Journal of Internet Services and Applications, 2025, 16:1, doi: 10.5753/jisa.2025.5041

From RockYou to RockYou2024: Analyzing Password Patterns Across Generations, Their Use in Industrial Systems and Vulnerability to Password Guessing Attacks

Gabriel Arquelau Pimenta Rodrigues [**University of Brasilia** | ***gabriel.arquelau@redes.unb.br***]
Pedro Augusto Giacomelli Fernandes [**University of Brasilia** | ***pedro.giacomelli2@gmail.com***]
André Luiz Marques Serrano [**University of Brasilia** | ***andrelms@unb.br***]
Geraldo Pereira Rocha Filho [**State University of Southwest Bahia** | ***geraldo.rocha@uesb.edu.br***]
Guilherme Fay Vergara [**University of Brasilia** | ***guilherme.vergara@redes.unb.br***]
Guilherme Dantas Bispo [**University of Brasilia** | ***guilherme.bispo@redes.unb.br***]
Robson de Oliveira Albuquerque [**University of Brasilia** | ***robson@redes.unb.br***]
Vinícius Pereira Gonçalves [**University of Brasilia** | ***vpgvinicius@unb.br***]

Cyber Security INCT Unit 6, Laboratory for Decision-Making Technologies (LATITUDE), Department of Electrical Engineering (ENE), Faculty of Technology, University of Brasília (UnB), Brasília, DF, 70910-900.

Received: 03 October 2024 • **Accepted:** 27 February 2025 • **Published:** 08 April 2025

Abstract Passwords are a common user authentication method, and must be safeguarded by effective security measures. However, there are many cases of compromised user credentials in data breaches. This work studies RockYou2024, a massive data breach that occurred in July 2024 and exposed over 9 billion passwords. We investigate the passwords with regard to their lengths, entropy, use of personal information and common strings, and evaluation from `zxcvbn`, as well as making a comparative assessment of the results with previous password databases, namely RockYou2021 and RockYou, which was leaked in 2009. This analysis found that the passwords from RockYou2021 and RockYou2024 are significantly more secure than those from RockYou, which suggests an improvement in password creation awareness and policies. It was also noted that RockYou2021 and RockYou2024 have similar statistical distributions in all the analyses conducted. We have also found that the country of origin for most passwords within these databases is most likely to be the United States of America. These datasets were searched for passwords that are often used in industrial systems, which pose potential security risks in critical infrastructure sectors. Finally, we also propose passBiRVAE, a contextualized Bidirectional Recurrent Neural Network , used to generate passwords based on the RockYou2024 database. Future works should make further improvements to the results obtained from this model. However, there is a risk of threats to the validity of these analyses.

Keywords: Authentication, autoencoder, cybersecurity, data breach, password, privacy, and RockYou

1 Introduction

Despite the growing trend toward passwordless systems [Parmar *et al.*, 2022], many current applications still rely on password-based authentication. However, when creating a memorable password, many users often develop bad habits by giving priority to simplicity over security, even in professional contexts, where they may be responsible for handling sensitive data [Styoutomo and Ruldeviyani, 2023]. Common mistakes include using easily guessable information, such as names, dates of birth and everyday words like `password`, or obvious numerical sequences, such as `123456`. Furthermore, many users create short passwords and fail to include a combination of upper and lower case letters, numbers and symbols, which further weakens the security of these credentials.

It should also be noted that email addresses and passwords are the most frequently compromised types of data in information breaches [Mayer *et al.*, 2021], reinforcing the importance of avoiding password reuse and regularly updating passwords. Reusing passwords increases the risk of multiple accounts being compromised, especially when this kind of data is exposed to large-scale attacks.

These concerns become more critical in industrial systems, used in sectors that are considered critical infrastructure, such as energy generation and transmission, due to their importance and the innovative techniques that form a part of Industry 4.0 [Bispo *et al.*, 2024]. One of the vulnerabilities of these systems is their inadequate authentication [Upadhyay and Sampalli, 2020]. In view of this, it is essential to strengthen authentication practices in these environments to protect critical industrial systems against cyber threats.

An analysis of leaked password databases has provided information about common password creation methods. Moreover, this type of study allows forensic investigators to understand user behavior when creating their credentials, such as ways of forming frequent or predictable combinations. This knowledge can be used both to improve password cracking techniques during criminal investigations [Kanta *et al.*, 2021; Bichara *et al.*, 2023] and to assist in the development of more effective security policies [Siponen *et al.*, 2020].

In this work, we investigate the main weaknesses that were discovered in passwords from the RockYou2024 leak, which occurred in July 2024 [Petkauskas, 2024]. Our study compares the results obtained from previous versions of this leak, namely RockYou and RockYou2021, and thus is able to detect changes in user behavior and possible technical advances that add to password complexity. The analysis includes password entropy, the use of personal information and common substrings, with particular attention being paid to passwords used in industrial systems. In addition, we estimated how many MD5 hashes in each database can be cracked by means of a simple word list.

Another key factor in this study is the proposal of a new model, passBiRVAE, a contextualized Bidirectional Recurrent Neural Network for password generation, based on the RockYou2024 leak. Our model incorporates valuable information, such as substrings and common combinations, to improve the understanding of such patterns.

1.1 Contributions and limitations of this work

This research analyzes password patterns in the RockYou2024 database, comparing the results with RockYou2021 and RockYou. One of its key contributions is assessing differences in these password patterns, demonstrating how user behavior regarding password selection has evolved.

RockYou2024 has only recently been breached, so there is limited research on it. This work contributes by analyzing its patterns and comparing them to previous datasets. This work indicates the current patterns of password use, thereby identifying common weaknesses. Thus, this study presents findings that can aid digital forensic investigators and guide the development of effective password policies, including for application in industrial systems.

Also, the exact origin of the passwords is not known. By analyzing the probable country of origin of personal names used in passwords, we can estimate the geographic distribution of the affected users. This analysis suggests that, similar to the original RockYou breach, the RockYou2021 and RockYou2024 compilations are also likely composed predominantly of passwords from users in the United States.

Additionally, since RockYou2024 is composed of many non-password values, this work proposes a methodology for filtering these lines out, enabling future researchers to use a cleaner database version.

We have also shown that RockYou2021 and RockYou2024 have negligible differences in their distributions of characteristics like passwords length, entropy, expectation entropy and cracking time. This similarity implies that researchers may use RockYou2021 instead of RockYou2024 for studies requiring a cleaner dataset, as it exhibits comparable statistics and less non-password rows. However, a limitation of this work is that we have not determined whether the passwords are duplicated in both these databases or if the observed similarity arises from minimal changes in user patterns due to the short time span between them.

Moreover, we searched for improvements in trawling attacks based on deep neural networks to evaluate the datasets' consistency and possible improvements in the field. This is the first study to use RockYou2024 as a training dataset for deep neural networks. However, the results require more investigation.

While the findings suggest a noticeable improvement in password security between RockYou and the more recent RockYou2021 and RockYou2024 datasets, it is challenging to attribute this change to a single factor. The observed enhancements could be a consequence, for example, of an evolution in the users' cultural habits or of stricter password creation policies enforced by systems. The lack of detailed contextual data for the passwords makes it difficult to discern the relative influence of these factors.

As another limitation, this work's observations are strictly related to these databases and may not reflect the natural pattern of password use worldwide or in other databases.

Furthermore, despite the cleaning methodology filtering out approximately 14% of the RockYou2024 database, results show that strings with non-password characteristics, such as 231-character long lines, are still present in the file.

1.2 Structure of this work

The remainder of this work is organized as follows. Section 2 discusses the related literature. Section 3 describes the methodology used in this work. Section 4 presents the password patterns observed in each RockYou database, whereas Section 5 shows the MD5 cracking results. Section 6 analyzes the results obtained from the autoencoder network and the proposed modification. Section 7 details the threats to the validity of our findings and Section 8 concludes the paper and proposes future works. Tables 11 and 12 summarize the acronyms and variables, respectively, mentioned in this paper.

2 Literature review

To prevent the leak of personal information, including authentication credentials such as passwords, it is fundamental to strengthen the security controls, both technical and administrative. Rodrigues *et al.* [2024] discusses mitigation measures aligned with established guidelines like the NIST Cybersecurity Framework, whereas Pimenta Rodrigues *et al.* [2024] focuses on the regulatory aspects of data protection. The efficient implementation of these security strategies may reduce the impact and the likelihood of future password breaches.

Kanta *et al.* [2024] shows that law enforcement may explore a suspect user's insecure password to obtain information potentially useful for the investigation. The authors indicate that targeted approaches, when used in combination with traditional strategies, increase the likelihood of success of the digital forensic process.

2.1 Password patterns

Since 2009, the RockYou wordlist was leaked, several works have used these passwords as the foundation of their investigations [Bojinov *et al.*, 2010; Weir *et al.*, 2010]. More recently, Xu *et al.* [2023] presented PassBERT, a bi-directional transformer framework designed for enhancing password guessing attacks using natural language processing tech-

Table 1. Comparison of this study with previous works

Reference	Passwords patterns	Attack models	RockYou	RockYou2021	RockYou2024
[AlSabah *et al.*, 2018]	✓	✓	✗	✗	✗
[Tatlı, 2015]	✓	✗	✓	✗	✗
[Jiang *et al.*, 2022]	✗	✓	✓	✗	✗
[Lykousas and Patsakis, 2023]	✓	✗	✗	✓	✗
[van den Berg, 2024]	✗	✗	✗	✗	✓
[СЛАТВІНСЬКА and БЕВЗА, 2024]	✗	✗	✗	✗	✓
This work	✓	✓	✓	✓	✓

niques, outperforming other state-of-the-art approaches. It uses RockYou2021 as a pre-training dataset.

One of the works that references RockYou2024 is that of СЛАТВІНСЬКА and БЕВЗА [2024], which explores the possible relationship between this password breach and the CrowdStrike security incident. Their research revealed a need for further investigation to conclude the impact of the incident on the password leak. To our knowledge, our work is the first academic paper that studies the RockYou2024 wordlist patterns and compares them to the previous versions of the leak.

Other password databases, however, have also been studied. Such leaks include a 2016 password database from a Middle Eastern bank [AlSabah *et al.*, 2018], XATO [Zhang *et al.*, 2021] and Weakpass [Biesner *et al.*, 2021]. Dubey and Martin [2021] have created a dataset with labeled credential data with its source and leak date. Works have also focused on password hash lists, for example, from hashes.org [Nisenoff *et al.*, 2023] and from Have I Been Pwned [Pal *et al.*, 2022].

In particular, AlSabah *et al.* [2018] examined the cultural and linguistic patterns that influence password choices. Their research investigated the commonly used security questions and the demographics that exhibited specific password behaviors, such as incorporating personal names and telephone numbers into passwords.

Although using such weak patterns in passwords seems obsolete, Lee *et al.* [2022] observed that 75% of the 120 popular websites they studied allowed users to configure common vulnerable passwords.

To assist users in creating secure passwords, Password Strength Meters (PSM) are commonly used tools designed to provide feedback on their credentials. These tools are usually based on the length and prerequisites based on the presence of lowercase and uppercase letters, digits and symbols (LUDS). In this respect, Thai and Tanaka [2024] proposed a PSM based on Markov models to enhance password creation with more comprehensive contributions. NIST SP 800-63B provides useful information for password creators and verifiers.

2.2 Autoencoders

Autoencoders are used to generate new data. They provide interpretable results, since their latent space can be organized to learn representations of data [Kingma, 2013; Higgins *et al.*, 2017]. Therefore, the Variational Autoencoder (VAE) can generate new passwords using its learned features and Generative Adversarial Networks (GANs) [Yu *et al.*, 2023]. Using this concept, Yang *et al.* [2022] proposed a VAE that receives the password and teaches the network to reconstruct it, developing the latent space to generate data. Since a password consists of a word, which may be more meaningful when analyzed sequentially, Xiao [2024] proposed a Recurrent VAE to process this characteristic of the data. Wu *et al.* [2023] explores a similar concept but using latent convolutional neural networks to explore the patterns that may appear in the password, taking into consideration a more comprehensive range of the password instead of looking for it locally (individual characters as the recurrent model did). In addition, Biesner *et al.* [2022] developed a new model that applies Transformers to explore the local meaning of characters and how it can influence the generation of passwords.

Yang and Wang [2024] have conducted a study on password guessing and proposed a new framework, RANKGUESS, based on adversarial ranking and using RockYou as the training data. They modeled the password creation process as sequential decision trajectories and used adversarial techniques to improve guessing accuracy. Their model improves password cracking success rates by 7.70% to 14.85%.

2.3 Comparison with other works

Table 1 compares the current study with previous works, with a focus on our contributions. Other works, like van den Berg [2024]; СЛАТВІНСЬКА and БЕВЗА [2024], mention the RockYou2024 leak, but did not analyze its patterns nor proposed a password guessing model.

Therefore, to the best of our knowledge, this is the first work to study the RockYou2024 breach, compare the patterns in passwords creation across RockYou, RockYou2021 and RockYou2024 and also the first to use RockYou2024 as the training dataset for a password guessing model.

3 Methodology

This work uses the RockYou, RockYou2021 and RockYou2024 password databases to investigate password creation patterns and analyze the use of personal information in these datasets. Studying these password databases is important to understanding common password selection practices and how personal information, such as names, dates of birth and others, can be incorporated into these choices, directly impacting digital security. However, these databases are not composed exclusively of unique passwords, as they

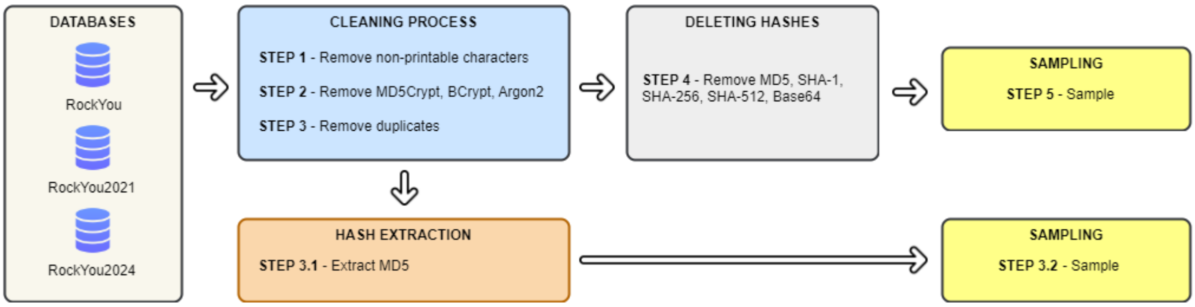

DATABASES
RockYou
RockYou2021
RockYou2024
CLEANING PROCESS
STEP 1 - Remove non-printable characters
STEP 2 - Remove MD5Crypt, BCrypt, Argon2
STEP 3 - Remove duplicates
DELETING HASHES
STEP 4 - Remove MD5, SHA-1, SHA-256, SHA-512, Base64
SAMPLING
STEP 5 - Sample
HASH EXTRACTION
STEP 3.1 - Extract MD5
SAMPLING
STEP 3.2 - Sample

Figure 1. Processing steps used in each file

		RockYou		RockYou2021		RockYou2024	
Step	Command	# lines after	Lines removed (%)	# lines after	Lines removed (%)	# lines after	Lines removed (%)
0	Download file	1.43×10^7	-	8.46×10^9	-	9.95×10^9	-
1	`strings`	1.43×10^7	0	8.46×10^9	0	9.95×10^9	0.02

2	`grep -v -Ea`
	`'^$[H,P]$[0-9,a-z,A-Z].{30}|`
	`^$[1]$[0-9,a-z,A-Z].{30}|`
	`^$[0-9]$rounds.{66}|`
	`^$argon2d$.{87}|`
	`^$[0-9]$.{8}'`
		1.43×10^7	0	8.46×10^9	0.00001	9.78×10^9	1.69

3	`sort | uniq`	1.43×10^7	0.04	7.99×10^9	5.50	9.24×10^9	5.40

4	`grep -v -Ea`
	`'^[a-f0-9]{32}$|`
	`^[a-f0-9]{40}$|`
	`^[a-f0-9]{64}$|`
	`^[a-f0-9]{128}$|`
	`^(?:[A-Za-z0-9+/]{4})*(?:[A-Za-z0-9+/]{2}==|`
	`[A-Za-z0-9+/]{3}=)?$'`
		1.43×10^7	0.0004	7.99×10^9	0	8.53×10^9	7.17

Table 2. Cleaning steps for each file and the amount of lines filtered out in each step

can contain binary data, duplicate lines and hashes, which would compromise the quality of the analysis if not removed.

To address this issue, a cleaning process, represented in Figure 1, is implemented, which aims to filter out irrelevant data and to prepare the dataset for the extraction of MD5 hashes for subsequent analysis. The MD5 hashes are chosen because of their documented vulnerabilities, and are subsequently subjected to a cracking process.

The cleaning process followed the steps detailed in Tables 2 and 3. Table 2 documents the number of rows resulting after each step and the percentage of filtered rows about the original dataset. Similarly, Table 3 presents the number of MD5 hashes extracted from each password database and their proportion to the total number of original lines. This data enables the analysis of the prevalence of hashes.

However, even after cleaning, the resulting files from the RockYou2021 and RockYou2024 databases, after step 4; and the hashes extracted from RockYou2024, after step 3.1, remained too large, rendering it unfeasible to perform a complete analysis in a timely manner. To solve this problem, a sampling technique is applied, as represented in steps 3.2 and 5 of Figure 1. Section 3.1 discusses the methodology adopted for this sampling, ensuring that the data analyzed continue to reflect the general characteristics of the original databases without a significant loss of relevant information.

3.1 Database sampling

Sampling is important when dealing with large volumes of data, as in the case of the RockYou2021 and RockYou2024 databases, since it allows for more agile analysis without compromising the representativeness of the observed patterns. Thus, the cleaning and sampling process adopted in this work facilitates the analysis and contributes to the reliability of the results.

To ensure a representative sample of the password databases, its size n is calculated by the Equation 1.

$$n = \frac{Z^2 \cdot p \cdot (1-p)}{E^2} \tag{1}$$

Considering a Z-score of $Z = 4.417$ for a 99.999% confidence level, a conservative estimate of the proportion $p = 0.5$, and a margin of error $E = 0.31\%$, Equation 1 yields a sample size of approximately $n = 500,000$ lines.

This number of lines is used for sampling the MD5 hashes extracted from RockYou2024 and the three password databases after step 4. As seen in Table 3, only 50 MD5 hashes were extracted from RockYou, and, therefore, no sampling is needed. No MD5 hashes were extracted from RockYou2021.

To achieve the random selection of the lines, the `shuf` command in Linux is used.

3.2 Pattern analysis procedures

This study uses the Python programming language, version 3.10.12, as the main tool for analyzing and comparing samples from the RockYou, RockYou2021 and RockYou2024 databases.

The analysis compares the passwords present in the three databases, considering parameters such as the length of the passwords, which is one of the factors that determines its strength. Longer passwords tend to be more secure, as they increase the difficulty of being cracked by brute force attacks. In addition, the robustness of the passwords is analyzed based on the presence of different characters, such as

		RockYou		RockYou2021		RockYou2024	
Step	Command	# MD5 hashes	%	# MD5 hashes	%	# MD5 hashes	%
3.1	`grep -Ea '^[a-f0-9]{32}$'`	50	0.0003	0	0	4.27×10^8	4.3

Table 3. MD5 hashes extraction from the files

upper and lower case letters, numbers and special symbols, which increases the complexity of the password.

Another important aspect addressed in the analysis is the use of personal information in passwords, such as first names and dates. This practice is widespread, but it makes passwords more vulnerable to targeted attacks, such as social engineering or dictionary attacks, in which attackers use the target's personal information to guess their credentials. Including these variables in the study is useful to identify user behavior patterns when creating passwords and, thus, propose better digital security practices.

By comparing these databases, it is also possible to evaluate the evolution of password creation patterns over time. With increasing awareness of the importance of digital security and the implementation of policies on online platforms, it is expected that changes in user behaviors will occur, such as the use of more diverse passwords. This historical analysis provides information for security professionals in developing risk mitigation strategies.

3.2.1 Entropy and expectation entropy calculation

One of the metrics used to compare the strength of these passwords is the entropy, as defined by Equation 2.

$$H = \log_2(N^L) \tag{2}$$

The entropy H, measured in bits, indicates how difficult it would be to brute force a password composed of L characters, each chosen from a character set of size N. The entropy is 0 bits for a known password and increases as the password length L grows or as the size N of the character set expands, such as when incorporating more categories like lower-case letters, upper-case letters, digits and symbols.

From Equation 2, it is noted that there is no theoretical upper bound for the entropy, that is $H \in [0, \infty)$. Conversely, the NIST Entropy Assessment Suite (NIST SP800-90B) returns a value $H_{NIST} \in [0, 1]$. To evaluate the strength of a password in this same scale, Reaz and Wunder [2022] proposed a new metric, called expectation entropy (HE) and defined by Equation 3. Both entropy and expectation entropy are used in this work.

$$HE = \frac{\log_2 (p_{\mathcal{L}} \cdot l + p_{\mathcal{U}} \cdot u + p_{\mathcal{D}} \cdot d + p_{\mathcal{S}} \cdot s)}{\log_2 |\mathcal{K}|} \tag{3}$$

The variable HE represents the expectation entropy of the password, as a number between 0 and 1. An HE value of x indicates that an attacker would need to search, on average, $x \times 100\%$ of the total possible guesses to successfully discover the password.

The variables p_L, p_U, p_D and p_S denote the proportions of lowercase letters, uppercase letters, digits and special characters, respectively. Corresponding lengths l, u, d and s reflect how many characters from each category contribute to the password's overall entropy. K is the total character space.

3.2.2 Detection of dates

It is a common practice to use dates associated to the user, such as wedding and birth dates, as their password. Such password is weak and easily discovered, and, to detect its usage, the regular expressions from Table 4 are used, enabling the identification of different date formats. It is important to note that these patterns only match if the date is used completely as the password, not a substring.

Date format	regex
YYYYMMDD	^(20[0-4]\d|19\d\d)(0[1-9]||1[0-2])(0[1-9]||[12]\d|3[01])$
DDMMYYYY	^(0[1-9]||[12]\d|3[01])(0[1-9]||1[0-2])(20[0-4]\d|19\d\d)$
MMDDYYYY	^(0[1-9]||1[0-2])(0[1-9]||[12]\d|3[01])(20[0-4]\d|19\d\d)$
YYMMDD	^([0-9]2)(0[1-9]||1[0-2])(0[1-9]||[12]\d|3[01])$
DDMMYY	^(0[1-9]||[12]\d|3[01])(0[1-9]||1[0-2])([0-9]2)$
MMDDYY	^(0[1-9]||1[0-2])(0[1-9]||[12]\d|3[01])([0-9]2)$

Table 4. Regular expressions for date format detection

One limitation to this approach is that a string could match to multiple patterns. For example, `011002` will positively match YYMMDD, DDMMYY and MMDDYY. All matched patterns are considered.

Also, a string like `111111`, although detected as a date, could be more likely chosen by the user due to the ease of remembering the password.

3.2.3 Detection of personal names

To identify the use of first and last names as substrings within the password, the `names-dataset` library, version 3.1.0, is used [Remy, 2021].

This library consists of 730 thousand first names and 983 thousand last names, which were obtained from the Facebook 2021 data breach. For each name searched, the library provides the probability of the name belonging to a man or woman, along with the likelihood of its most probable countries of origin.

To detect the use of names as substrings, we remove digits and symbols from the password string. The resultant letter-only substring is inputted into the library.

As a limitation, this method will not be able to detect the presence of name in a password like `JohnDoe-2018` or `Abc-JOHN-cbA`. It will, however, detect the presence in a password similar to `1J!o9h#n2`.

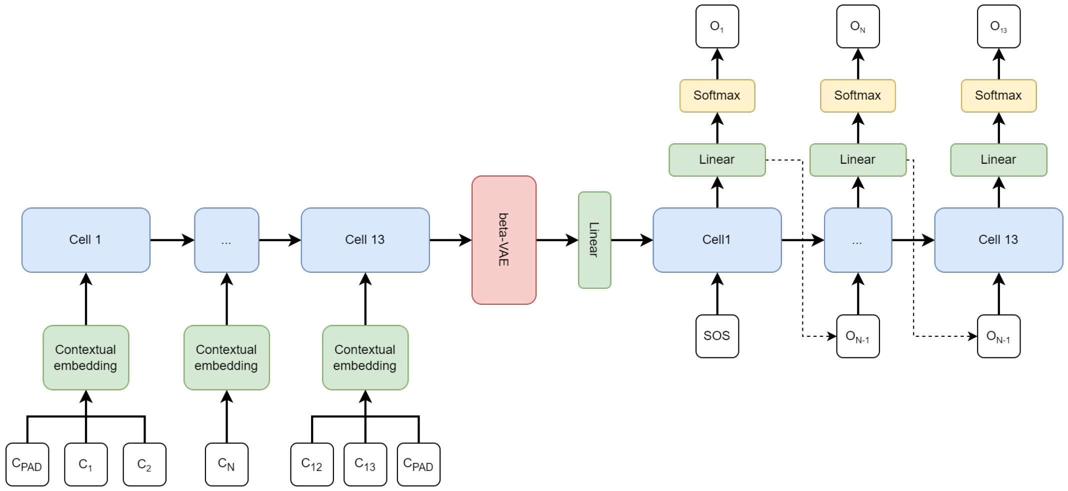


Figure 2. passBiRVAE architecture

3.2.4 Detection of common substrings

People regularly use sequences, such as `1234`; default credentials, like `admin`; or keyboard patterns, such as `qwerty`, as their password or as part of it. As consequence, some malware, like Mirai and Conficker, leverage these weak passwords to gain unauthorized access to their targeted devices [Grilo *et al.*, 2022].

To identify the use of these patterns, we use four dictionary lists elaborated by Miessler [2020], `mirai-botnet.txt`, `conficker.txt`, `walk.txt` and `scada.txt`.

The first two are related to the passwords attempted by the Conficker and Mirai malwares, respectively. The third refers to keyboard patterns and the latter lists commonly used Supervisory Control and Data Acquisition (SCADA) passwords, with potential to affect industry sectors that use these systems, such as energy generation and transmission.

3.2.5 zxcvbn library

This work uses `zxcvbn`, version 4.4.28, an open-source password strength estimator created by Dropbox [Wheeler, 2016]. It identifies common patterns, estimates how easily a password could be guessed and provides detailed feedback on how a password could be improved.

All passwords from the sample databases are inputted into `zxcvbn`, for further analysis regarding their strength, improvement suggestions and crack time.

3.3 MD5 hashes cracking procedure

The MD5 hashes extracted from each RockYou password database were input into Hashcat version 6.2.6, which performed a dictionary attack using the wordlist from `crackstation.net`. This wordlist contains 15 billion plaintext entries designed to target both MD5 and SHA1 hashes.

3.4 Distributions comparison

To statistically compare the distribution of these features among the different RockYou databases, we use Cliff's Delta ($\delta \in [-1, 1]$). It is a non-parametric measure used to assess the magnitude and direction of the difference between two distributions.

Table 5. Cliff's delta effect size interpretation

Cliff's delta value	Effect size
$0.474 \leq \lvert\delta\rvert$	Large
$0.330 \leq \lvert\delta\rvert < 0.474$	Medium
$0.147 \leq \lvert\delta\rvert < 0.330$	Small
$\lvert\delta\rvert < 0.147$	Negligible

A δ value of 1 indicates that all values in the first distribution are greater than those in the second, -1 indicates the opposite, and 0 suggests that the distributions are equal, with $\delta(x_1, x_2) = -\delta(x_2, x_1)$. Because of this, its modulus is used ($\lvert\delta\rvert$) for assessing the effect size between the two distributions.

Table 5 presents commonly used interpretations for the Cliff's delta effect size [Romano *et al.*, 2006; Chen *et al.*, 2019; Wan *et al.*, 2019].

3.5 passBiRVAE

This section is dedicated to explain the proposed network, Bidirectional Recurrent Neural Network with context, named passBiRVAE. We first develop the architecture and the fundaments that lead the modifications. Next, we describe the training and dataset used.

3.5.1 Architecture

From Xiao [2024], we adapted the model to additionally process the password backwards looking for relation between the ending of the password and how it may influence the results. This also enables more information flow between the network and learning more complex patterns. However, there is still a concern regarding local information, and how it may be more efficiently processed to make more evident types of substrings. Being able to recognize and reproduce such patterns improves the ability of the netwrok to exploit more passwords and generate with increased accuracy reliable results.

Motivated by Wu *et al.* [2023], which discovered that a good amount of information may be found the previous and following character of a certain element in the password, we propose a embedding context to help the network better understand local dependencies and differentiate the patterns seen during training. Thus, subwords or substrings may be better processed and incorporated as representations by the VAE. For example, analyzing the hypothetical password (inspired in real ones): `myVAE2024` and `myVAE5892`. Both have the same structure but the numbers on it differ in meaning (in the first case it is a year, and, therefore, more predictable).

Another pattern seen in this password is the influence of the ending (the numbers) in the start of the password, enabling the network to better divide this occasion where a sequence of letters is followed by numbers. Other example is how the number may affect the following. In the hypothetical password `myVAE1234` and `myVAE4321`, both have the same characters but the appearance of 4, instead of 1, lead the inversion of the pattern. It is worth mentioning that the benefits of the passBiRVAE with context are not limited to these examples. They serve only to clarify the motivations. The architecture of the model is depicted in Figure 2.

The contextual embedding is designed as shown in figure 3.

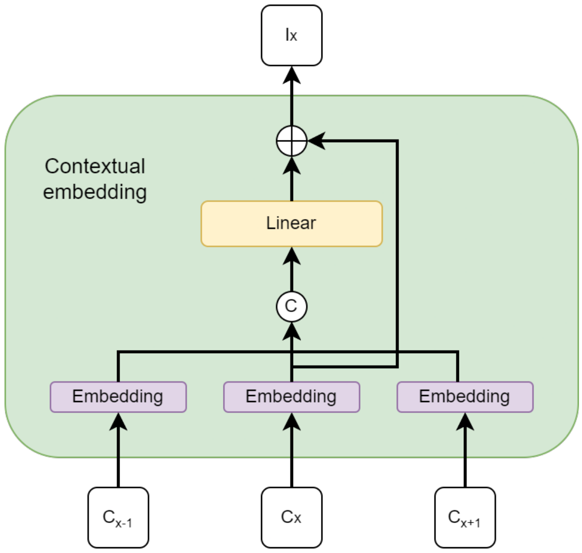


Figure 3. Contextual embedding architecture

The encoder is composed of 13 Gated Recurrent Unit (GRU) cells each of them having as its input the contextual embedding. The contextual embedding is composed of a embedding, which have shared parameters, which receives the input token with values between 0 and 95. The previous and next token are also processed by the same embedding.

Next, they are concatenated and passed trough a linear layer with 384 dimensions, and then summed with the C_x. The concatenation is done to preserve the information and the order of the input sequence. The cells in Figure 2 are composed of 3 stacked layers of Bidirectional GRUs. GRUs were chosen for being cheap and efficient for small amounts of sequential data (like in present work). At each cell the hidden state of the previous one is passed as hidden state of the next. The cells have in the input and output, respectively, 384 and 256 dimensions. In bidirectinal mode, the hidden states are summed to its dimension be divided by 2, since its hidden state dimension is composed of (2*hidden layers, batch size, hidden layers size), which have the respectively shape (6,128,256). After passing through the cells, the hidden state is used as input in beta-VAE with latent dimension of size 128. In the decoder, there is a linear layer with 256 dimension which output is used as hidden state in the first GRU cell. The GRU cells have as input shape 96, which is the dimension of our vocabulary, including padding and End of Sequence (EOS), and 256 of hidden state size. Then, the output of the cell goes to a linear layer (which is also shared thorugh all cells) with 96 dimensions. From there, its logits (each unit of the layer represents a class, i.e. a character in the vocabulary) are passed as input to the next cell. After passing trough all cells, or receiving a EOS, the logits of each linear layer are used in the loss. Top-K was chosen (after passing in the Softmax) aiming the generation of diverse vocabulary.

3.5.2 Training and database

Although Xiao [2024] used around 24M passwords, we were unable to replicate these database due to limitations in training time. Thus, we used 4.56M unique passwords, randomly selected from the RockYou2024 total database, which 80% were used as training and the rest as validation. The number of samples was chosen mainly due to hardware and training time limitations. Despite being only 20% of the usual train set used by hitherto referenced networks, it can represent with safety the passwords contained in RockYou2024, since only 500000 samples are required to have a broad comprehension of the patterns, as explained in Section 3.1.

Regarding the characters used, it was only considered the characters 33 to 126 (totalizing 93 valid characters) from ASCII table. In total, our vocabulary have 95 token taking into account padding and EOS tokens. We trained the network for 30 epochs using Adam optimizer with initial learning rate set to $3 \cdot 10^{-6}$. Adam's parameters were set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$, as these are standard values used in literature due to their proven effectiveness in balancing convergence speed and stability. While fine-tuning these parameters might enhance performance further, our focus was on improving the overall network performance rather than on optimizing specific hyperparameters at this stage. Future studies could explore variations in β_1 and β_2 to refine results further.

We also applied Learning Rate Scheduler using Cosine Annealing, with minimum of 10^{-8} of learning rate. It is worth mentioning that some stabilization problems were faced, and, therefore, the learning rate was set to theses values. Cross-entropy was used as part of the loss, which was divided in two types: reconstruction loss (R_l), as defined by Equation 4, and similarity loss (S_l), as Equation 5. The total loss is given by Equation 6.

$$R_l = \sum_i X_i \log X_i' \tag{4}$$

$$S_l = \frac{-\beta}{2}(1 + \log \omega^2 - \mu^2 - \omega^2) \tag{5}$$

$$Total\ loss \;=\; R_l \;+\; S_l \tag{6}$$

Table 6. Percentage of password lengths across different password databases.

Length (L)	[Mannuela *et al.*, 2021]	RockYou	RockYou2021	RockYou2024
$L < 8$	15.1%	33.08%	10.96%	10.05%
$8 \leq L \leq 12$	55.7%	59.84%	66.24%	65.77%
$13 \leq L \leq 16$	14.2%	6.22%	18.79%	17.61%
$L > 16$	15.1%	0.85%	4.00%	6.57%

In the initial steps of the first epoch, β was also initialized using the following formula to enable more diversity in the latent space and increase the capability to learn complex patterns Higgins *et al*. [2017]; Bowman *et al*. [2015]. The value of β, as according to Equation 7, was set to 1.

$$\beta = \frac{1}{1 + e^{-0.0025 \cdot (iteration - 2500)}} \quad (7)$$

4 Password patterns across the databases

This section presents the results of the analysis, comparing the password patterns between RockYou, RockYou2021 and RockYou2024.

4.1 Passwords length

One factor that determines password strength is its length. Longer passwords are generally more difficult to crack because more possible combinations exist.

The RockYou database contains passwords ranging from 1 to 255 characters in length. The RockYou2021 database has a narrower range, with passwords varying from 4 to 20 characters. Ultimately, the RockYou2024 database includes passwords between 4 and 231 characters. The lengths for RockYou and RockYou2024 suggest that non-password lines are still present in the database after the cleaning process, as a 255-character string is unlikely to be a password.

Nevertheless, 95% of the passwords in the RockYou dataset contain 13 characters or fewer. The 95th percentile for password length is 16 characters in RockYou2021 and 18 characters in RockYou2024. Consequently, Figure 4, which depicts the amount of passwords in each database with specific lengths, is restricted to a maximum of 30 characters.

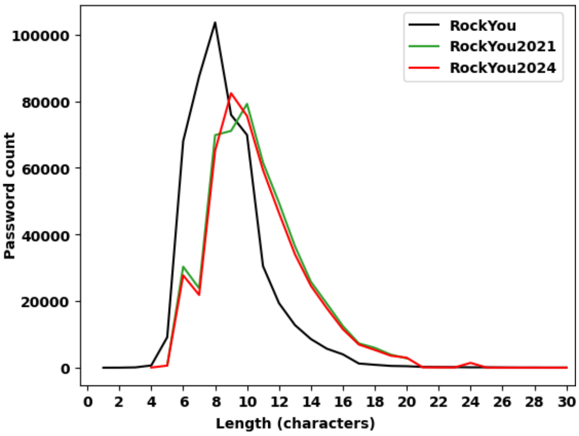


Figure 4. Number of passwords for each length

It may be noted, from Figure 4, that the most frequent password length in RockYou is 8 characters, whilst RockYou2021 and RockYou2024 are more distributed in longer lengths, with global peaks in 10 and 9 characters, respectively.

These findings are aligned with those of Mannuela *et al.* [2021], who found that the majority of the respondents of a questionnaire use passwords with lengths between 8 and 12 characters. In contrast, as shown in Table 6, the greatest difference is for passwords with more than 16 characters, as we have found significantly less credentials with those lengths than Mannuela *et al.* [2021].

Also, the medians of these distributions are 8 characters for the RockYou database and 10 characters for RockYou2021 and RockYou2024, indicating a trend of increasing the length of the passwords within these 15 years.

The shift toward longer passwords observed in RockYou2021 and RockYou2024 may be attributed to stricter password creation requirements imposed by systems. As an example, Imamaliyev and Khudoykulov [2021] found that among six email services investigated, two had minimum length requirements of more than 8 characters, whilst three set the minimum between 6 and 8 characters. Similar restrictions across general systems may explain the decline in the proportion of passwords shorter than 8 characters ($L < 8$) from 33.08% in RockYou to approximately 10% in both RockYou2021 and RockYou2024, as observed in Table 6. It could also be a consequence of different social identities comprehended in the leaks [Grobler *et al.*, 2021].

Although this suggests that the passwords have become more secure since the RockYou leak, the longer passwords may use weak substrings or fewer LUDS combinations. Sections 4.2 and 4.6 promote the discussion on password strength more robustly.

Figure 4 also shows that RockYou2021 and RockYou2024 follow a very similar distribution regarding the lengths of the passwords, including the presence of a local peak in 6-character long passwords.

4.2 Entropy and expectation entropy

Entropy is a common metric for evaluating the strength of passwords, as it measures the credential's unpredictability. A higher value of entropy indicates more randomness and, therefore, a stronger password.

Figure 5 presents the entropy distribution within the databases, demonstrating a peak in RockYou password entropy of around 40 bits. This concentration in this entropy value has also been observed in other datasets [Walia *et al.*, 2020], leaked between 2008 and 2016 [Wang *et al.*, 2018].

For the RockYou2021 and RockYou2024, the entropy peak shifted to around 50 bits, suggesting stronger passwords

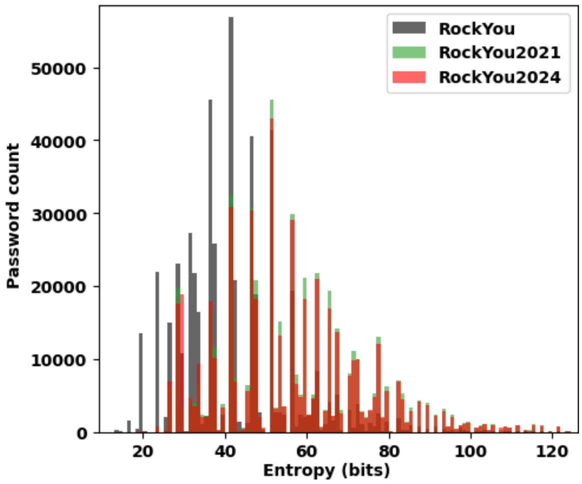


Figure 5. Distribution of the entropy

in these databases than in the original RockYou. This observation is also supported by the expectation entropy, as seen in Figure 6, which shows that the RockYou passwords are more distributed around lower values of expectation entropy than the RockYou2021 and RockYou2024 ones.

All passwords in RockYou and RockYou2021 have expectation entropy below 0.4, meaning that an attacker would have to guess, on average, 40% of the total number of possible guesses, based on each password's length and number of LUDS, before successfully obtaining the correct credential.

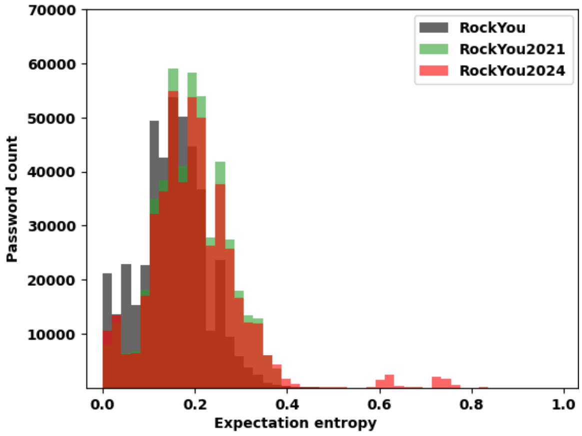


Figure 6. Distribution of the expectation entropy

Most of RockYou2024 passwords are also below this threshold, except for a few around 0.6 and below 0.8. These high expectation entropy values could be associated with non-password strings with higher lengths, as pointed in Section 4.1

It is also observed that the RockYou2021 and RockYou2024 distributions for both entropy and expectation entropy are very similar.

4.3 Use of dates

Users frequently rely on deducible information when creating passwords to make them easier to remember. Although this practice is common, it is widely considered insecure and inadvisable. This is because personal data, such as birth dates and family names can be easily obtained through Open Source Intelligence (OSINT) techniques. These techniques involve collecting publicly available information, often from social networks or other online platforms, allowing attackers to easily access this data [Keküllüoğlu *et al.*, 2022].

For example, an attacker may identify a user's birth date through social media posts, publicly celebrated birthdays, or other digital interactions that reveal this data type. Once this information is discovered, the attacker can use it to conduct social engineering or dictionary attacks, in which password-guessing attempts are based on predictable combinations of personal information. This type of attack becomes even more effective when users combine this data with simple sequences of numbers or letters, reducing the password's protective capability. In addition, using personal information in passwords contradicts good digital security practices, which recommend creating strong, complex credentials that combine alphanumeric characters, special symbols and variations of upper and lower case letters.

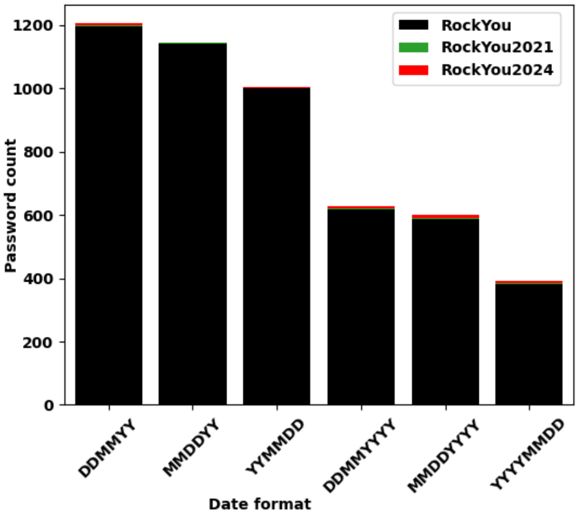


Figure 7. Distribution of the used date formats

Thus, including personal data in passwords increases users' vulnerability to cyberattacks. To minimize this risk, it is essential that users adopt more secure methods of creating passwords, such as automatic random password generators, or implement multi-factor authentication (MFA) to add an extra layer of protection [Dastane, 2020].

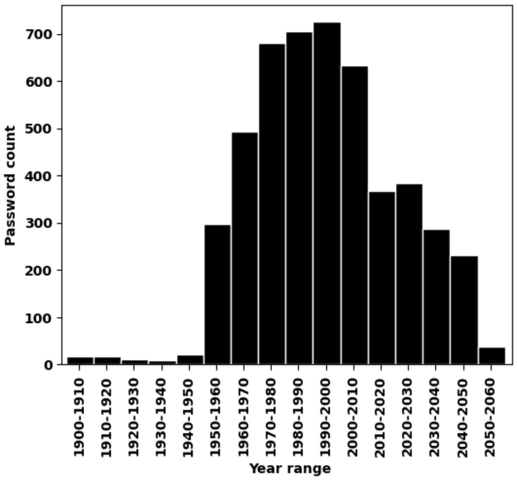
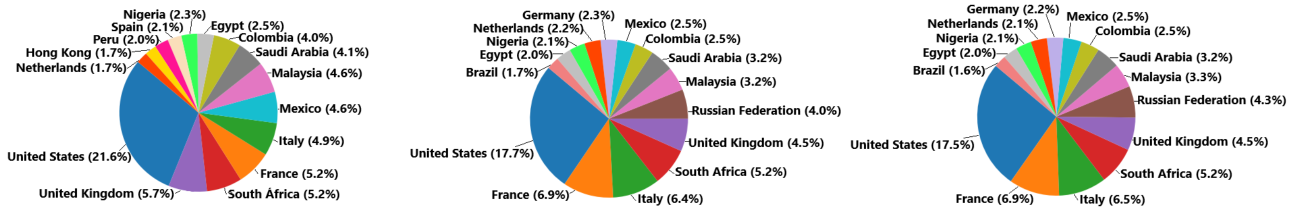


Figure 8. Distribution of the used date years



(a) From RockYou **(b)** From RockYou2021 **(c)** From RockYou2024

Figure 9. Probability of origin of first names used in passwords

As seen in Figure 7, date-like passwords are mainly used in RockYou database, with significantly less use in RockYou2021 and RockYou2024. This could be due to more restrictive requirements that prohibit the use of digits only passwords [Alroomi and Li, 2023]. Also, 6 and 8 characters passwords that represent dates are much more common in RockYou than in the other databases, as seen in Section 4.1.

The year from these RockYou date passwords are, then, plotted as in Figure 8. As the RockYou database is dated from 2009, the passwords composed of years from 2010 and beyond are more likely related to the ambiguities discussed in Section 3.2.2. Nonetheless, the findings indicate a higher concentration of date-like passwords predominantly spanning the years 1980 to 2010, which are realistic dates considering the time range comprehended in the database.

4.4 Use of personal names

Similar to the use of dates, passwords often include personal names as part of a strategy to make them easier to remember. Table 7 shows the number of passwords that contain a first or last name as a substring. This behavior is concerning from a security perspective since names are easily accessible information, often publicly available on social networks, increasing vulnerability to social engineering attacks and other forms of hacking.

This is relevant to analyzing password creation patterns, as it illustrates how including commonly known information can make passwords more predictable. By using common names, users inadvertently reduce the complexity of their passwords, making it easier for attackers who use dictionary techniques or lists of common names to crack passwords. Including personal names in passwords also reflects the tendency for users to prioritize ease of memorization over security.

Table 7. Counts of use of personal names in each database.

Database	First name	Last name
RockYou	95,852	93,437
RockYou2021	77,859	83,303
RockYou2024	72,253	77,662

It is important to note that specific names, such as `Jordan`, can be identified as both a first and a last name, resulting in double counting.

Using the `names-dataset` library, the probability that the first names present in the passwords originate from the ten most likely countries was calculated. For this calculation, the sum of the probabilities of each name belonging to these countries was performed. Then, the final result was divided by the total number of first names analyzed, as according to Table 7, resulting in an average probability of all names being associated with each of these countries.

This approach allows a detailed analysis of the geographic origins of the most commonly used names in passwords, revealing demographic patterns of password choice that may vary between regions. The results of this analysis are illustrated in Figure 9, which presents the average probability of the origin of the names for the most likely countries.

The RockYou dataset consists of passwords breached by a U.S.-based company of the same name [Weir *et al.*, 2010]. Consequently, most of the leaked passwords are expected to originate from U.S. users. However, the origins of RockYou2021 and RockYou2024 are unclear, as they were first distributed anonymously on hacking forums without further information.

Figure 9 shows that the United States is the most probable country of origin for all databases, with comparable probabilities. This indicates that, like RockYou, the RockYou2021 and RockYou2024 are also composed primarily of passwords from users of this country. It is also observed that the general distribution among the countries in RockYou2021 (Figure 9b) and RockYou2024 (Figure 9c) are very similar.

Using a similar approach, the gender of the names is estimated, with the results depicted in Figure 10. It shows that there is no predominance of gender in the usage of first names within passwords, with this pattern being consistent across all databases.

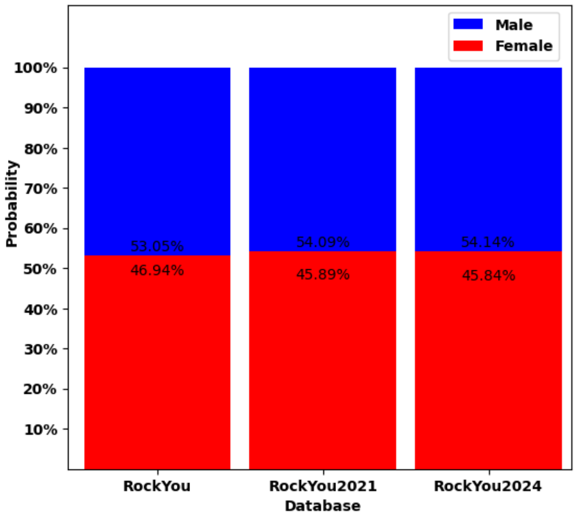


Figure 10. Probability of gender of the names used

4.5 Use of common substrings

Using standard strings in passwords, such as sequential characters or easily memorable combinations like `1234` or `qwerty`, poses significant security risks, as they are predictable and vulnerable to brute force and dictionary attacks.

Figure 11 shows the most common substrings in RockYou2024 and their occurrences in the other databases. These substrings are extracted from malware behavior (Figures 11a and 11b), from industrial default credentials (Figure 11c) and from keyboard spatial patterns (11d).

Substrings that appeared in multiple dictionaries were suppressed from one of them, to maximize the number of patterns presented. From Conficker, the suppressed patterns are `1111`, `54321`, `12345` `123456`, `12345678` and `admin`. For SCADA, the strings are `1234`, `123456` and `admin`. The corresponding values for each of these strings can be visualized in other plots of Figure 11.

The results show that the use of these common patterns is significantly lower in the recent RockYou versions than the original 2009 version. This indicates that the users are now more frequently prioritizing security over convenience, which is in accordance with the study of Wash and Rader [2021].

In conjunction with the findings from Sections 4.1 and 4.2, Figure 11 also indicates that the passwords in RockYou2021 and RockYou2024 are more secure than those within RockYou.

Of the most common Mirai substrings within these databases, from Figure 11a, `12345`, `1111`, `123456`, `pass` and `admin` have been observed to be also most commonly used by the botnet [Pimenta Rodrigues *et al.*, 2017].

4.5.1 Industry common passwords

Figure 11c highlights the most common SCADA substrings, notably `master`, `guest`, `wago` and `engineer`. These terms are frequently used as default passwords in energy management devices [Miessler, 2020], which are considered critical infrastructure [Rehak *et al.*, 2022], and, therefore, must be secured to prevent unauthorized access and potential disruptions to essential services.

Malicious users targeting SCADA system's usual password attempts include `12345`, `123456789`, `0670` and `amine`, as observed in a SCADA honeypot environment [Belqruch and Maach, 2019]. From these, `12345` and `123456789` are also commonly used as passwords (Figures 11a and 11b, respectively). The use of these strings, however, is lower in the 2021 and 2024 versions of the database, indicating that they have been decaying in use.

Recent Common Vulnerabilities and Exposures (CVEs) highlight significant security risks associated with password usage in SCADA environments. For instance, vulnerabilities like CVE-2024-23784, CVE-2021-22741, CVE-2023-4986 and CVE-2023-42493 indicate weaknesses in managing and storing passwords, making it easier for attackers to exploit them. The latter is graded as a critical Common Vulnerability Scoring System (CVSS).

In industrial environments, where operational technology is fundamental, the implications of these vulnerabilities are severe. Compromised passwords can lead to unauthorized manipulation of industrial processes, resulting in safety risks and operational disruptions. Hence, it is crucial to adopt robust password policies that discourage using common substrings to enhance security posture.

4.6 zxcvbn results

This study uses `zxcvbn` strength estimator [Wheeler, 2016] to evaluate the passwords more thoroughly. This library evaluates the complexity of a password based on various factors, including length, character diversity and common patterns.

Figure 12 shows the estimated time to crack the passwords present in each database, according to the results obtained from the `zxcvbn` tool. The graph reveals a sharp drop in the number of passwords with shorter cracking times, especially in the RockYou database, which suggests a significant proportion of weak passwords. This behavior reflects a lower level of awareness about digital security.

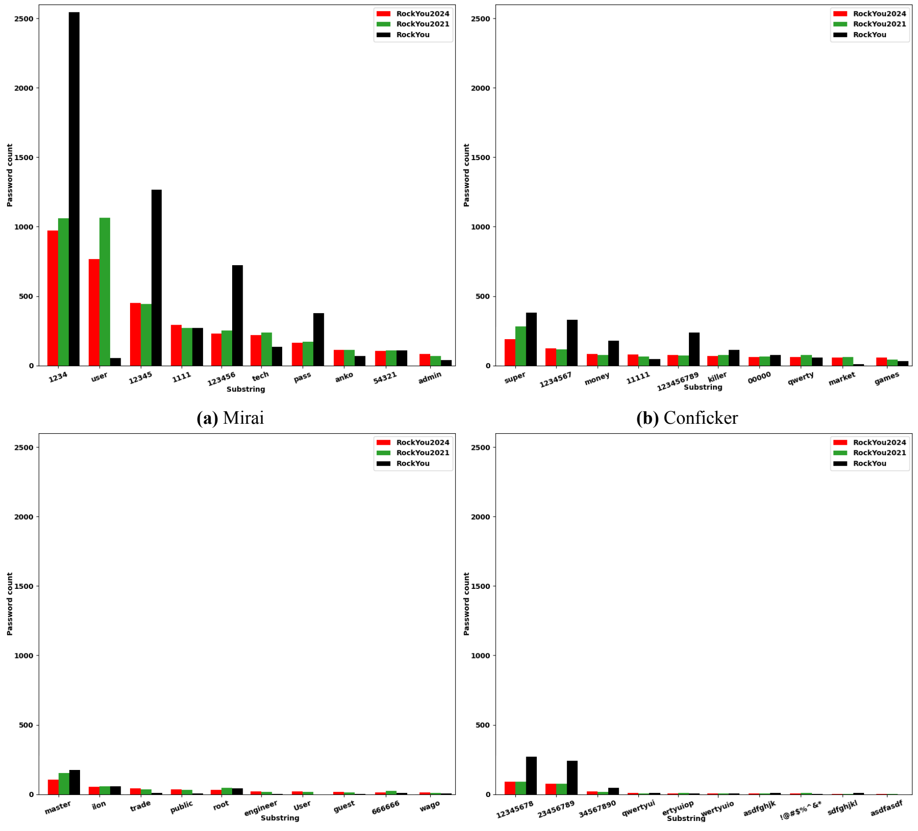
In contrast, the RockYou2024 database displays a broader and more persistent distribution of longer cracking times, indicating stronger passwords compared to the older datasets. This trend can be explained, in part, by the presence of strings of characters that are not actual passwords but that remained in the file, as detailed in Section 4.1. These additional elements may have contributed to the perception of greater robustness in some entries. Furthermore, the discussions in Sections 4.2 and 4.5 suggest that passwords in the RockYou2021 and RockYou2024 databases tend to be significantly more robust than those in the original RockYou. This conclusion is corroborated by the analysis of `zxcvbn`, which, as illustrated in Figure 13, classifies most RockYou passwords as weak or moderate.

In contrast, passwords in the RockYou2021 and RockYou2024 databases are predominantly classified as strong. It is also worth noting the significant increase in the number of passwords considered very strong in the more recent databases, as opposed to the RockYou database, where such passwords are much less frequent.

This can be attributed to the gradual increase in awareness of the need for stronger passwords, driven both by the evolution of cyber threats and by implementing more stringent security policies by digital platforms. Using more diverse characters, longer passwords and adopting good practices, such as password managers, contribute to this improvement observed in the most recent databases. Consequently, it is possible that the increasing adoption of more advanced authentication mechanisms, such as MFA, has influenced the quality of passwords, as users are encouraged to create more complex combinations. This scenario reflects significant progress in digital security practices over time, highlighting the importance of educating users and improving protection mechanisms against password-cracking attacks.

For each password, `zxcvbn` may suggest improvements. These suggestions were translated into weaknesses, as according to Table 8, which are later plotted by their frequency of occurrence in the datasets, as shown in Figure 14.

The ”Few words” is the predominant weakness in Figure 14, showing the highest frequency across all three datasets. The presence of Dates is also more significant in the Rock-



(a) Mirai

(b) Conficker

(c) SCADA

(d) Keyboard walk

Figure 11. Occurrence of common substrings

Table 8. Password creation advice and corresponding categories.

Suggestion	Weakness
Add another word or two. Uncommon words are better.	Few words
Avoid dates and years that are associated with you.	Dates
All-uppercase is almost as easy to guess as all-lowercase.	All-uppercase
Avoid repeated words and characters.	Repeated words/chars
Capitalization doesn't help very much.	Capitalization
Avoid years that are associated with you.	Years
Predictable substitutions like '@' instead of 'a' don't help very much.	Leet
Reversed words aren't much harder to guess.	Reversed words
Avoid recent years.	Recent years
Use a longer keyboard pattern with more turns.	Keyboard patterns
Avoid sequences.	Sequences

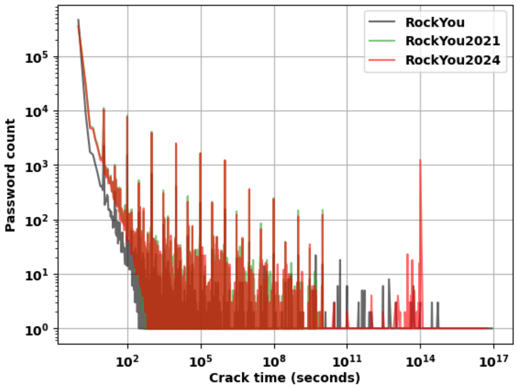


Figure 12. Crack time

You dataset. The remaining weaknesses exhibit much lower frequencies across all datasets, with minor differences between them.

Figure 14 suggests that while some password creation patterns persist, there are weaknesses that have been improved since the RockYou database. Nevertheless, other weaknesses, like the use of leet, remain relatively consistent in all datasets.

4.7 Cliff's delta effect size

Given that the analysis in Sections 4.1, 4.2 and 4.6 suggest great similarity between the distributions of features of RockYou2021 and RockYou2024, we use Cliff's delta to provide a statistical measure of the magnitude of this similarity. The analysis in Sections 4.3 to 4.5 are excluded from the Cliff's delta measurement because they involve categorical values,

Table 9. Modulus of Cliff's delta and interpreted effect sizes between features of different password databases

Compared feature	Section	RockYou to RockYou2021	RockYou to RockYou2024	RockYou2021 to RockYou2024
Length	4.1	0.404 (medium)	0.419 (medium)	0.020 (negligible)
Entropy	4.2	0.494 (large)	0.460 (medium)	0.011 (negligible)
Expected entropy	4.2	0.363 (medium)	0.327 (small)	0.020 (negligible)
Crack time	4.6	0.206 (small)	0.198 (small)	0.007 (negligible)

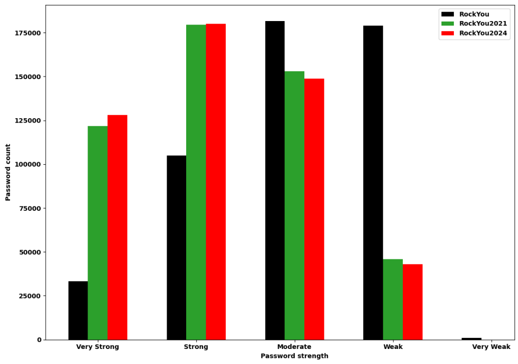


Figure 13. Score evaluation

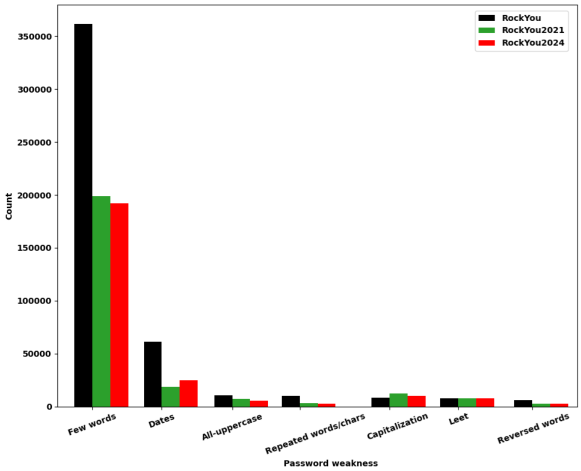


Figure 14. Weaknesses and their suggestions for passwords improvement

which are not suitable for this type of effect size calculation.

The Cliff's delta results are shown in Table 9, indicating negligible differences between RockYou2021 and RockYou2024 features. Conversely, these databases demonstrated small to large differences with the RockYou wordlist, and the analysis indicate more secure patterns in the more recent leaks.

These results indicate that the passwords databases have changed from 2009 to 2021, but no significant change is noted between 2021 and 2024.

5 MD5 hashes cracking

The security of cryptographic hash functions has become a critical concern in the realm of data protection, particularly with respect to MD5 hashes, which are now considered unsuitable for further use in security-sensitive applications, such as password [Marchetti and Bodily, 2022].

Despite this, many websites still use MD5 associated to authentication processes, making them vulnerable to cracking attempts [Nugroho and Mantoro, 2023].

Table 10. MD5 hashes cracked.

Database	Cracked hashes (%)
RockYou	2 (4%)
RockYou2024	26058 (5.21%)

The fact that MD5 is still used in more recent datasets, such as RockYou2024, is concerning. As demonstrated in Table 10, 5.21% of the MD5 hashes in RockYou2024 were cracked using basic wordlists and publicly available tools. This was facilitated by the weak passwords in the dataset.

6 passBiRVAE analysis

Although the proposed method has a fundamental basis, the results did not converge to a good value. Both training and validation had almost equal performance, suggesting that the model learned to generalize well, despite its limitations in producing good results, as shown in figure 15. We credit this performance to the lack of training, since we trained with only around 20% of the advised quantity, and, therefore, complete training and improvements should be made in future studies. Therefore, we could not validate whether the model exploited the incorporated patterns to generate new samples according to observed vulnerabilities. However, the idea of maintaining the extra context remains the same. During our tests, the network with context converged faster than the one without contextual embedding, suggesting that it is worth using such architecture.

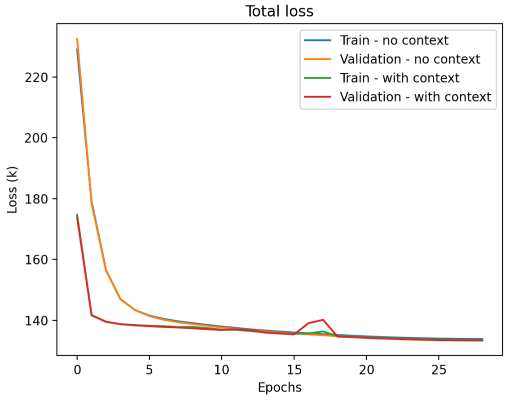


Figure 15. Training loss graph

7 Threats to validity

As proposed by Wohlin *et al.* [2012], the findings of this work face four different types of threats to their validity:

7.1 Construct validity

Construct validity refers to the extent to which a measure accurately assesses the intended construct.

Errors in data analysis or processing, especially regarding the cleaning of non-password lines from RockYou2024, as detailed in Section 3, can distort the results, leading to inaccurate representations of password patterns. This affects the reliability of the measures and, consequently, the validity of the conclusions.

The techniques used in this study could also fail to encompass all relevant dimensions related to passwords usage. Inadequate representation of the full scope of what constitutes a strong password may lead the study to draw conclusions that do not accurately reflect the construct of password security.

7.2 Internal validity

Internal validity indicates the degree to which a study's findings can be attributed to the independent variable rather than confounding factors.

The exact origin of the passwords leaked is unknown, and the selection of users' credentials could be biased. Suppose the database predominantly contains, for example, passwords from users of a particular age group, such as college students. In that case, the findings may not accurately represent the password behaviors of older adults.

It may also apply to the users' level of education, as people with higher awareness may create stronger passwords independently of the system's complexity requirements.

7.3 External validity

External validity assesses how much a study's findings can be generalized to other populations and settings.

Factors external to user behavior, such as the origin of passwords or the demographic profile of users, can influence the observed patterns, potentially distorting the results. For example, if a password database is predominantly composed of passwords from a specific type of service, such as social networks, while another database contains passwords from online gaming services, the differences in password patterns may reflect the complexity requirements or characteristics of the services in question rather than user behavior. This raises the question of to what extent the patterns observed in a sample of passwords can be attributed to actual user behavior since these patterns can be strongly shaped by the policies of each type of service.

Furthermore, the diversity of password databases used may be limited, compromising the generalizability of the results. If the dataset originates from a specific geographic region, the conclusions drawn about password creation patterns may not be applicable to a more heterogeneous global population. For example, password creation patterns observed in a database from a country with strict digital security policies may differ significantly from a database from a country where password creation rules are more relaxed. The lack of diversity in the data may lead to an overestimation or underestimation of password strength across different cultural, economic and social contexts.

7.4 Conclusion validity

Conclusion validity refers to the degree of accuracy with which the conclusions of a study are corroborated by the data and the research design.

Despite the use of a substantial sample of 500,000 passwords, which theoretically provides a confidence level of 99.999% with a margin of error of 0.31%, (as demonstrated in Section 3.1), the sample size may still give rise to threats to validity.

Although the sample size can be regarded as statistically robust, there is a risk that specific password creation patterns, particularly those that occur less frequently, are not fully integrated in the database. These rare patterns may include passwords with unusual character combinations or creative password formulation strategies that deviate from conventional practices. Underrepresenting these cases may result in distorted conclusions about the diverse and complex passwords analyzed, and lead to an incomplete or biased view of user behavior. Another factor that should be taken into account is the geographic and cultural variability that may influence password creation.

8 Conclusions and future works

This study presented a comparative analysis of password vulnerabilities in the RockYou, RockYou2021 and RockYou2024 databases, emphasizing the observed improvements and the persistent challenges in password security. The results indicated that password security has improved significantly compared to the 2009 RockYou database. Improvements include a notable increase in password entropy, a reduction in using dates as part of passwords and a decrease in the reliance on common substrings. These advances suggest that users adopt more secure practices when creating access credentials.

However, despite these improvements, some security challenges remain. For example, the number of passwords containing weak substrings, such as `1234`, is still considerably high. This behavior results in relatively low password cracking times for most analyzed sets. Furthermore, the percentage of MD5 hashes cracked using simple word lists remained constant between the RockYou and RockYou2024 databases, indicating the continued prevalence of passwords commonly found on MD5 cracking lists. This is concerning, as it reflects the persistence of inappropriate password selection practices.

These results indicate the importance of users and organizations complementing these improved practices with more robust password creation habits, using more secure hashing algorithms and implementing more comprehensive password policies. Adopting MFA mechanisms and using password managers are also effective strategies for strengthening security in digital environments.

The RockYou2021 and RockYou2024 databases showed similar distributions across several analyses, including password length (Section 4.1), entropy (Section 4.2) and `zxcvbn` results (Section 4.6). These similarities can be explained by two main factors. First, the relatively short three-year gap between the two databases may indicate a stagnation in users' password creation practices, suggesting that there has been no significant change in password creation habits during this period. Second, the similarity between the two datasets can also be attributed to the same passwords in both databases, implying that RockYou2024 may contain many passwords already compromised in previous breaches.

Furthermore, the findings indicate that the United States is the most probable country of origin for most of the passwords in both RockYou2021 and RockYou2024. Given the prior uncertainty surrounding the geographical origins of the passwords, as a consequence of the lack of transparency regarding the collection methods employed by the hackers who disclosed these leaks, comprehensive analysis has often been hindered. By elucidating the probable origins of the passwords, our research contributes to future works that may use these datasets considering demographic and cultural patterns of password creation.

As a proposal for future work, we suggest conducting similar analyses in a broader range of password databases, including those from different geographic regions and contexts, to determine whether the observed patterns are universal or significant variations in password usage across cultures. We also propose developing improved data cleaning methodologies that can more effectively filter out entries that do not correspond to actual passwords. This could include using machine learning techniques to identify and eliminate irrelevant records more accurately. Furthermore, after adequate cleaning of the RockYou2024 database, future research should investigate the causes of the observed similarities with RockYou2021.

Regarding passBiRVAE, future work should seek to increase the size of the dataset, training with 24 million passwords instead of just 3.65 million. With a network that already presents promising results, studies on the effectiveness of the contextual layer compared to current methods should be carried out to evaluate the ability of contextual incorporation to understand more complex patterns and substrings. In addition, an improved version of the decoder could better capture prior information, just as the encoder does through the contextual layer. Since our information loss is still higher than ideal, future work should also explore methods to reduce this loss throughout the autoencoder structure. With these modifications, we hope to achieve even more robust results, which will allow advances in understanding password vulnerabilities.

After all these modifications, it is expected that better results will be achieved, which will permit advancements in understanding vulnerabilities of passwords.

Acknowledgements

The authors would like to thank the support of the Brazilian National Confederation of Industry (CNI) and the University of Brasilia.

Funding

This research received no funding

Authors' Contributions

All the authors contributed equally to this work. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The data used in this work is publicly available online

References

Alroomi, S. and Li, F. (2023). Measuring website password creation policies at scale. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 3108–3122. DOI: 10.1145/3576915.3623156.

AlSabah, M., Oligeri, G., and Riley, R. (2018). Your culture is in your password: An analysis of a demographically-diverse password dataset. *Computers & security*, 77:427–441. DOI: 10.1016/j.cose.2018.03.014.

Belqruch, A. and Maach, A. (2019). Scada security using ssh honeypot. In *Proceedings of the 2nd International Conference on Networking, Information Systems & Security*, pages 1–5. DOI: 10.1145/3320326.3320328.

Bichara, M. d. A., dos Reis, M. A., Marcondes, M. R., da Silva Eleutério, P. M., and Vieira, V. H. (2023). Forensic method for decrypting tpm-protected bitlocker volumes using intel dci. *Forensic Science International: Digital Investigation*, 44:301514. DOI: 10.1016/j.fsidi.2023.301514.

Biesner, D., Cvejoski, K., Georgiev, B., Sifa, R., and Krupicka, E. (2021). Advances in password recovery using generative deep learning techniques. In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part III 30*, pages 15–27. Springer. DOI: 10.1007/978-3-030-86365-4_2.

Biesner, D., Cvejoski, K., and Sifa, R. (2022). Combining variational autoencoders and transformer language models for improved password generation. In *In Proceedings of the 17th International Conference on Availability, Reliability and Security*. DOI: 10.1145/3538969.3539000.

Bispo, G. D., Vergara, G. F., Saiki, G. M., Martins, P. H. d. S., Coelho, J. G., Rodrigues, G. A. P., Oliveira, M. N. d., Mosquéra, L. R., Gonçalves, V. P., Neumann, C., and Serrano, A. L. M. (2024). Automatic literature mapping selection: Classification of papers on industry productivity. *Applied Sciences*, 14(9). DOI: 10.3390/app14093679.

Bojinov, H., Bursztein, E., Boyen, X., and Boneh, D. (2010). Kamouflage: Loss-resistant password management. In *Computer Security–ESORICS 2010: 15th European Symposium on Research in Computer Security, Athens, Greece, September 20-22, 2010. Proceedings*

Table 11. Acronyms used in the study along with their meanings.

Acronym	Meaning	Acronym	Meaning
CVE	Common Vulnerabilities and Exposure	CVSS	Common Vulnerability Scoring System
EOS	End Of Sequence	GAN	Generative Adversarial Networks
GRU	Gated Recurrent Unit	LUDS	Lowercase, Uppercase, Digits and Symbols
MFA	multi-factor authentication	OSINT	Open Source Intelligence
PSM	Password Strength Meters	SCADA	Supervisory Control and Data Acquisition
VAE	Variational Autoencoder		

Table 12. Variables used in the study along with their meanings.

Variable	Meaning	Variable	Meaning
n	Passwords sample size	Z	Z-score
p	Proportion of the population	E	Margin of error
H	Entropy	HE	Expectation entropy
N	Number of possible characters in the space	L	Password length
p_L	Proportion of lowercase letters	l	Number of lowercase letters
p_U	Proportion of uppercase letters	u	Number of uppercase letters
p_D	Proportion of digits	d	Number of digits
p_S	Proportion of special characters	s	Number of special characters
K	Total character space	δ	Cliff's Delta
β_1	Decay rate for gradients in Adam	β_2	Decay rate for squared gradient in Adam
R_l	Reconstruction loss	X_i	True output
X_i'	Predicted output	S_l	Similarity loss
β	Scaling factor	ω	Variance
μ	Mean		

15, pages 286–302. Springer. DOI: 10.1007/978-3-642-15497-3$_1$8.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*. DOI: 10.48550/arXiv.1511.06349.

Chen, D., Chen, X., Li, H., Xie, J., and Mu, Y. (2019). Deepcpdp: Deep learning based cross-project defect prediction. *IEEE Access*, 7:184832–184848. DOI: 10.1109/access.2019.2961129.

Dastane, D. O. (2020). The effect of bad password habits on personal data breach. *International Journal of Emerging Trends in Engineering Research*, 8(10). DOI: 10.30534/ijeter/2020/538102020.

Dubey, R. and Martin, M. V. (2021). Fool me once: A study of password selection evolution over the past decade. In *2021 18th International Conference on Privacy, Security and Trust (PST)*, pages 1–7. IEEE. DOI: 10.1109/PST52912.2021.9647823.

Grilo, M., Campos, J., Ferreira, J. F., Almeida, J. B., and Mendes, A. (2022). Verified password generation from password composition policies. In *International Conference on Integrated Formal Methods*, pages 271–288. Springer. DOI: 10.1007/978-3-031-07727-2$_1$5.

Grobler, M., Chamikara, M., Abbott, J., Jeong, J. J., Nepal, S., and Paris, C. (2021). The importance of social identity on password formulations. *Personal and Ubiquitous Computing*, 25(5):813–827. DOI: 10.1007/s00779-020-01477-1.

Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., and Botvinick, M. M. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*. Available at:`https://openreview.net/forum?id=Sy2fzU9gl`.

Imamaliyev, A. and Khudoykulov, Z. (2021). Analysis password-based authentication systems with password policy. In *2021 International Conference on Information Science and Communications Technologies (ICISCT)*, pages 1–3. DOI: 10.1109/ICISCT52966.2021.9670312.

Jiang, J., Zhou, A., Liu, L., and Zhang, L. (2022). Omecdn: A password-generation model based on an ordered markov enumerator and critic discriminant network. *Applied Sciences*, 12(23). DOI: 10.3390/app122312379.

Kanta, A., Coisel, I., and Scanlon, M. (2024). A comprehensive evaluation on the benefits of context based password cracking for digital forensics. *Journal of Information Security and Applications*, 84:103809. DOI: https://doi.org/10.1016/j.jisa.2024.103809.

Kanta, A., Coray, S., Coisel, I., and Scanlon, M. (2021). How viable is password cracking in digital forensic investigation? analyzing the guessability of over 3.9 billion real-world accounts. *Forensic Science International: Digital Investigation*, 37:301186. DOI: 10.1016/j.fsidi.2021.301186.

Kekülüoğlu, D., Magdy, W., and Vaniea, K. (2022). From an authentication question to a public social event: Characterizing birthday sharing on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 16, pages 488–499. DOI: 10.1609/icwsm.v16i1.19309.

Kingma, D. P. (2013). Auto-encoding variational bayes. *arXiv*. DOI: 10.48550/arXiv.1312.6114.

Lee, K., Sjöberg, S., and Narayanan, A. (2022). Password policies of most top websites fail to follow best practices. In *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*, pages 561–580. Available at: https://www.usenix.org/conference/soups2022/presentation/lee.

Lykousas, N. and Patsakis, C. (2023). Tales from the git: Automating the detection of secrets on code and assessing developers' passwords choices. In *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 68–75. IEEE. DOI: 10.1109/EuroSPW59978.2023.00013.

Mannuela, I., Putri, J., Anggreainy, M. S., *et al.* (2021). Level of password vulnerability. In *2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI)*, volume 1, pages 351–354. IEEE. DOI: 10.1109/ICCSAI53272.2021.9609778.

Marchetti, K. and Bodily, P. (2022). John the ripper: An examination and analysis of the popular hash cracking algorithm. In *2022 Intermountain Engineering, Technology and Computing (IETC)*, pages 1–6. IEEE. DOI: 10.1109/IETC54973.2022.9796671.

Mayer, P., Zou, Y., Schaub, F., and Aviv, A. J. (2021). " now i'm a bit {angry:}" individuals' awareness, perception, and responses to data breaches that affected them. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 393–410. Available at:https://www.usenix.org/conference/usenixsecurity21/presentation/mayer.

Miessler, D. (2020). Seclists. Available at:https://github.com/danielmiessler/SecLists/blob/master/Passwords/.

Nisenoff, A., Golla, M., Wei, M., Hainline, J., Szymanek, H., Braun, A., Hildebrandt, A., Christensen, B., Langenberg, D., and Ur, B. (2023). A {Two-Decade} retrospective analysis of a university's vulnerability to attacks exploiting reused passwords. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5127–5144. Available at:https://www.usenix.org/conference/usenixsecurity23/presentation/nisenoff-retrospective.

Nugroho, A. and Mantoro, T. (2023). Salt hash password using md5 combination for dictionary attack protection. In *2023 6th International Conference of Computer and Informatics Engineering (IC2IE)*, pages 292–296. IEEE. DOI: 10.1109/IC2IE60547.2023.10331606.

Pal, B., Islam, M., Sanusi, M., Sullivan, N., Valenta, L., Whalen, T., Wood, C., Ristenpart, T., and Chatterjee, R. (2022). Might i get pwned: A second generation password breach alerting service. In *USENIX Security*. Available at:https://www.usenix.org/conference/usenixsecurity22/presentation/pal.

Parmar, V., Sanghvi, H. A., Patel, R. H., and Pandya, A. S. (2022). A comprehensive study on passwordless authentication. In *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, pages 1266–1275. IEEE. DOI: 10.1109/ICSCDS53736.2022.9760934.

Petkauskas, V. (2024). Rockyou2024: 10 billion passwords leaked in the largest compilation of all time | cybernews. Available at:https://cybernews.com/security/rockyou2024-largest-password-compilation-leak/.

Pimenta Rodrigues, G. A., de Oliveira Albuquerque, R., Gomes de Deus, F. E., de Sousa Jr, R. T., de Oliveira Júnior, G. A., Garcia Villalba, L. J., and Kim, T.-H. (2017). Cybersecurity and network forensics: Analysis of malicious traffic towards a honeynet with deep packet inspection. *Applied Sciences*, 7(10):1082. DOI: 10.3390/app7101082.

Pimenta Rodrigues, G. A., Marques Serrano, A. L., Lopes Espiñeira Lemos, A. N., Canedo, E. D., Mendonça, F. L. L. d., de Oliveira Albuquerque, R., Sandoval Orozco, A. L., and García Villalba, L. J. (2024). Understanding data breach from a global perspective: Incident visualization and data protection law review. *Data*, 9(2):27. DOI: 10.3390/data9020027.

Reaz, K. and Wunder, G. (2022). Expectation entropy as a password strength metric. In *2022 IEEE Conference on Communications and Network Security (CNS)*, pages 1–2. IEEE. DOI: 10.1109/CNS56114.2022.9947259.

Rehak, D., Slivkova, S., Janeckova, H., Stuberova, D., and Hromada, M. (2022). Strengthening resilience in the energy critical infrastructure: methodological overview. *Energies*, 15(14):5276. DOI: 10.3390/en15145276.

Remy, P. (2021). Name dataset. Available at: https://github.com/philipperemy/name-dataset.

Rodrigues, G. A. P., Serrano, A. L. M., Vergara, G. F., Albuquerque, R. d. O., and Nze, G. D. A. (2024). Impact, compliance, and countermeasures in relation to data breaches in publicly traded us companies. *Future Internet*, 16(6):201. DOI: 10.3390/fi16060201.

Romano, J., Kromrey, J. D., Coraggio, J., and Skowronek, J. (2006). Appropriate statistics for ordinal level data: Should we really be using t-test and cohen'sd for evaluating group differences on the nsse and other surveys. In *annual meeting of the Florida Association of Institutional Research*, volume 177.

Siponen, M., Puhakainen, P., and Vance, A. (2020). Can individuals' neutralization techniques be overcome? a field experiment on password policy. *Computers & Security*, 88:101617. DOI: 10.1016/j.cose.2019.101617.

Styoutomo, Y. A. and Ruldeviyani, Y. (2023). Information security awareness raising strategy using fuzzy ahp method with hais-q and iso/iec 27001: 2013: A case study of xyz financial institution. *CommIT (Communication and Information Technology) Journal*, 17(2):133–149. DOI: 10.21512/commit.v17i2.8272.

Tatlı, E. I. (2015). Cracking more password hashes with patterns. *IEEE Transactions on Information Forensics and Security*, 10(8):1656–1665. DOI: 10.1109/TIFS.2015.2422259.

Thai, B. L. T. and Tanaka, H. (2024). A statistical markov-based password strength meter. *Internet of Things*, 25:101057. DOI: 10.1016/j.iot.2023.101057.

Upadhyay, D. and Sampalli, S. (2020). Scada (supervisory control and data acquisition) systems: Vulnerability assessment and security recommendations. *Computers and Security*, 89:101666. DOI: 10.1016/j.cose.2019.101666.

van den Berg, J. (2024). Present-day cybersecurity: Actual challenges and solution directions. In Rath, D. M. and Samal, D. T., editors, *Key Issues in Network Protocols and Security*, chapter 0. IntechOpen, Rijeka. DOI: 10.5772/intechopen.1007021.

Walia, K. S., Shenoy, S., and Cheng, Y. (2020). An empirical analysis on the usability and security of passwords. In *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, pages 1–8. IEEE. DOI: 10.1109/IRI49571.2020.00009.

Wan, Z., Xia, X., Lo, D., and Murphy, G. C. (2019). How does machine learning change software development practices? *IEEE Transactions on Software Engineering*, 47(9):1857–1871. DOI: 10.1109/TSE.2019.2937083.

Wang, C., Jan, S. T., Hu, H., Bossart, D., and Wang, G. (2018). The next domino to fall: Empirical analysis of user passwords across online services. In *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, pages 196–203. DOI: 10.1145/3176258.3176332.

Wash, R. and Rader, E. (2021). Prioritizing security over usability: Strategies for how people choose passwords. *Journal of Cybersecurity*, 7(1):tyab012. DOI: 10.1093/cybsec/tyab012.

Weir, M., Aggarwal, S., Collins, M., and Stern, H. (2010). Testing metrics for password creation policies by attacking large sets of revealed passwords. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 162–175. DOI: 10.1145/1866307.186632.

Wheeler, D. L. (2016). zxcvbn:{Low-Budget} password strength estimation. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 157–173. Available at:`https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_wheeler.pdf`.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wesslén, A., *et al.* (2012). *Experimentation in software engineering*, volume 236. Springer. DOI: 10.1007/978-3-662-69306-3.

Wu, Y., Wan, X., Guan, X., Ji, T., and Ye, F. (2023). Pgtcn: A novel password-guessing model based on temporal convolution network. *Journal of Network and Computer Applications*. DOI: 10.1016/j.jnca.2023.103592.

Xiao, Y. (2024). Passrvae: Improved trawling attacks via recurrent variational autoencoder. In *In Proceedings of the 2024 3rd International Conference on Cryptography, Network Security and Communication Technology*. DOI: 10.1145/3673277.3673295.

Xu, M., Yu, J., Zhang, X., Wang, C., Zhang, S., Wu, H., and Han, W. (2023). Improving real-world password guessing attacks via bi-directional transformers. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 1001–1018.

Yang, K., Hu, X., Zhang, Q., Wei, J., and Liu, W. (2022). Vaepass: A lightweight passwords guessing model based on variational auto-encoder. *Computers and Security*. DOI: 10.1016/j.cose.2021.102587.

Yang, T. and Wang, D. (2024). Rankguess: Password guessing using adversarial ranking. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 40–40. IEEE Computer Society. DOI: 10.1109/SP61157.2025.00040.

Yu, W., Yin, Q., Yin, H., Xiao, W., Chang, T., and He, L. (2023). A systematic review on password guessing tasks. *Entropy*. DOI: 10.3390/e25091303.

Zhang, H., Wang, C., Ruan, W., Zhang, J., Xu, M., and Han, W. (2021). Digit semantics based optimization for practical password cracking tools. In *Proceedings of the 37th Annual Computer Security Applications Conference*, pages 513–527. DOI: 10.1145/3485832.3488025.

СЛАТВІНСЬКА and БЕВЗА (2024). ВПЛИВ ЗБОЮ crowdstrike НА МЕГА-ВИТІК ПАРОЛІВ: ЧИ Є ЗВ'ЯЗОК? Ч. 1. *Herald of Khmelnytskyi National University. Technical sciences*, 339(4):332–338. DOI: 10.31891/2307-5732-2024-339-4-52.