A comparison of password characteristics between RockYou and other password leaks

by

YUJIE DONG

A project submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Information Technology Security
- Artificial Intelligence in Security Field (MITS-AIS)

Faculty of Business and Information Technology
Ontario Tech University
Oshawa, Ontario, Canada
Nov 2020

ACKNOWLEDGMENTS

I would like to thank my professor for his help in preparation of this work – Dr. Miguel Vargas Martin, without whose instructions I would have been doomed to never complete it, GanYu Wang, who helped to shed new light on many of my ideas, Noopa Jagadeesh, by whom I was often received a lot of advice, Jana Kanagalingam, who has provided me with technical support.

Special thanks are given to the friends and colleagues who made this work possible.

Any views and conclusions contained herein are those of the author, and do not necessarily represent the official positions, express or implied, of the funders.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This project aims to study for the evolution of passwords. There are two data sets chosen for this purpose. They include a data set that is from the RockYou company leaked in 2009. Another data set is a leaked compilation of various password breaches over time [1]. One paper [1] this project mentions in the literature review has mentioned an article, 1.4 Billion Clear Text Credentials Discovered in a Single Database [2]. It also mentions that the database was recently updated with the last set of data inserted on 11/29/2017 [2]. There are still some data sets, of which leakage time and sources cannot be identified [2]. Therefore, the leakage time of various data sets in this compilation is until the end of 2017 or before 2018. The leakage of passwords related with the RockYou company happened in December 2009 [3]. The 1.4 Billion Password data set and the RockYou password data set can both be found online. This project thought of the possibility that the 1.4 Billion Password dataset contains the RockYou password leak. The 1.4 Billion Password package includes a file called imported.log, which lists a number of password leaks datasets, however RockYou is not in the list. Furthermore, this project also did a non-exhaustive cross-search to see if the RockYou passwords were included in the 1.4 Billion Password dataset but didn't find any match, except for the most common passwords, which are likely to appear in all, or most, password leaks datasets anyways. These common passwords include 123456, 123456789, qwerty, password, etc [4], [5]. Even if this project assumed that RockYou is not contained within 1.4 Billion Password, another consideration was that, if we wanted to do a comparison of password datasets through time, we needed to confirm that all the leaks in 1.4 Billion Password were older than RockYou (2009) but unfortunately, we don't have certainty as to the composition of the 1.4 Billion Password dataset, and thus we cannot make any assertions as to whether all the leaks in 1.4 Billion Password are newer than the RockYou leak. Furthermore, even if we assumed that the 1.4 Billion Password leaks were newer than 2009, it is obvious that the date of a leak is not representative of the time when the passwords in it were created. Therefore a comparison through time was not feasible for this project. In the remaining of this project, this project will cautiously assume that RockYou is not in the 1.4 Billion Password dataset, and will make no time-related assumptions.

1.1 Contributions

There are several contributions this project have made here that are:

1. A large scale comparison of two password datasets in terms of password composition characteristics
2. Such comparison shades new light as to the strength evolution of passwords over time.
3. A comparison of the strength of the passwords in both datasets

1.2 Overview of Methodology

The methodologies used here is analyzing the components of passwords by the amounts of numbers, upper cases, special symbols. It is also through test samples zxcvbn scores and the value of T-tests. After that, machine learning methods are utilized to predict or generate something useful.

1.3 Overview of Findings

That general strengths of RockYou 2009 and 1.4 Billion Password 2017 are similar. Meanwhile, this project also found that as long as passwords containing at least one number, one upper case or one special symbol, the passwords until the end of 2017 is much stronger than in 2009. The RNN-LSTM (recurrent neural network -long short-term memory) can generate some meaningful passwords, which can be used as the substitutes of the weak passwords in RockYou 2009.

CHAPTER 2
LITERATURE REVIEW

This project will apply zxcvbn algorithm, tensorflow keras (recurrent neural network) that are also recommended by the papers to the analysis and prediction in capstone project II. These parts can also be seen in the following paragraphs.

2.0.1 Guessing, Modelling and Machine Learning

Melicher et al. [6] studied modeling password guessability using neural networks. Artificial neural networks is applied into guessing passwords, with the goal of gauging the strength of human-chosen text passwords. The neural networks are leveraged to create a password-guessing model followed by a benchmark checker for the passwords. Probabilistic context-free grammars are built with template structures(e.g., 5 letters with 3 digits), which are to check the possibility of a password. The possibility is relying on efficient heuristics to model common password characteristics. However, a real-time password checkers entirely client-side is needed for accurate strength estimations. Guessing attack here is to calculate the probability to predict the next character of a password fragment. In this method, they have to tokenize the password and generate the next character in the context of character-level natural language. Transference learning can be utilized here to recognize different data sets. Precomputation is utilized for optimizing for latency, which greatly increases the efficiency.

Xia et al. [7] studied GENPass, which is a multi-source deep learning model for password guessing. Users prefer setting their passwords with short and easy-to-memory passwords, such as iloveyou, password123, and 123456. PCFG (probabilistic context-free grammar) rules change the passwords to be represented by L (letters), D (Digits), S (Special Characters) and numbers standing for the length. The results showed that substrings with length 4 are most used, followed by length 7 and 8. Compared with English users, Chinese prefer digits to set passwords. Pinyin is also most Chinese choices for their passwords, which can be used for PCFG to increase the accuracy of password guessing. The model of GENPass consists of a one-site test and a cross-site test, which try to build a new general model.The problem of neural networks is that it also contains many duplicates in this neural networks. Using two-character sequences in stead of one character can be a choice to overcome this shortcoming. Model PCFG+LSTM (PL) contains preprocessing, generating and weight choosing, which also preplaces letters, digits, and special chars with tags. Another model is GENPass containing prediction of model n, weight choosing and classifier.

Hitaj et al. [8] studied PassGAN, which is a deep learning approach for password guessing. One part of PassGAN is to learn the password from actual password and generate high-quality passwords. PassGAN is to augment the pass- word rules and showed the password generated. PassGAN can output many passwords and this can also help the users to understand the strength of the passwords deeply. PassGAN includes Conv 1D, input and output. PassGAN comprises batch size, number of iterations, number of discriminator iterations per generator iteration, model dimensionality, gradient penalty coefficient, output sequence length and size of the input noise vector(seed) and maximum number of examples. PassGAN will be at a risk of overfitting, if the step has been increased. Meanwhile, different algorithms can also be combined , such as combining PassGAN with HashCat or FLA. Finally, it found that the combination with HashCat can achieve a better result. Generally, character-level GANs can excel in guess the passwords through generating them. Efficient Rule-based password guessing has its limitation so that multiple tools can be applied in practice. GANs can still perform better than other algorithms.

2.0.2 Cracking, Modelling and Machine Learning

Ciaramella et al. [9] studied neural network techniques, which are for proactive password checking. This paper has classified the week characters and digits and strong characters. A pattern recognition are composed of a feature extractor and a classifier (a discriminant function). The neural networks comprises supervised learning and unsupervised learning, in which the networks are known and unknown. Four features are used to classify the passwords, including classes, strong characters, upper-lower distribution and diagrams.

2.0.3 Strength, Modelling and Machine Learning

Pasquini et al. [10] studied interpretable probabilistic password strength meters via deep learning. Probabilistic password strength meters have been proved to be the most accurate tools to measure password strength, by which the password is disentangled and to provide immediate feedback for the user. An efficient and lightweight deep learning framework suitable for client-side operability is implemented. Password features such as LUDS (which counts lower, upper letters, and symbols). It reminds the user which part of the password is weak so that it can change the passwords little by little through the character-level feedback mechanism. Many symbols are developed to replace some certain characters for the increment of password strength, which is from an applied perturbation.

He et al. [11] studied group password strength meter based on attention mechanism. AM LSTM is called as attention mechanism long short-term memory, in which the memory can remember and react to some of the passwords existing in the previous period. Most of the data sets adopted from this paper are from China, which shows that date and name occupy most parts in the components of passwords. For popular group passwords analysis, most popular passwords are purely numbers. For the password strength meter process, the process comprises offline training and online evaluation. In offline training, it starts in password evaluation preparation. Password evaluation preparation starts getting group information, coding group information and vectorize group information. Meanwhile, password data set is preprocessing screening, encoding password text sequence and vectorize password text sequence. After password evaluation preparation, it can be judged that it is trained by AM-LSTM model. The performance of AM-LSTM password metrics is out- performed than PCFG (probabilistic context-free grammar) password metrics and NST password metrics, especially on Mailbox.

Daniel et al. [1] studied password similarity models by using neural networks. The authors proposed the personalized password strength meters, which is less than 3MB and estimate the complexity of passwords accurately. Natural language processing is a technique to build interactions between humans and computers. Natural language processing (NLP) can analyze passwords generated from the choice of human. Recurrent neural networks (RNNs) and deep generative adversarial networks (GAN) can be utilized to generate password models. PSM can be applied to check the similarity between the sample and other various passwords. There are two models, including a generative model and a similarity model. After cleaning in the data, the distribution of password length and composition is that length 3-5 is 2 %, 6-8 is 48 %, 9-12 is 40 % and 13-50 is 10 %. To get a clean data set, it is necessary to eliminate duplicated passwords, which are email based, username based and mixed method. The authors choose the mixed method to improve the accuracy finally. The model as a supervised learning task can learn the structure of the password and calculate possibility. Reordering the password can result in more password-pairs which are from the input password. There is also attack efficacy calculated in the model. If the password are reused, it will be targeted easily by credential stuffing. There is one kind of L8C3 password including 8 characters. Upper-case letters, digits and lower-case letters constitute L8C3 password. The password similarity model is combined with word embedding techniques. Similarity scores can be an effective tool for binary classification. The guessing attack can be also used for classification also. Sequence-to-sequence learning can be utilized in a generative model and the embedding techniques can be utilized in a discriminative model. Different data sets are applied to get the passwords from different data sets, such as alumni, staff, faculty and other(current students, contract workers, and affiliates).

Rathi et al. [12] studied a comparative analysis of soft computing techniques for password strength classification. These computing techniques includes back propagation neural network (BPN), logistic regression, hopfield neural network (HNN), brain state in a box (BSB) and associative memory network (BAM), convolution neural network (CNN) system. Hopfield neural network has one layer in which every node is connected to every other node with a link. Back propagation neural network includes input layer, hidden layers and output layer. Passwords can be categorised as weak, medium and strong. Through predicting whether the password used is secure, they can strengthen their passwords. Finally, logistic regression is a statistical model to build a model to predict strength of password. Convolutional neural network is utilized for password strength detection and generation. As a result, the accuracy found is different from each other. The highest is LR (logistic regression), which is 81 % and the lowest is BSB, which is 68 %. CNN is not applicable.

Vijaya et al. [13] studied password strength prediction using Supervised Machine Learning Technique. The machine learning techniques to measure the password strength prediction include multilayer perceptron network (MLP), decision tree classification, naive bayes classifier (NB), and support vector machine. Except choosing the algorithm, feature selection is also significant to get the accurate result more efficiently. Password strength can use 1, 2, 3, 4, 5 to describe the passwords from very weak to very strong.

2.0.4 Password Strength Meter Without Machine Learning

Wang et al. [14] studied fuzzyPSM, which is a new password strength meter using fuzzy probabilistic context-free grammars. There is a distance between the machine-generating passwords and the realistic passwords. Modelling realistic behaviors of users can achieve a better effect when applied. There is much valuable information that can be utilized in the further researches. Most of the users would like to reuse an existing password or modify an existing password rather than creating an entire new one. To increase security is accountable for the half of the purposes to modify an existing password for the new account. However, it is also because of policies enforced and for better memorability for the total 75.34 % people.Adding digit and symbol at the beginning/end will be the first choice when users modify their existing passwords. If the new passwords involving capitalization is mandatory, half of the users will put the capitalized letter at the beginning. In the PCFG-based model, the vulnerable passwords will be improved by changing the lower case into special characters. All of them will also be titled with the probability for the changed version and unchanged version.

Galbally et al. [15] studied a probabilistic framework for improved password strength metrics. Passwords guessing attacks can be as a tool to gauge the strength of a password. The guessing attacks are composed of dictionary attacks and brute-force attacks. Some passwords resistant to dictionary attack may be easily cracked by brute-force attacks because of the limitation of dictionary. The occurrence probability of a password can be calculated by each single character. sMC and lMC both were used to gauge the strength of the broken and non-broken password sets. These new metrics can also distinguish different passwords strength to aid users to pick the stronger passwords.

Kelley et al. [16] studied measuring password strength by simulating password-cracking algorithms. This paper mentions the relationship between the resistance of passwords to the cracking algorithms with the strength and password. It also includes the relationship between the entropy estimates with the strength of the passwords. The entropy is a concept also utilized in the other science, technology, engineering, and mathematics (STEM) areas. Entropy is used to describe the degree of chaos of the password. The guessability represented by guess number can also be one of metrics of the strength of a password. In the experiment done by authors, conditions are basic8survey, basic8, basic16, dictionary8, comprehensive8, blacklistEasy, blacklistMedium and blacklistHard. Calculators especially BFM calculator (brute-force Markov) can be used to calculate how many times the cracking algorithms will spend on cracking password. Meanwhile, Weir algorithm calculator can also be another algorithm to calculate the probabilities of different structures, whether the next character should be the letter, digit or symbols. Above all, basic16 is the easiest way.

Taha et al. [17] studied password strength measurements by password entropy and password quality. Password entropy is can be the standard of password quality. High entropy means high quality and low entropy means low quality. High entropy always means more uncertain and random.After using n represents number, s represents special characters and a represents small letters, it seems that more combinations mean higher entropy values. For example, $n+s+a$ can achieve higher entropy value than just $s+a$. Entropy Based checker can be one of the password checkers.

Guo et al. [18] studied LPSE, which is lightweight password-strength estimation for password meters. The LPSE algorithm is a lightweight password-strength estimation method and the background for developing the password metrics is to visualize the strengths of user-chosen passwords. It is a serious problem that LPSE labels the weak password as strong and labels the strong password as weak password. Comparing with other evaluation methods, similarity-evaluation method can efficiently seize the characteristics between the weak passwords and the strong passwords. As mentioned by the previous papers, there are two primary ways to gauge the password, which consists of evaluation of the password patterns and using password- guessing algorithms. An easiest way to mark every password through different scores. These scores are judged by the complexity of passwords, including digits, upper and lower cases and special characters. There is another solution that is probabilistic context-free grammar constructed by the leaked passwords first. Second, it is to get the generated rules from the trained context-free grammar, which can be used for further usage. These guessing attacks can be executed by an artificial neural. Meanwhile, the neural network can occupy several hundred kb storage, which can save the storage resource. Rule-based algorithms consume very large disk space, so a similarity-evaluation method that can run on the client side is proposed in this paper. For LPSE algorithm, the vector values of digits, lowercase letters, uppercase letters and special characters are 1, 1, 2 and 3 in sequence. If it is a two-letter phrase, the values of them will be added with each other and accumulated. The common weak password patterns also include two-letter combinations and three-letter combinations in English. For example, the vector values of 2-letter and 3-letter can be calculated the way as a single letter. The password-strength evolution includes cosine-length and edit-distance similarity. LPSE has a better effect to identify strong/weak passwords than PM and zxcvbn as the final results show.

2.0.5 Password Strength Estimation (combined methods)

Galbally et al. [19] studied a new multimodal approach, which is for password strength estimation. A multimodal approach is utilized to combine different algorithms to overcome the weaknesses generated from these algorithms. Categorizing the passwords into trivial and non-trivial passwords can greatly improve the efficiency to classify the passwords in the further step. Password cracking algorithms entail lists of popular passwords, brute-force, dictionary based and probabilistic. Lists of popular passwords, brute-force are the methods to crack trivial passwords. Dictionary based and probabilistic attacks are the ways to crack non-trivial passwords. Password strength estimation algorithms can be attack-based, heuristic-based and probabilistic-based. Probabilistic-based algorithms can be used to deal with non-trivial passwords. Heuristic-based and attack-based algorithms can deal with trivial passwords. It builds the bridge between password cracking algorithms and password strength estimation algorithms. If multimodal password strength estimation method is applied, it is combined with several algorithms, including attack-based algorithms, heuristic-based algorithms and probabilistic-based algorithms. This combined methods can overcome the every single algorithm that has to process different passwords with different algorithms. The attack based and heuristic-based algorithms are applied into dealing with trivialized passwords.The adaptive memory Markov chain and hierarchical markov chain can be utilized into dealing with non-trivialized passwords.

2.0.6 Others

Olade et al. [20] studied SemanticLock, which is an authentication method for mobile devices using semantically-linked images. SemanticLock can be utilized in mobile devices as a simple, fast, memorable and graphical authentication approach. There are two other ways, such as a personal identification number (PIN) or a pattern. A pattern is to draw several lines among nodes. The problems caused by two ways is performing a series of experiments and analysis to judge which images and image pairs users prefer and then select images. These images can avoid any type of explicit or implicit bias, resulting in an effective password essentially same as the total password space. SemanticLock performing similarly to PIN and Pattern in usability has significantly increased memorability and security. Fingerprint or face recognition are vulnerable to spoofing. However, the user would like to choose a poor password if the device did not include a biometric authentication mechanism. SemanticLock can balance the security and easy-to-remember while the actual space of passwords choose is much smaller than the total space available. Graphical system is much more popular, graphical authentication system that has been developed because of this. The usability and memorability comparison among SemanticLock, PIN and Swipe can figure out that usability study of SemanticLock can be long-term and daily use.

Veras et al. [21] studied visualizing semantics in passwords, which is the role of dates. Semantic patterns are underlying user choice in passwords. However, these patterns are hard to recognize so that the visualization has been introduced to analyze. Meanwhile, the RockYou dataset has been utilized in this paper. The semantic meaning can aid people to remember their passwords while it also greatly creates the potential to crack the passwords by the guessing attack. The author would like to find the semantic patterns which do not cause security vulnerabilities. These can also guide the new usage of successful password. Numbers are commonly the choices while dates are common amongst 4- digit sequences. For the processing of the data, passwords containing sequences of at least 4 digits occupies 24 %. Passwords containing 5-8 consecutive digits is 15.29 %. Visualization is to guide the investigation, facilitate exploration of diverse scenarios and easily accessible. There are also many users that would like to choose valentine's day, Christmas day involved in their passwords.

CHAPTER 3

METHODOLOGY

3.1 Description of Data Sets

RockYou data set was from the RockYou company leaked in 2009. 1.4 Billion Password data set dump firstly was found in in an underground community forum in 2017 [2]. After both datasets have been processed, the effective passwords of 1.4 Billion Password data set 2017 is 1,342,406,300. The number of RockYou data set passwords is 14,316,560. RockYou password data set is composed of plaintext passwords as well as passwords to connected accounts at partner sites (including Facebook, Myspace, and webmail services) [22]. This data sets aggregates 252 previous breaches, including Anti Public, Exploit.in, Bitcoin, Pastebin sites, LinkedIn and other known credential lists [2]. They also include many email passwords. Therefore, these two data sets can represent the hobby of people in using passwords in 2009 and until the end of 2017.

3.2 Description of Categories: Numbers, Upper Cases and Special Symbols

To check the strengths and formats of the passwords, we need to check numbers, upper cases and special symbols under every condition.

In the following paragraphs, numbers will be represented by N, upper cases will be represented by U and special symbols will be represented by S. There is also other items needed to be explained here. The password[0] denotes the first character, where the first character is indexed as 0. Meanwhile, password[len-1] or password[length-1] denotes the last character. The password[1] denotes the second character, where the second character is indexed as 1. Meanwhile, password[length-2] denotes the last second character.

There is an example explained here. Password is IloveCanda2020. Therefore, password[0] is . Password[length -1] is 0. Password[length-2] is 2. Password[1] is I. The numbers of this password are 2, 0, 2, 0. The value of Numbers (N) in this password is 4. The Upper Cases are I, C and the special symbol is #. The value of Upper Cases (U) in this password is 2. The value of special symbol is 1.

3.2.1 N, U, and S in password[0] and password[length-1]

This method is to analyze the amount of the numbers. Upper cases and special symbols the first character and the last character both in RockYou 2009 and 1.4 Billion Password samples.

3.2.2 N, U, and S from password[1] to password[length -2]

This method is to analyze the amount of the numbers, upper cases and special symbols from the first character to the last second character both in RockYou 2009 and 1.4 Billion Password samples.

3.2.3 N, U, and S from password[0] to password[length-1]

This method is to analyze the amount of the numbers, upper cases and special symbols from the first character to the last second character both in RockYou 2009 and 1.4 Billion Password samples.

3.3 zxcvbn Score

zxcvbn is a password strength estimator inspired by password crackers. Through pattern matching and conservative estimation, it recognizes and weighs 30k common passwords, common names and surnames according to US census data, popular English words from Wikipedia and US television and movies, and other common patterns [23].

This project classify the data sets into different arrays. We can see there are seven different arrays. The first array contains the passwords that at least have one upper-case character. The second array contains the passwords that at least have one number. The third array contains the passwords that have one special symbol. The fourth array contains the passwords that at least have one number and one upper-case character. The fifth array contains the passwords that at least have an upper-case character and a special symbol. The sixth array contains the passwords that at least have an number and an special symbol. The seventh array contains the passwords that at least have one number, one upper-case character and one special symbol.

3.3.1 Two 3000 samples from RockYou 2009 and 1.4 Billion

To estimate the strength of passwords, this project utilized zxcvbn to test each password. This project chose random samples from RockYou data set and 1.4 Billion data set. There are 1.4 billion data set allocated in 27 folders and this project chose 357MB passwords from these 1.4 billion data set. this project chose the sequence from 0 to 1000, from 5000000 to 5001000 and from 10000000 to 10001000 from each data set. Therefore, the total length of either the dataset from RockYou or 1.4 billion is 3000.

3.3.2 Two 300,000 samples from RockYou 2009 and 1.4 Billion

To get a more accurate result, this project enlarged the amount of my samples to 300,000 samples without distinguishing numbers, upper cases and special symbols. The 300,000 samples are the samples from 0 to 100000, 5000000 to 5100000 and 10000000 to 10100000.

3.3.3 Two 3 million samples from RockYou 2009 and 1.4 Billion

This project increased the amount of the samples to 3 million. The 3 million samples are the samples from 0 to 1000000, 5000000 to 6000000, 10000000 to 11000000.

3.3.4 Every 3000 sample in every category

To get a more accurate result, this project sets standards to choose 3000 samples containing at least one upper-case character, 3000 samples containing at least one number, 3000 samples containing at least one special symbol, 3000 samples containing at least one upper-case character and one number, 3000 samples containing at least one upper-case character and one special symbol, 3000 samples containing at least one special symbol and one upper-case character and 3000 samples containing at least one special symbol, one upper-case character and one number. These samples are chosen separately from RockYou data set and 1.4 Billion data set in sequence from the beginning to the end .

3.4 t-Test

t-Test is for the means of two independent samples of scores and the average values of these two independent samples are identical [24]. The results of t-Test are usually composed of two parts, including statistic,pvalue. P-value illustrates null hypothesis. Null hypothesis shows the lack of a difference. Higher p values mean it is more likely with a true null, a higher possibility of the lack of a difference so that these two sample data sets are more similar. Low p values means they are not similar. Usually, if a p value is bigger than 0.05, it is a high p-value [25]. As for the following samples, this project has separated the data set into two data sets as two samples.

3.4.1 RockYou 2009 and 1.4 Billion Passwords 2017 from password[1] to password[length-2]

The ratios of numbers, upper-case characters and special symbols in RockYou 2009 and 1.4 Billion Passwords are chosen for comparison .

3.4.2 1.4 Billion from password[1] to password[length-2]

According to the alphabetical order in 1.4 Billion Passwords, this project has set the data sets from a folder to the a-half folder in m foler and 0-9 folers before a folder as the first sample. The folder from the a-half2 folder in m folder to z folder as the second sample. In these two samples, this project chose the same last rows about ratios of number, upper-case characters and special symbols to test whether they are same or not.

3.4.3 RockYou 2009 and 1.4 Billion password data sets from password[0] to password[length-1]

The different lengths of number, upper-case characters, special symbols and passwords are used here.

3.5 Machine learning utilization

After the data analysis, this project tries to use different AI tools to predict the character after another character and get other meaningful results, which might be useful in societies.

3.5.1 KNeighborsClassifier

This project has made the classification by giving numbers 0 to 9 labes from 0 to 9 and lower-case characters a to z from 10 to 35. After that, this project chose random passwords from 1.4Billion passwords.

3.5.2 RNN

This project imported a password file and sorted this text. This process find the unique characters, which is applied in the file, such as !, +. Meanwhile, we will assign some certain values to them. For example, ! is assigned with 2 and + is assigned with 8. Then a model was built to process the data. This model used in my paper is tensorflow keras, embedded with vocabulary size, embedding dimension and batch input shape. The running units of this model is 1024. The batch size is 64.

Predict the next character by an untrained model

The input size limitation is 100. The following picture shows input examples and the output examples. By this model, we have the input words and next character predictions.

RockYou

train the whole RockYou data set (epochs 10), 50 to 60 hours
After training all the passwords of RockYou data set, these are the passwords generated, which can be referenced by users.

train the number-containing password array from RockYou, epochs = 5000 and epochs = 20000

This is to generate the passwords that is from a RockYou 2009 password array, in which every password contains at least one number. This project chose the first 2000 passwords from the array. The length of passwords is between 8 and 20.

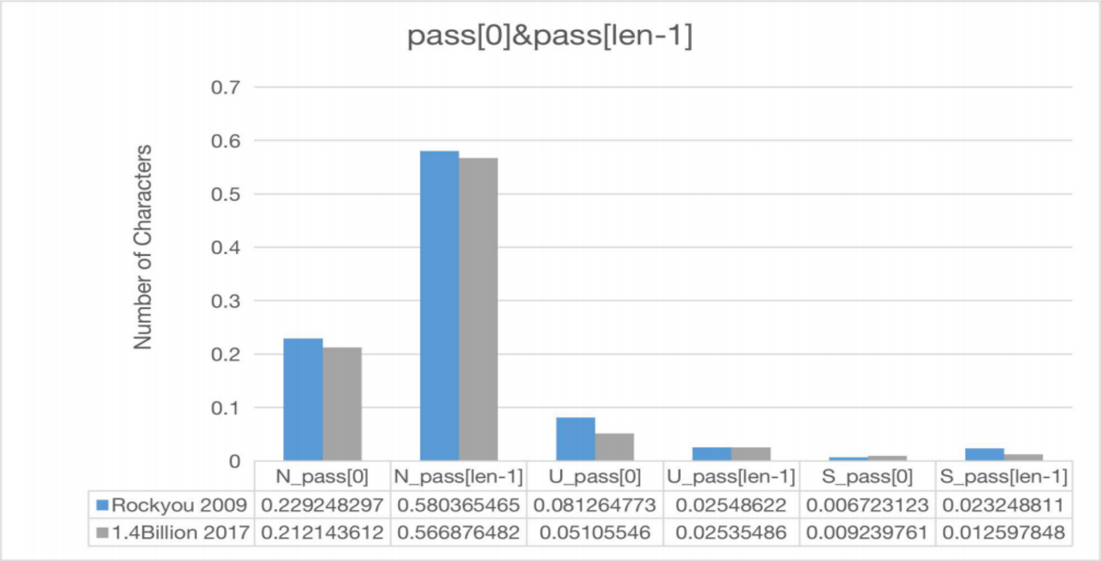
CHAPTER 4
RESULTS

4.1 Description of Categories: Numbers, Upper Cases and Special Symbols

N pass[0] representing the average value of first character of passwords is a number. N pass[len-1] representing the average value of last character is a number. U pass[0] representing the average value of first character of passwords is an upper case. U pass[len-1] representing the average value of last character is an upper case. S pass[0] representing the average value of first character of passwords is a special symbol. S pass[len-1] representing the average value of last character is a speical symbol. N Average User stands for how many numbers will be used by users. U Average User stands for how many upper cases will be used by users. S Average User stands for how many special symbols will be used by users. len pass ave User stands for what will be the average length of passwords chosen by the users.

4.1.1 N, U, and S in password[0] and password[length-1] (Figure 4.1)

In Rock 2009, users would averagely use 0.229248297 number, 0.081264773 upper case or 0.006723123 special symbol at the beginning of passwords. They also put averagely 0.580365465, 0.02548622 or 0.023248811 at the end of passwords. Until the end of 2017, the users chose 0.212143612 number, 0.05105546 upper case or 0.009239761 special symbol at the beginning of passwords. They would put 0.566876482 number, 0.02535486 upper-case character or 0.012597848 special symbol at the end of passwords.This shows that during several years, the choices of users for numbers, upper-case characters at the beginning of passwords have declined by 0.017104685 and 0.030209313, but their choice for special symbols has increased by 0.002516637. At the end of the passwords, after 8



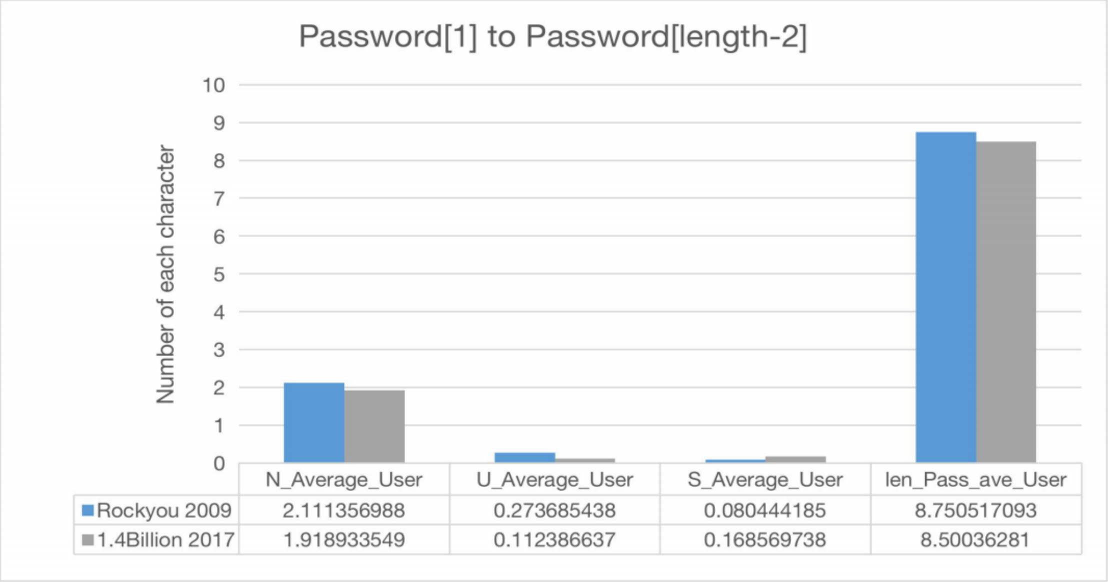
	N_pass[0]	N_pass[len-1]	U_pass[0]	U_pass[len-1]	S_pass[0]	S_pass[len-1]
■ Rockyou 2009	0.229248297	0.580365465	0.081264773	0.02548622	0.006723123	0.023248811
■ 1.4Billion 2017	0.212143612	0.566876482	0.05105546	0.02535486	0.009239761	0.012597848

Figure 4.1: password[0] and password[length-1]

years, the ratio of choosing a number, an upper-case character or special symbols has been shortened by 0.013488983, 0.00013136 and 0.010650964. All the detailed analysis results and codes can be viewed on one GitHub branch [26].

4.1.2 N, U, and S from password[1] to password[length -2] (Figure 4.2)

The average length of RockYou 2009 data set is 8.750517093. From second character to last second character, the user will averagely use 2.111356988 numbers, 0.273685438 upper-case character and 0.080444185 special symbol. The average length of 1.4 Billion Passwords until the end of 2017 is 8.50036281. From second character to last second character, the user will use 1.918933549 numbers, 0.112386637 upper-case character and 0.168569738 special symbol. This shows that after 8 years, the average length of the numbers used in per password from second character to last second character had decreased 0.192423439. the average length of the upper-case characters used in per password had decreased 0.161298801. However, the average length of the special symbols had increased 0.088125553. The length of the whole passwords have been shortened 0.250154283. All the detailed analysis results and codes can be viewed on my GitHub page [27].

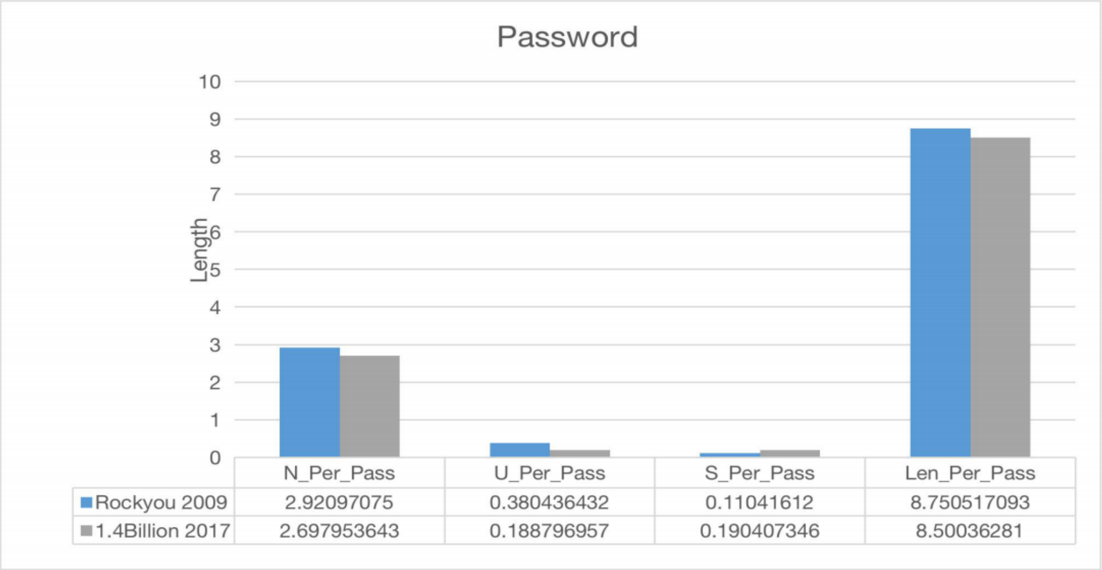


	N_Average_User	U_Average_User	S_Average_User	len_Pass_ave_User
■ Rockyou 2009	2.111356988	0.273685438	0.080444185	8.750517093
■ 1.4Billion 2017	1.918933549	0.112386637	0.168569738	8.50036281

Figure 4.2: Pass[1] to Pass[len-2]

4.1.3 N, U, and S from password[0] to password[length-1] (Figure 4.3)

As for RockYou 2009 pasword data set, from first character to last character, the user averagely used 2.92097075 numbers, 0.380436432 upper-case character and 0.11041612 special symbol. As for 1.4 Billion Password data set until the end of 2017, from first character to last character, the user averagely used 2.697953643 numbers, 0.188796957 upper-case character and 0.190407346 special symbol. These results illustrate that after 8 years, the average length of the numbers used in per password from first character to last character had decreased by 0.223017107 (2.92097075-2.697953643) . The average length of the upper-case characters used in per password had been reduced by 0.191639474 (0.380436432-0.188796957). However, the average length of the special symbols had been added by 0.079991227 (0.11041612-0.190407346). The length of the whole passwords have been shortened 0.250154283 (8.750517093-8.50036281). All the detailed analysis results and codes can be viewed on one GitHub branch [28].



	N_Per_Pass	U_Per_Pass	S_Per_Pass	Len_Per_Pass
🟦Rockyou 2009	2.92097075	0.380436432	0.11041612	8.750517093
⬜1.4Billion 2017	2.697953643	0.188796957	0.190407346	8.50036281

Figure 4.3: Password

4.2 zxcvbn Score

4.2.1 Two 3000 samples from RockYou 2009 and 1.4 Billion

The score of random samples of 1.3 billion data set is 1.625333. The score of random samples of RockYou data set is 2.076667. Figure 2.4 is about the arrays from RockYou data set in 2009. Figure 2.5 is about the arrays from the samples of 1.4 Billion Password data sets.

4.2.2 3000 samples

RockYou zxcvbn	
length of data set:	3000
U array1 zxcvbn, length of passwords:	2.8, 10
N array1 zxcvbn, length of passwords:	2.1563604240282683, 1132
S array1 zxcvbn, length of passwords:	2.9514563106796117, 103
N U array1 zxcvbn, length of passwords:	3.0, 2
U S array1 zxcvbn, length of passwords:	4.0, 1
N S array1 zxcvbn, length of passwords:	3.150943396226415, 53
N U S array1 zxcvbn, length of passwords:	0, 0

4.2.3 300,000 samples

1.4 Billion zxcvbn 300,000	
zxcvbn pswd, length of passwords:	1.66183, 300000

RockYou zxcvbn 300,000	
zxcvbn pswd, length of passwords:	1.7364833333333334, 300000

1.4 Billion zxcvbn 3 million	
zxcvbn pswd, length of passwords:	1.6556763333333333, 3000000

RockYou zxcvbn 3 million	
zxcvbn pswd, length of passwords:	1.7946386666666667, 3000000

The zxcvbn scores between RockYou samples and 1.4 Billion 2017 samples are completely different, which can be seen in figure 2.10 and figure 2.11. In both samples containing at least one upper-case character, the average socres of 1.4 Billion 2017 samples are almost 2.00, which is almost 1.4297 more than RockYou 2009 data. In both samples containing at least one number, 1.4 Billion 2017 samples are almost 1.599, which is almost 0.691 more than RockYou 2009 data set. In both samples containing at least one special symbol, 1.4 Billion 2017 samples is 2.64, which is almost 1.255 more than RockYou data set. In both samples containing at least a number and an upper-case character, 1.4 Billion 2017 password samples are almost 2.17, which are 1.153 more than RockYou data set. In both samples containing at least one upper-case character and one special symbol, 1.4 Billion 2017 password samples are 2.606, which are 0.746 more than RockYou data set. In both samples containing at least one number and an speical symbol, 1.4 Billion 2017 password samples are 2.78, which are 1.276 more than RockYou data set. In samples containing at least one number, one upper-case character and one special symbol, 1.4 Billion

2017 password data set are 2.613, which are 0.483 more than RockYou data set.

4.3 t-Test

4.3.1 RockYou 2009 and 1.4Billion Passwords 2017 from password[1] to password[length-2]

The result of t-Test is statistic=-0.07525419439279683, pvalue=0.9436258458447557. This pvalue is 0.893625846 higher than 0.05 (0.9436258458447557 - 0.05).

4.3.2 1.4 Billion from password[1] to password[length-2]

The t-Test results for this are statistic = -0.1616760624684957, pvalue = 0.871757568924027. This pvalue is 0.821757569 higher than 0.05 (0.871757568924027 - 0.05).

4.3.3 RockYou 2009 and 1.4Billion password data sets from password[0] to password[length-1]

t-Test results are that statistic is 0.05213376224556177 and pvalue is 0.9601144784877369. This pvalue is 0.910114478 higher than 0.05 (0.9601144784877369 - 0.05).

4.4 Machine Learning Utilization

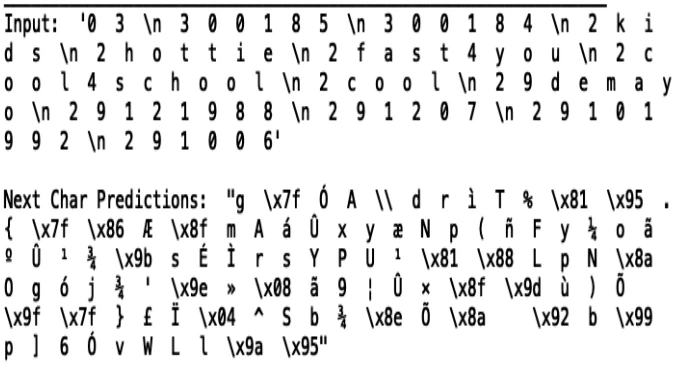
4.4.1 KNeighbors classifier

The technical details of this model is a 357 MB random sample file with both email names and passwords from 1.4 Billion. The final predicting precision rate by using KNeighborsClassifier is 0.0869687184782414, which is equal to 8.69687184782414 percent.

4.4.2 RNN

Predict the next character by an untrained model

The technical details of this model is RockYou data set, which is 139.9 MB and includes only passwords. The result is Figure 4.4.



```
Input: '0 3 \n 3 0 0 1 8 5 \n 3 0 0 1 8 4 \n 2 k i
d s \n 2 h o t t i e \n 2 f a s t 4 y o u \n 2 c
o o l 4 s c h o o l \n 2 c o o l \n 2 9 d e m a y
o \n 2 9 1 2 1 9 8 8 \n 2 9 1 2 0 7 \n 2 9 1 0 1
9 9 2 \n 2 9 1 0 0 6'

Next Char Predictions: "g \x7f Ó A \\ d r ì T % \x81 \x95 .
{ \x7f \x86 Æ \x8f m A á Û x y æ N p ( ñ F y ¼ o ã
º Û ¹ ¾ \x9b s É Ì r s Y P U ¹ \x81 \x88 L p N \x8a
0 g ó j ¾ ' \x9e » \x08 ã 9 ¦ Û × \x8f \x9d ù ) Õ
\x9f \x7f } £ Ï \x04 ^ S b ¾ \x8e Õ \x8a   \x92 b \x99
p ] 6 Ó v W L l \x9a \x95"
```

Figure 4.4: First Character and next character predictions

RockYou

train the whole RockYou data set (epochs 10), 50 to 60 hours, the generated passwords

RMEO: RELOLOVER	
*Po*PinP*****	*PuPPine*
*PurSesoo	*PPRPPELLP
*PPPPL	*PPPTEOPPIEPPPPPPPPPPYPPPPRORPKPERTYPP
*PPPPP	OPPPPP*
*PPPPPPPPPP	*PPPPEPP
*PPPPP#poPePRPPP	*PPPPPPPPPR**8PoPPPPPPPP
*PPPIPPIRPP	*PPPSPr
*PPPPPPPPKPPPPPPPP	*PPPPPP
*PPPRY1POPPPPP	*PPPPIP1PPPP*
*PPPP*PPPPPPPPOPP	*PPPPPP

*PPRiPP*PP	*POOPEPIPPPANSOPPAKIS*1131
*PPRPPP2	*PPPPPPP
*PPPP*PRIEPPP1Ps	*PPRRPPOPER!
*PPRPPM	*PPRROOSP
*PPPPrieP	*PPROPPRSP
*PEPPFON	silis950123
000086303	004549764
&cirshfle3)	(cc030005
(cc00205)	**yspakeyabr91litars*
**ysly00*mol	**hwnidsp***
**rgiot84*	**rgiot84*
**rise87813*	**luff*
**Obroursh*	**382SEP&*
**UTPuny*	**5LONA.VERANAE2
**LINPOOLONE/*/	//09/gmaceswab//
//2eyeu7	//vahouaca8087
/Q25147110	//linesucmc //////-ximo-0
//lizanw3/////	/*/2/////*//-/92///XXXXu3ama1//
/*//23d	/I/diede
/12AM....xx¡.	.::/:::)
(Medatock)	.1478123
.1478074	.147832443
0047504451	0045563181
0045567050	0045433
00454172	00455329

train the number-containing password array from RockYou, epochs = 5000 and epochs = 20000

epochs = 5000, the generated passwords	
debbie11fBLEErO9	N8j8OoSw7eyr
hLJmvonfaf5Jqr5	XszyI1073CxrsA
1OazszvI1w8rBR	2Xuo4uyAr
aNummowie1du4a	Cas0d6wconk2
pJkcofoc14rumRqaR	gr4N6N84GgoC
l2b2oSDIm2vR	aL3O19126m
jAvdWkIvyh9Wkbz97SN	E9daAOWovk1q7hg2eI
EawcEgRe17r	ipkmpeS7kieejmcS45
5prJuE9Ceicd2A1X1N	rmdfd0rdedpcfRCx
count gene, count length 8 to 20, percentage:	39, 18, 46.153846153846156%
zxcvbn average, length of passwords:	3.888888888888889, 18

epochs = 20000, RockYou zxcvbn

PRYOWO12dG8Ge	wereC2RSSYD73y4IGI7
RXyLPyEGI	ACwy1@h1J
rCAD9ID23Xn@	19A46sEEDYz9
147uniswexrbJr1	NI9L6ymud25
NRSLpNglcy1r	s07hby16aOz5RX
OXyy1sDOO@	S15XpyenOR585
698nxeSL1	3tIovNzXvoA3
kvieCOnSNY12CIA	Roh1ste1j69lvPxAX
Jne56bRaXh8SX	z7@RWosh1J5LR4
y3e9Cp1X6E	3ngemwLa1
yme2J2andB	gIScdILO
9k1rDJh25oncehiX6	

count gene, count length 8 to 20, percentage:	58, 32, 55.17241379310345%
zxcvbn, length of passwords:	3.6526, 32

CHAPTER 5
LIMITATIONS

This project cannot count all the zxcvbn scores of any password because the limitation of CPU speed. Therefore, this project can only calculate the zxcvbn scores of samples, which are chosen from two data sets. Meanwhile, when this project ran some RNN codes, this project also chose samples to get the results quickly.

5.1 Data Set Size Imbalance

The number of passwords in each of the two data sets are disproportionately different from each other. 1.4 Billion Passwords is about 45 GB and Rockyou 2009 is about 140MB.

5.2 Password Rule and Strength Meters Have Improved Over Time

The password rule has been improved from year to year to improve the strength of passwords. The passwords used 10 years ago in many websites did not need a combination of numbers, upper cases and special symbols. However, now, many websites demands that the passwords of users should be a combination of them. Meanwhile, there are also other rules.

CHAPTER 6
CONCLUSIONS

These conclusions describe the differences of these two data sets by the categories of numbers, upper cases and special symbols, compare vulnerabilities by zxcvbn scores, find similarities by t-Test and get a final data analysis. After that, these conclusions also include the functions of predicting next character and generating passwords by machine learning algorithms.

6.1 Description of Categories: Numbers, Upper Cases and Special Symbols

At beginning, the users in RockYou 2009 used more numbers and special symbols than the users in 1.4 Billion 2017. At end, the users in RockYou 2009 used more numbers, special symbols and special symbols than the users in 1.4 Billion 2017.

From second character to the last second character, the users in RockYou 2009 used more numbers and special symbols than the users in 1.4 Billion 2017, but special symbols are less. The average length of RockYou passwords in 2009 is longer than the average length of 1.4 Billion 2017.

The final conclusion of the whole passwords are the same as situations from second character to the last second character .

6.2 zxcvbn Score

1.4 Billion Password data set 2017 has a lower score in every item compared with the passwords in RockYou data set 2009. So in these groups of samples, the 1.4 Billion Password data set is more vulnerable.

Though the 300,000 samples, this project found that their average zxcvbn socres are very close though RockYou 2009 is still sligtly bigger than 1.4 Billion 2017.

The situations of 3 million samples are very similar to 300,000 samples.

The users until the end of 2017 use much stronger password than the users in 2009 as long as it contains at least one number or one upper-case character or one special symbol. However, the average scores of whole passwords of RockYou data set is slightly stronger than 1.4 Billion 2017 password data set samples. Meanwhile, when referenced with all the results generated above, a conclusion is that the users until the end of 2017 use shorter length of passwords, less numbers and upper-cases and a little more special symbols, but the strength of the passwords until the end of 2017 had not been reduced a lot. In the contrary, as long as the users until the end of 2017 choose to include any number, upper case or special symbol in passwords, the strength is much stronger compared with the passwords in RockYou 2009.

6.3 t-Test

This p-values means the ratios are similar to each other [27].

In RockYou 2009 and 1.4 Billion Passwords 2017 From password[1] to password[length-2], the p-values has greatly larger than 0.05. This means 1.4 Billion Passwords from password[1] to password[length-2] has a great similarity among them.

In 1.4 Billion From password[1] to password[length-2], here the P-value is much bigger than 0.05, which indicates this project rejects the null hypothesis. There is a big difference among the ratios of number, upper-case characters and special symbols in different data sets.

In RockYou 2009 and 1.4 Billion password data sets from password[0] to password[length-1], the pvalue is still very high between RockYou 2009 and 1.4 Billion Password data sets, which means the whole passwords are very similar.

6.4 Data Analysis

According to different kinds of comparisons of lengths and pvalues, we can know that RockYou 2009 is very similar to 1.4 Billion 2017 about using amount of numbers, upper cases or special symbols. The overall strength of RockYou 2009 passwords is also close to 1.4 Billion 2017 samples based on zxcvbn scores. However, as long as the users until the end of 2017 include any number, upper case or special symbol in their passwords, they can provide much stronger passwords.

6.5 Machine Learning Utilization

When KNeighbors classifier is applied, the result is almost 10 percentage and if there is more training data set. This next character prediction will be more accurate.

This project used RNN(Recurrent Neural Network) to train the whole RockYou data set with 10 epochs. After 50 to 60 hours, the whole The generated passwords seemed wired because there were many passwords including many character P. Therefore, the methods could be improved a little.

This project also used RNN to train the number-containing password array from RockYou with epochs 5000 or epochs 20000. These passwords have stronger strength and after trained by RNN either in epochs 5000 and epochs 20000. They can be referred as the substitutes of the weak passwords in RockYou 2009 data set. These stronger passwords can be especially recommended to RockYou data set users, when they wants to find a familiar and stronger password. These generated passwords are from the trained model, which is familiar with the habits of users. Therefore, they are the good substitutes for weak passwords.

REFERENCES

[1] B. Pal, T. Daniel, R. Chatterjee, and T. Ristenpart, "Beyond credential stuffing: Password similarity models using neural networks," in *2019 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2019, pp. 417–434.

[2] J. Casal, "1.4 billion clear text credentials discovered in a single database," https://medium.com/4iqdelvedeep/1-4-billion-clear-text-credentials-discovered-in-a-single-database-3131d0a1ae14, Dec, 2017.

[3] C. Nik, "Rockyou hack:from bad to worse," https://techcrunch.com/2009/12/14/rockyou-hack-security-myspace-facebook-passwords/, Dec, 2009.

[4] W. Matt, "Reusable security," https://reusablesec.blogspot.com/2009/12/rockyou-32-million-password-list-top.html, Dec, 2009.

[5] 4iQ, "Insights into the 1.4 billion clear text credentials trove," https://4iq.com/wp-content/uploads/2018/03/1.4-Billion-Clear-Text-Credentials-Trove-Report_2018.pdf, March, 2018.

[6] W. Melicher, B. Ur, S. M. Segreti, S. Komanduri, L. Bauer, N. Christin, and L. F. Cranor, "Fast, lean, and accurate: Modeling password guessability using neural networks," in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016, pp. 175–191.

[7] Z. Xia, P. Yi, Y. Liu, B. Jiang, W. Wang, and T. Zhu, "Genpass: A multi-source deep learning model for password guessing," *IEEE Transactions on Multimedia*, vol. 22, no. 5, pp. 1323–1332, 2019.

[8] B. Hitaj, P. Gasti, G. Ateniese, and F. Perez-Cruz, "Passgan: A deep learning approach for password guessing," in *International Conference on Applied Cryptography and Network Security*, Springer, 2019, pp. 217–237.

[9] A. Ciaramella, P. D'Arco, A. De Santis, C. Galdi, and R. Tagliaferri, "Neural network techniques for proactive password checking," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 4, pp. 327–339, 2006.

[10] D. Pasquini, G. Ateniese, and M. Bernsaschi, "Interpretable probabilistic password strength meters via deep learning," *arXiv preprint arXiv:2004.07179*, 2020.

[11] D. He, B. Zhou, X. Yang, S. Chan, Y. Cheng, and N. Guiana, "Group password strength meter based on attention mechanism," *IEEE Network*, 2020.

[12] R. Rathi, P. Visvanathan, R. Kanchana, and R. Anand, “A comparative analysis of soft computing techniques for password strength classification,” in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, IEEE, 2020, pp. 1–3.

[13] M. Vijaya, K. Jamuna, and S. Karpagavalli, “Password strength prediction using supervised machine learning techniques,” in *2009 international conference on advances in computing, control, and telecommunication technologies*, IEEE, 2009, pp. 401–405.

[14] 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)*, IEEE, 2016, pp. 595–606.

[15] J. Galbally, I. Coisel, and I. Sanchez, “A probabilistic framework for improved password strength metrics,” in *2014 International Carnahan Conference on Security Technology (ICCST)*, IEEE, 2014, pp. 1–6.

[16] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. Lopez, “Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms,” in *2012 IEEE symposium on security and privacy*, IEEE, 2012, pp. 523–537.

[17] M. M. Taha, T. A. Alhaj, A. E. Moktar, A. H. Salim, and S. M. Abdullah, “On password strength measurements: Password entropy and password quality,” in *2013 INTERNATIONAL CONFERENCE ON COMPUTING, ELECTRICAL AND ELECTRONIC ENGINEERING (ICCEEE)*, IEEE, 2013, pp. 497–501.

[18] Y. Guo and Z. Zhang, “Lpse: Lightweight password-strength estimation for password meters,” *computers & security*, vol. 73, pp. 507–518, 2018.

[19] J. Galbally, I. Coisel, and I. Sanchez, “A new multimodal approach for password strength estimation—part i: Theory and algorithms,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 2829–2844, 2016.

[20] I. Olade, H. Liang, and C. Fleming, “Semanticlock: An authentication method for mobile devices using semantically-linked images,” *arXiv preprint arXiv:1806.11361*, 2018.

[21] R. Veras, J. Thorpe, and C. Collins, “Visualizing semantics in passwords: The role of dates,” in *Proceedings of the Ninth International Symposium on Visualization for Cyber Security*, 2012, pp. 88–95.

[22] Wikipedia, “Rockyou,” https://en.wikipedia.org/wiki/RockYou.

[23] "Zxcvbn," https://github.com/dropbox/zxcvbn.

[24] Scipy.org, "Scipy.stats.ttest.ind," https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html.

[25] M. B. Editor, "How to correctly interpret p values," https://blog.minitab.com/blog/adventures-in-statistics-2/how-to-correctly-interpret-p-values, April 17, 2014.

[26] Y. Dong, "The degenration of passwords," https://github.com/MichaelLoveMr7/Password-0-Password-len-1-, October, 2020.

[27] ——, "The degenration of passwords," https://github.com/MichaelLoveMr7/The-de-of-pass-1tolast2char, October, 2020.

[28] ——, "The degenration of passwords," https://github.com/MichaelLoveMr7/Data-analysis-0-To-length-1-ALL, October, 2020.