018) [9]. A password meter that either guide or enforce users to comply a stronger password at the moment of creation. For instance, require meeting the minimum password policy. Such as, including a minimum number of characters or special characters and etcetera (Carnavalet & Mannan, 2015) [1]. At the moment of creation, many studies and applied research on the human-computer interaction are pointed to influence users to create a stronger password. As well, many approved practical solutions have a considerable effect on motivating users to create a stronger password.

Nonetheless, many users tend to have password habits that the cybersecurity does not recommend for many reasons; either to create a memorable password or to not updating their password regularly (Yıldırım & Mackie, 2019) [31]. Furthermore, most frequently used websites fail to either oblige, encourage or influence users to follow the right path to change these habits that the cybersecurity does not recommend in the last decade (Furnell, 2018) [4].

On the other hand, involving human psychological behaviour to computer interaction on nudging the end-user, or actively providing rich interactive feedback (at the moment of password creation) had a significant effect on password strength and user awareness (Furnell et al., 2018) [6]. Equally on the same involvement, using an emoji-base with the text feedback has a tangibly better performance on the password length and the average password strength score [5].

Furthermore, on investigating the user's behaviour of choosing a weak and short password to be easy to memorise [29] argue that the well understanding of a user's memory perception would increase the password memorability. Woods and Siponen [29] have concluded that users can memorise passwords more by understanding the memory perception. In a like manner in a trade-off between password memorability and user inconvenience. Woods and Siponen [29] experimented that the user's password recall and memorability have increased from 40% to 70% by verifying the password twice to verifying the password three times, without convincing users.

In the light of enhancing the end-user text feedback to study the interactive text-feedback effect on the created password strength, Seitz et al. [21] and Ur et al. [26] have concluded that the combination of enriching an actionable text feedback, various password meter, various data-driven guidance of explaining what is wrong with the inputted password and some examples of enhancing the week password parts. This guidance enhanced the inputted password and nudged the end-user to create a stronger scale password. Equally important on the password strength evaluation methods and usability. Shay et al. [24] and Segreti et al. [20] recommended that the password-meter base policy that combines special letters with a longer-length. This recommendation will lead to a fewer guessed and a more usable password — this built-in recommendation comparison with enforcing the end-users to follow a shorter length requirement with a comprehensive policy.

According to Yiannis [30] and Galbally et al. [7], [8] on highlighting the password strength and its time to crack, several open-source tools and systems running in a general machine could crack any password in less than an hour. This case is possible in case of existing this password on any dataset, for instance, password dictionaries.

Spotlighting the password dictionaries files and its efficiency role on reducing the password cracking-time. There are many New Password Cracking (NPC) techniques that are continually working to improve dictionary attacks involving Artificial Intelligence (AI). This technique reduces the searching time within its billions of records Houshmand et al. [14].

Involving the dictionaries as a criterion for accepting password (blacklist password), Habib et al. [12] studied the user interaction on rejecting his blacklisted attempted passwords. Habib et al., found that the text feedback has more effectiveness than rejecting users blacklisted password. As well, on nudging users to create a stronger password through a different approved technique, Furnell et al. [6] induced users to create a stronger password by adding contextual information, at the same time, tweaking the displayed warning messages to the inputted password generating behaviours.

To shed a light on password meters accuracy and infer the password strength calculation algorithms, in a purpose to clear the user confusion while choosing a stronger password, Carnavalet and Mannan [1] proposed an accurate meter that measures the password strength, this meter has a different calculation algorithm depending on many factors (blacklisted passwords, commonly used passwords and the additional regular password mark by the regular meters).

Equally important, Guo and Zhang [10] produced an accurate identification Lightweight Password Strength Estimator (LPSE), that requires a small storage space at the client-side (33 kilobits) and secure online service integration steps, LPSE depends on many factors on password evaluation, this evaluation is using the most advanced password-cracking algorithm (0.181 ms running time). Also, it produces less false-negative rate on both weak and secure passwords.

Spotlighting various available searching algorithms could be used in a large dataset. TRIE & RABIN-KARP algorithms have been approved to support large dataset (billion records) searching time. This hybrid algorithm gained searching results in seconds, according to Luo et al. [18], Gupta [11], and Liu et al. [17]. Equally important on string match search, the

RABIN-KARP algorithm approved its time and text efficiency on comparing strings on a large dataset [16], [22], [23].

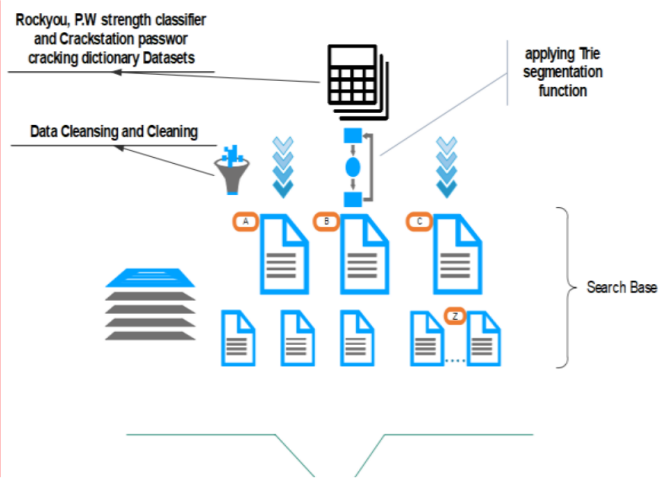


Fig. 1. The proposed meter leaked password search function

## III. Datasets

This research is conducted using extensive and publicly available datasets of compromised and leaked passwords. In this part, we describe each of these datasets and summarize them in Table I.

**RockYou Dataset.** This dataset contains over 32 million compromised passwords that were leaked using an SQL injection attack that targeted RockYou web service in 2009. This dataset is considered a goldmine due to the fact that RockYou did not hash the passwords prior to the attack. In fact, the leaked passwords were stored unencrypted in plaintext format when attack happened. After removing duplicated passwords, this dataset provided 14,341,564 unique plaintext passwords.

**000Webhost Leak.** A free web space provider, 000Webhost, was hacked in 2015 by a cyber attack that exploited a software vulnerability in an outdated PHP version. This attack leaked 13 million plaintext passwords. The passwords in this dataset are generally stronger than the RockYou dataset, as 000Webhost had a stronger minimum password policy that enforced composing lowercase and digits passwords. This dataset contains 714,173 unique plaintext passwords.

**CrackStation Dictionary.** This massive dictionary stores a mapping between the hash of a password and the plaintext password for that hash. The dictionary indexes these hashes to optimize searching the dictionary for a given hash to recover its plaintext password. If the hash of a given password is stored in this dictionary, it takes less than a second to look-up its plaintext password. This dictionary contains 15 billion entries for MD5 and SHA1 hash functions, and 1.5 billion entries for other hash functions.

## IV. Design and Implementation

In this section, we explain the design and steps to implement and apply the proposed password meter.

TABLE I
Summary of Datasets

| dataset | year | password policy | number of unique passwords |
|---|---|---|---|
| RockYou | 2009 | 5+ | 14,341,564 |
| 000Webhost | 2016 | 6+ [a-Z][0-9] | 714,173 |
| CrackStation | 2019 | 5+ | 64,000,000 (human passwords only) |

### A. Passwords Collection and Preprocessing

If a password was leaked, then it does not matter if it is strong as it will be cracked using dictionary attack or wordlist based bruteforce preprocessing so first, we search leaked databases, for this, we combined the three datasets mentioned above into a massive database of leaked passwords second, searching such dataset is a time consuming process. To reduce the search time (look-up) for this step, we performed the following steps:

1) Removed all duplicated passwords. In this step, we retained only unique passwords and discarded duplicated passwords. This reduced the datasets into the following numbers.
2) Removed overlong passwords. Strange lengthy password entries that are larger than 256 characters were removed to maintain dataset with realistic human provided passwords.
3) Removed noisy entries. Some entries were words such as ”N/A” or a separating character such as a single comma or semi-column ”,” or ”;”. We decided that such entries are noise or misreadings that were introduced during some part of data processing, and therefore, we removed such entries.

### B. Scoring Passwords Strength

We calculate the strength score for a provided password in two steps: first, we identify phrases that are found in leaked password databases, and then we apply measuring techniques to provide the most accurate score for password strengths.

*1) **Identifying Compromised Passwords**:* A chosen password such as `Abcde123@` or `P@ssword1` is not considered weak because of its characters nor complexity. In fact, they are considered weak and insecure because they have been chosen many times and are present in leaked password datasets. Such passwords are used early in the password cracking process by cyber attacks [27].

To identify common and leaked passwords, our approach performs a comprehensive look-up in the collected databases RockYou, 000Webhost, and CrackStation. If a chosen password by the user is found in the leaked databases, our approach will directly label is as ŕery weak.

Searching for a password (string) in large databases could be a time consuming process. To provide an enhanced look-up function that searches the three large datasets for a chosen password, we developed a search function that combines the

Trie search tree with the RABIN-KARP algorithm [3]. We explain each next.

**RABIN-KARP**. This string searching algorithm utilizes hashing to find string patterns in text [3]. It works by calculating the hash values for string patterns of length M to be compared with a given string. By using this method, only one comparison per string pattern (or sequence of characters) occurs. The intuition behind this algorithm is that it preprocesses the string pattern with M length (m sequence of characters) to search that number (m) in the text and find an exact match of that pattern.

**Trie**. This data structure can be visualized as a graph. This graph starts with a root node that has 26 outgoing edges (one for each letter in the English alphabet). First, strings are stored in this structure in a top to bottom manner to associate the length if a given string (word) to a specific level in the Trie. This type of storage allows for efficient retrieval of strings by traversing down a path of the tree. The time complexity to do this top-down search depends on the string length that is being searched for such that its runtime is O(n) where n is the number of characters in the string.

It is noteworthy that in this work, to search for a provided password in the three large datasets. First, apply the Trie segmentation function on the three datasets. We implement this segmentation to fragment the datasets in an alphabetical manner. Then, we use RABIN-KARP on the unified database segment that contains first letter of the provided password. By splitting the database that combines the three leaked password databases, into segments based on the first letter, we consumed more space but achieved better retrieval times. Figure 1 illustrates this process.

**Optimization for the second search.** If a user chooses a password that is present in these databases, our system will alert and influence the user to change the choose a different password. For this reason, a second search is needed to look up the new chosen password if the user decides to so. To further enhance the performance of the second search process, for the new password, we took advantage of the work of Zhang-Kennedy et al. [32] which discovered that 50% of users reuse the first letter of their previous password when they create a new one. To take advantage of this finding, we retained the searching index result of the first character in the user's password. By using this technique, if the user changes her password, to enhance its security, our approach will be able to perform the search operation of the second password at much faster rate. To keep the memory from being depleted by our system, we cleared this searching index result 10 seconds after the user chooses a new password.

*2) **Measuring Password Strength**:* Traditional approaches measure the entropy –a metric in information theory that measures randomness in a given string (or password). Such a metric might be misleading because its heavily affected by common English words from the dictionary. In our approach, such words will be identified and filtered by the previous step that searches the three large datasets mentioned earlier. For this reason, we take advantage of previous works which use the number of guesses that the attacker would take to crack the password. Therefore, We utilize the implementation of Daniel Lowe password strength estimator, zxcvbn [28]. This implementation measures the strength of a given password by calculating the attacker's worst case scenario that has chosen the best available approach to crack the password (later referred to as Time-to-Crack). This makes it a natural choice for our work as it provides the time to crack a provided password (in days, months, or years).

### C. VISUAL DESIGN of the Password Meter

In this section, we describe our visual design of the password meter. The meter's screen consists of three key components. Figure x illustrates the components of the proposed visual design of the meter. We explain each component next.

- **Sliding Bar (slider)**. This bar displays the password strength as a sliding scale where if the slider is at most-left, it means that is is the weakest where most-right means the strongest password.
- **Text Feedback**. The meter screen displays the time-to-crack metric, which explains to the user the intuition behind the the visual bar (slider) that represents the strength of the password. This type of feedback provides an important detail which shows the strength of the password from an attacker's point-of-view. Showing the average time duration (hours to years) that would take the cyber attack to crack the created password is an intuitive method to influence users to choose stronger password.
- **Visual Feedback (colors and emojis)**. To increase the usability of the meter and support the older and younger audience, the meters slider is highlighted with various colors to represent the strength of the password (e.g., the green color means a strong password, where red means a weak password). Finally, the slider itself will be changed to an emoji that represents the strength pf the created password. A smiley-face emoji means a strong password, and a face-screaming-in-fear emoji (most left in Figure x) means a weak password was chosen.

## V. User Survey

To study the effectiveness of the proposed password meter to inform users about the strength of their chosen password and influence them to create stronger ones, we designed a user study and ran experiments with the participation of real users. This user study comprises of three steps: (1) determining the appropriate sample size for the experiments, (2) using eye-tracking technology to evaluate our meter's ability to capture users' attention, and (3) the effectiveness of our meter to influence users to create stronger passwords. We explain each of these steps next.

### A. Determining the Sample Size

Choosing a sample size that is appropriate for a given user study is not a straightforward process. Determining a sample size based on the size of the population has been extensively discussed in the literature [2], [13], [15], [19], [25]. We use

the approach proposed by Stallard et al. [25] to determine our optimal sample size ($n$) as shown by Equation 1.

$$n = \frac{Z^2 p(1-p)}{e^2} \quad (1)$$

where $Z$ is the confidence level at 95% (standard value of 1.96). P is the estimated proportions of the study area. E is range of confidence interval.

By using Equation 1, the appropriate sample size for this study was calculated and it is 210 users.

We recruited 210 participants from the University of Sunderland for our study on password strength meter. All participants who participated in this study were age 18+.

### B. Eye-Tracking: Capturing Users' Attention

Previous work showed that capturing the attention of users is crucial for password meters to be effective at providing any subsequent feedback. To this end, we investigate the effectiveness of our proposed meter to capture the participant's attention using eye tracking technology.

In this experiment, Tobii Studio version 3.3.2 eye tracker was used. This eye tracking technology keeps track of users' eye movements and uses sensors to detect the focus points of the user's eyes on the screen content. The output of each session of this experiment is a color-coded heatmap. This heatmap represents the eye focus points on the screen content using three colored main colors: Green, Yellow, and Red. Green bubbles are used in situations when the eye focus of the user was minimal. Yellow bubbles are used for moderate eye focus, and the red bubbles represent high eye focus. We present and discuss the results of these experiments in the next section.

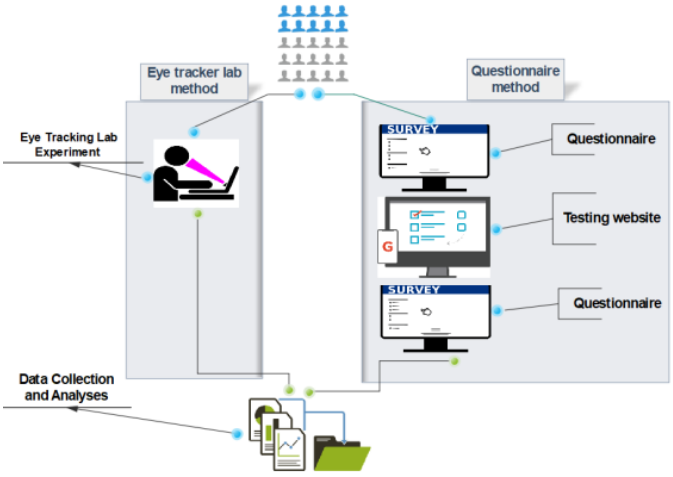


Fig. 2. Evaluation methodology

## VI. Evaluation

To obtain more accurate evaluation of of the strength of user created passwords and the influence of our proposed meter, first, we need to measure the effectiveness of our proposed meter to capture the attention of our users. Second, we need to measure the influence of our meter in terms of the number of users who changed their passwords to be more secure. Figure 2 illustrates the evaluation methodology.

### A. Effectiveness of Capturing Attention

Our first set of evaluation results show the effectiveness of our proposed meter to capture the focus of participants when they create passwords. Capturing users' attention is the first step towards influencing users to create stronger passwords. Failing to capture users' attention renders any subsequent steps (e.g., providing feedback) pointless due to the fact that users did not see them.

Therefore, we ran the eye tracking experiment using two password meters: our proposed meter and a regular meter (inspired by the work in [5]). Then, we compared the eye focus heatmaps between these two password meter. Figures 3 and 4 show the heatmaps for using our proposed meter and the regular meter, respectively.

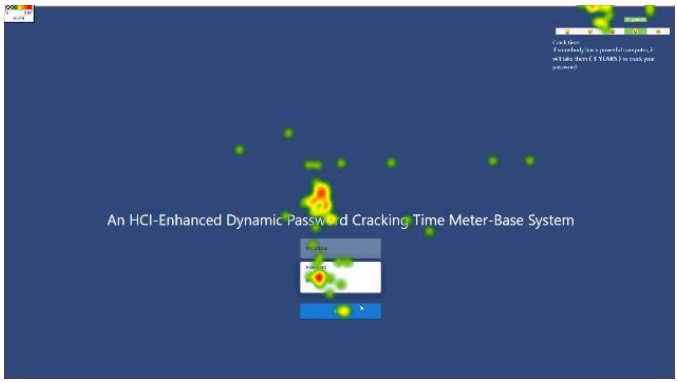

Fig. 3. The cumulative user eye focus when using the proposed password meter proposed algorithm

Fig. 4. The cumulative user eye focus when using the regular password meter

As shown by the Figure 3 and 4, our meter achieved a better effectiveness at capturing users attention to both the meter visual and the textual feedback on top right corner of the webpage.

### B. Measuring the influence on changing passwords

The measured effectiveness of our meter to capture users' attention is a positive indicator of the overall effectiveness

of our proposed password meter. However, a key metric to evaluate password meters is to measure the percentage of users who actually changed their passwords based on the presented feedback of the password meter. To this end, we captured two data points: (1) the users' created passwords before presented with the feedback of the meter, and (2) the new chosen passwords after presented with the proposal meter's feedback.

TABLE II
THE IMPACT OF OUR PROPOSED METER ON THE STRENGTH OF USERS' PASSWORDS (BEFORE & AFTER THE METER'S FEEDBACK)

| | Group 1 | Group 2 | Group 3 |
|---|---|---|---|
| Number of users | 13 | 195 | 2 |
| Average password cracking time difference | 121.049 years | 108.37 years | -0.46 years |
| Gained time-to-crack (in percentage) | 333% | 902% | -901% |

We grouped the participants in this experiment into three groups: Group 1 is comprised of users who had prior knowledge and understanding of the time-to-crack for given passwords. The participants in Group 2 had no prior knowledge of the time-to-crack for given passwords. The two participants in Group 3 had partial knowledge about time-to-crack for given passwords.

As shown by Table II, the participants in group 1 (with prior knowledge about time-to-crack) changed their passwords after seeing the feedback of our password meter. The average time-to-crack for their passwords increased from 121.04 to 172.85 years which translates to 333% stronger passwords in terms of years required by cyber attacks to crack these passwords.

Participants with no prior knowledge about time-to-crack (Group 2) showed an outstanding improvement for their created passwords after seeing the feedback of our password meter. The average time-to-crack for their passwords increased from 13.51 to 121.88 years which translates to passwords that are 902% stronger in terms of years required by cyber attacks to crack.

Group 3 is comprised of 2 participants with partial knowledge about the time-to-crack. This group showed a decline of 901% for their password strengths after seeing the meter's feedback. The average time-to-crack decreased from 0.419 years (5 months) to 0.046 years (0.55 months or 16.7 days). Unlike the groups 1 and 2, group 3 performed worse at creating passwords after seeing the feedback of our password meter. We believe the participants in this group provided abnormal entries (or outliers). The main reason for this belief is that this group created weak passwords in the first attempt (prior seeing any feedback from our meter) despite group answers of understanding the time-to-crack concept which is an anomaly. Then, in the second attempt, group created much weaker passwords despite the fact that our password meter displayed the face-screaming-in-fear emoji.

## VII. DISCUSSION

The experiments in this study show that our password meter takes an important step to detect, warn, and educate users about password creation and password strength importance to authentication-based security.

**Target groups.** Our evaluation shows that the proposed meter influenced a large number of participants to create significantly stronger passwords. However, based on the user study, participants who did not have prior knowledge about the time-to-crack concept and how it related to the security of authentication-based systems, showed an outstanding improvement that translated into 900% improvement of chosen passwords. Therefore, we believe that the proposed meter performs very well at educating and influencing users with little or no prior knowledge about the time-to-crack concept.

**Error/Mis-influence analysis.** Surprisingly, participants with partial knowledge about the time-to-crack concept, performed worse after seeing the proposed meter's feedback. While these entries could be outliers, however, this decline might be due to misconceptions users have about the time-to-crack. If this is the case, then removing prior misconceptions of participants could an area of improvement for the proposed meter.

## VIII. FUTURE WORK

Data-driven password meters depend profoundly on searching datasets of leaked passwords. In this work, we combined three large datasets into one database and used an innovative search function to perform rapid look-up operations for leaked passwords. One future direction to improve on our approach is to gather more datasets of leaked passwords and English words from websites such as KAGGLE and STATISTA, and then test the performance of the search functions on big data. An innovative integration of users' password habits and big data (for password look-ups) is a promising direction for future research.

As mentioned earlier, we used the Tobii Studio version 3.3.2 for the eye-tracking experiments. There are various and more up-to-date tools that provide the state-of-the-art eye-tracking technology that may provide more accurate results for measuring users' attention, not only on workstations, but also, on tables, mobile phones and other electronics. Such research becomes a necessity with the rapid increase of using mobile devices for account and password creation.

## IX. CONCLUSION

The future will have many alternative authentication methods alongside passwords, but passwords will always be there; users are required to use the type of authentication method that was chosen by the system developers, and today passwords are the most common method used. The method used in this paper investigated the percentage of the improvement in the number of influenced users while creating their password. We then analysed the resulted inputted password on our real web service system. As well we have contributed a new searching algorithm to search in a large dataset to be used as a first checkpoint of the end-user inputted password. As a result, we have concluded that our contributed meter has influenced 88% of our participants to create passwords that take in average 150 years to crack.

## REFERENCES


[1] X. D. C. D. Carnavalet and M. Mannan. A large-scale evaluation of high-impact password strength meters. *ACM Transactions on Information and System Security (TISSEC)*, 18(1):1–32, 2015.

[2] Y. Cheng, F. Su, and D. A. Berry. Choosing sample size for a clinical trial using decision analysis. *Biometrika*, 90(4):923–936, 2003.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2009.

[4] S. Furnell. Assessing website password practices–over a decade of progress? *Computer Fraud & Security*, 2018(7):6–13, 2018.

[5] S. Furnell and R. Esmael. Evaluating the effect of guidance and feedback upon password compliance. *Computer Fraud & Security*, 2017(1):5–10, 2017.

[6] S. Furnell, R. Esmael, W. Yang, N. Li, et al. Enhancing security behaviour by supporting the user. *Computers & Security*, 75:1–9, 2018.

[7] J. Galbally, I. Coisel, and I. Sanchez. A new multimodal approach for password strength estimation—part ii: Experimental evaluation. *IEEE Transactions on Information Forensics and Security*, 12(12):2845–2860, 2017.

[8] J. Galbally, I. Coisel, I. Sanchez, Z. Zhu, Z. Chu, N. Wang, S. Huang, Z. Wang, I. Lee, A. Papadopoulos, et al. A new multimodal approach for password strength estimation—part i: Theory and algorithms.....

[9] M. Golla and M. 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.

[10] Y. Guo and Z. Zhang. Lpse: lightweight password-strength estimation for password meters. *computers & security*, 73:507–518, 2018.

[11] S. Gupta. Optimized text data processing. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 1000–1004. IEEE, 2018.

[12] H. Habib, J. Colnago, W. Melicher, B. Ur, S. Segreti, L. Bauer, N. Christin, and L. Cranor. Password creation in the presence of blacklists. *Proc. USEC*, page 50, 2017.

[13] S. W. Hee, T. Hamborg, S. Day, J. Madan, F. Miller, M. Posch, S. Zohar, and N. Stallard. Decision-theoretic designs for small trials and pilot studies: a review. *Statistical methods in medical research*, 25(3):1022–1038, 2016.

[14] S. Houshmand, S. Aggarwal, and R. Flood. Next gen pcfg password cracking. *IEEE Transactions on Information Forensics and Security*, 10(8):1776–1791, 2015.

[15] T. Kikuchi and J. Gittins. A behavioral bayes method to determine the sample size of a clinical trial considering efficacy and safety. *Statistics in medicine*, 28(18):2293–2306, 2009.

[16] B. Leonardo and S. Hansun. Text documents plagiarism detection using rabin-karp and jaro-winkler distance algorithms. *Indonesian Journal of Electrical Engineering and Computer Science*, 5(2):462–471, 2017.

[17] Y. Liu, J. Xu, X. Gong, J. Bai, L. Xiao, H. Zhu, C. Tan, and T. Lou. Ternary search trie based algorithms for recognizing the names of power devices. In *Journal of Physics: Conference Series*, volume 1213, page 032025. IOP Publishing, 2019.

[18] Y. Luo, G. H. Fletcher, J. Hidders, and P. De Bra. Efficient and scalable trie-based algorithms for computing set containment relations. In *2015 IEEE 31st International Conference on Data Engineering*, pages 303–314. IEEE, 2015.

[19] H. Pezeshk, N. Nematollahi, V. Maroufy, P. Marriott, and J. Gittins. Bayesian sample size calculation for estimation of the difference between two binomial proportions. *Statistical methods in medical research*, 22(6):598–611, 2013.

[20] S. M. Segreti, W. Melicher, S. Komanduri, D. Melicher, R. Shay, B. Ur, L. Bauer, N. Christin, L. F. Cranor, and M. L. Mazurek. Diversify to survive: Making passwords stronger with adaptive policies. In *Thirteenth Symposium on Usable Privacy and Security ({SOUPS} 2017)*, pages 1–12, 2017.

[21] T. Seitz, E. von Zezschwitz, S. Meitner, and H. Hussmann. Influencing self-selected passwords through suggestions and the decoy effect. In *Proceedings of the 1st European Workshop on Usable Security*, pages 1–2, 2016.

[22] P. Shah and R. Oza. Improved parallel rabin-karp algorithm using compute unified device architecture. In *International Conference on Information and Communication Technology for Intelligent Systems*, pages 236–244. Springer, 2017.

[23] J. Sharma and M. Singh. Cuda based rabin-karp pattern matching for deep packet inspection on a multicore gpu. *International Journal of Computer Network and Information Security*, 7(10):70–77, 2015.

[24] R. Shay, S. Komanduri, A. L. Durity, P. Huh, M. L. Mazurek, S. M. Segreti, B. Ur, L. Bauer, N. Christin, and L. F. Cranor. Designing password policies for strength and usability. *ACM Transactions on Information and System Security (TISSEC)*, 18(4):1–34, 2016.

[25] N. Stallard, F. Miller, S. Day, S. W. Hee, J. Madan, S. Zohar, and M. Posch. Determination of the optimal sample size for a clinical trial accounting for the population size. *Biometrical Journal*, 59(4):609–625, 2017.

[26] B. Ur, F. Alfieri, M. Aung, L. Bauer, N. Christin, J. Colnago, L. F. Cranor, H. Dixon, P. Emami Naeini, H. 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.

[27] D. Wang, D. He, H. Cheng, and P. Wang. fuzzypsm: A new password strength meter using fuzzy probabilistic context-free grammars. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 595–606. IEEE, 2016.

[28] D. L. Wheeler. zxcvbn: Low-budget password strength estimation. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 157–173, 2016.

[29] N. Woods and M. Siponen. Too many passwords? how understanding our memory can increase password memorability. *International Journal of Human-Computer Studies*, 111:36–48, 2018.

[30] C. Yiannis. Modern password cracking: A hands-on approach to creating an optimised and versatile attack. *Info. Security Grp.*, pages 5–6, 2013.

[31] M. Yıldırım and I. Mackie. Encouraging users to improve password security and memorability. *International Journal of Information Security*, 18(6):741–759, 2019.

[32] L. Zhang-Kennedy, S. Chiasson, and P. van Oorschot. Revisiting password rules: facilitating human management of passwords. In *2016 APWG symposium on electronic crime research (eCrime)*, pages 1–10. IEEE, 2016.