REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

EXPLORING SOCIAL INFLUENCE IN THE DESIGN AND EVALUATION OF ENHANCED PASSWORD METERS WITH PEER-DRIVEN SECURITY

Hawraa Rida Abdulrazzaq AL-QAZZAZ

Master's Thesis

Supervisor

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

İstanbul, 2025

EXPLORING SOCIAL INFLUENCE IN THE DESIGN AND EVALUATION OF ENHANCED PASSWORD METERS WITH PEER-DRIVEN SECURITY

Hawraa Rida Abdulrazzaq AL-QAZZAZ

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2025

The thesis titled EXPLORING SOCIAL INFLUENCE IN THE DESIGN AND EVALUATION OF ENHANCED PASSWORD METERS WITH PEER-DRIVEN SECURITY prepared by HAWRAA RIDA ABDULRAZZAQ AL-QAZZAZ and submitted on 13/05/2025 has been **accepted unanimously** for the degree of Master of Science in Electrical and Computer Engineering.

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Thesis Defense Committee Members:

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	Department of Computer Engineering, Altınbaş University	_____
Assoc. Prof. Dr. Sefer KURNAZ	Department of Computer Engineering, Altınbaş University	_____
Asst. Prof. Dr. Tareq Abed MOHAMMED	Department of Computer Engineering, Altınbaş University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Hawraa Rida Abdulrazzaq AL-QAZZAZ

Signature

DEDICATION

To my beloved mother and father,

To my my husband and daughters and sister,

And all my friends,

You have always respected what I wanted to do and have given me full support and encouragement throughout my life. I would like to express my great appreciation for your great love.

ABSTRACT

EXPLORING SOCIAL INFLUENCE IN THE DESIGN AND EVALUATION OF ENHANCED PASSWORD METERS WITH PEER-DRIVEN SECURITY

AL-QAZZAZ, Hawraa Rida Abdulrazzaq

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Date:05/2025

Pages: 47

Passwords have controlled the world of authentication. As a result of their extensive use, they have become a desired target for attackers. To thwart such assaults, many strategies have been used to increase password security. The websites and applications use password meters to assist users in creating complex passwords. The purpose of a password meter is to give the users knowledge about their password selection by identifying it as “weak,” “medium,” or “strong,” for example. This study has taken into account social effect, or how the other influences a person's behavior and attitude. This social influence, often known as peer feedback, was taken into account while designing a peer feedback password meter. Participants who used the peer feedback meter and were instructed to create a special password had stronger passwords than those who used the conventional meter. In this research, many works related to methods of measuring password strength were reviewed, and algorithms used to measure password strength were also reviewed. Some of the ways in which the attacker hacked passwords were also mentioned. We aim to provide effective methods for evaluating passwords with five levels of integrity.

Keywords: Password Strength Tester, Man-in-the-middle (MITM) Attacks, Brute Force Attack, Dictionary Attack, Entropy.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

ABBREVIATIONS

PST : Password Strength Tester

MITM : Man-in-the-middle attacks

BFA : Brute Force Attack

DA : Dictionary Attack

OFDM : Orthogonal Frequency Division Multiplexing

LIST OF SYMBOLS

H : Entropy

P(X) : Probability of a given character x appearing in the password

1. INTRODUCTION

1.1 DEFINITION

Due to some projections, Internet users are predicted to exceed 6 billion by 2025, creating over 60 “zettabytes” of data by 2026. All this information is essential for the person, the institution, and any malevolent actor. “Yahoo!”, “Dropbox” and “LinkedIn” recently suffered data breaches that exposed up to 732 million user details. Email addresses, passwords, personal questions, and answers are examples of personally identifiable information (PII) that has been exposed. It has jeopardized the confidentiality and integrity of data, as well as the user's security and privacy.

Organizations that are subjected to these attacks risk losing money as well as their reputation. According to some estimates, victims of cybercrime spent $126 billion in 2015 dealing with the aftermath of an attack. Hacking was responsible for around 62 percent of data breaches in 2016with, with 81 percent of those relying on information that has been stolen previously or weak passwords.

Passwords have long dominated the authentication industry and continue to be a key part of people's defense against numerous dangers. As a result of their extensive use, they have become a desired target for attackers. If an attacker obtains a password, they have effectively compromised the entire system's security. According to one research, "123456" was the most used password, consequently, it's not surprising that weak passwords were the cause of 28 percent of the 500 intrusions in 15 nations. The hacked bank accounts (such as PayPal) were sold on the black market for over $60 per account.

A password strength tester is a software application that evaluates the strength of passwords based on various criteria, including length, complexity, and randomness. The tester provides feedback on the strength of the password and suggests improvements to make the password stronger. Password strength testers are commonly used by individuals and organizations to ensure that their passwords are secure and protect against unauthorized access to their accounts.

1.1.1 Types of Password Strength Testers

1.1.1.1 Offline password strength testers

Unlike other applications that run online, offline password strength testers are software programs that are stored on the user's local computer. The user can generate and test passwords without the risk of sending them over the internet or having the password intercepted by unauthorized individuals. Furthermore, there is an advantage to being offline. In areas where there is very little or no internet connection, offline password strength testers can be relied upon since they do not require any internet access.

Offline Password Strength Testers come in many different types including, stand alone applications, browser extensions and plug-ins. Some of these programs include KeePass, LastPass and Dashlane.

1.1.1.2 Online password strength testers

Online password strength testers are hosted on the web, and therefore require an internet connection to operate. The users can generate and get hold of passwords from any online device. One good aspect of these test programs is that they utilize the most current security features enabling users to generate non compromiseable passwords.

You can find online password strength testers on password management and security websites, as well as on password generator tools. Some of the websites that offer such online password strength testers include Norton Password Generator, Kaspersky Password Checker, and How Secure Is My Password.

1.1.2 Techniques Used by Password Strength Testers

1.1.2.1 Length-based analysis

Length based analysis is one of the methods which is used for evaluating how strong a password is. This method helps password testers in evaluating a particular password based on its length. The greater the length of the password the more difficult it is for attackers to crack it due to the greater number of combination guesses that would be required. A password strength tester’s determination of the strength of a password based on length relies

on a scale. For example, there would be no doubt that a password of 8 characters in length is weak, but a password of 20 characters will relatively be much more secure.

1.1.2.2 Dictionary-based analysis

This section will outline dictionary-based techniques for assessing password strength, specifically searching for the presence of popular phrases or words. Reference Passwords that are found in dictionaries tend to be weaker than passwords that are not in the dictionaries. For instance, password strength testers maintain a dictionary of commonly used words and phrases. Any expression or word he coined which adheres to the criteria is rendered unreliable.

1.1.2.3 Brute-force analysis

Brute-force analysis is a method used to test a password's strength by systematically trying every combination of characters until the correct one is found. It requires a large amount of time and computer resources, and has little practical application in every-day situations. Still, password strength testers apply this method to evaluate the complexity of passwords by attempting to ascertain them through exhaustive combinatorial searches.

1.1.2.4 Markov-based analysis

The Markov model analysis methods are often employed in password strength testing by estimating the sequential character occurrence probabilities. When testing against dictionary-based algorithms, Markov model analysis is more potent because it can discern patterns within passwords. Markov models assist in testing the strength of passwords and their security levels by computing the chances of a password’s characters being contiguous.

1.1.3 Effectiveness of Password Strength Testers

Overall, password strength testers are useful in determining the strength of a given password and offering suggestions for improvement. Regardless, they can be defeated through social engineering hacks or more aggressive brute-force methods. Moreover, password strength testers can only measure the strength of a password defined by evaluation parameters. A password perceived to be strong from one evaluating perspective can be weak from another defined perspective.

1.2 LITERATURE REVIEW

We'll discuss password strength meters in this part, along with methods for ensuring their accuracy.

Information security is the process of safeguarding data and preventing unauthorized access. Information security is working to make at least one of the three objectives—safeguarding the information system valid. There are three categories [1]:

a. Integrity: Security measures protect data from being altered without authorization. These processes ensure that the information is complete and accurate. Data must be safeguarded when stored on computers as well as when it is transmitted between systems, including via email. It is imperative to restrict access at the system level, but so is ensuring that users can only make changes to the data they are authorized to alter in order to ensure integrity.

b. Confidentiality: is roughly equivalent to privacy, which means the Information should be unavailable to the people who unauthorized.

c. Availability: Only the appropriate individuals should have access to accurate information and its accessibility. Systems that store and display the data are involved in this.

There are two distinct methods to determine the strength of a password. The first one is based on the password's complexity, which includes capital and lowercase letters, symbols, spaces, and other characters.

The second one is the ability to guess passwords, which is the number of guesses it takes for the attacker to guess the password.

The password strength needed to divide into five clusters which are:

1= very weak.
2= weak.

3=medium.

4=strong.

5=very strong.

And this five-cluster selected by how much time it needs to crack the password.

1.2.1 Threats against information systems

Even in their private lives, people grow to rely so heavily on the internet. The internet gave us the ability to expand our information, but on the other hand, it gave the unauthorized people the power to access other people's information to achieve their own goals [2].

1.2.1.1 Phishing

Phishing is accursed when a hacker collects the information by sending an email as if it is from a trusted source. As the email looks real as if it was sent from the user's bank as the hacker wants the person to put his personal information such as passwords, there are three ways of phishing which are the most common:

Initially, and most frequently, there is an impression. The impression is when they create a fake website, and this fake website includes pictures like the actual website to attract the users.

The second phishing method is forwarding. The crook sends an email that makes you fill in your account information. In this case, most of the users don't release that this is a fake email until too late. There is one weak point in this method which is the HTML, as in some time it may block the email. The third method is a popup attack. A popup attack shows when the user clicks on the link in the website in an email. After that, they send you to another page which is a popup window.

In this popup window, there is a specific aim that is collecting the information.

1.2.1.2 Man-in-the-middle (MITM) attacks

It is when an attacker deciphers the information and data that they share, like passwords.

Where the attacker puts himself in the place of one of the parties (i.e., in a conversation between two parties) and hopes to impersonate or eavesdrop, making it like it was a natural exchange of information between two people.

This attack's primary goal is to steal personal data, including credit card numbers and login credentials. Users of banking apps, e-commerce websites, and websites requiring personal information are typically the target audience.

Attackers who wish to take a more active approach may use one of the following attacks:

a. IP spoofing: The attacker assumes the identity of an application and modifies the IP address's packet headers.

b. ARP spoofing is the practice of using fictitious ARP messages to link the IP address of a legitimate user to the MAC address of an attacker.

c. DNS spoofing is known as “Domain Name System Cache Corruption”, as it infiltrates the DNS server and changes the website address record.

In 2017 “Equifax Company” removed its apps from google play and the app store because it was transferring important information in an unsafe channel so that the hacker could get the customer information.

1.2.1.3 Brute force attack

Brute force attacks use a battering ram, and this hacking technique uses trial and error to try and crack the password. The attacker can obtain illegal access to company systems and non-individual accounts using this dependable and easy way.

On the other hand, the attacker frequently tests a variety of groups using a computer and multiple users' names and passwords. Using this technique, a hacker can attempt more than 2.18 trillion usernames and passwords in 22 seconds.

Types of Brute Force Attacks

a. Basic Brute Force Attacks: In a basic brute force attack, an attacker guesses a user's login credentials without the use of a software.
b. Derogatory References.
c. Hybrid Brute Force Attack: This kind of attack combines a simple brute force attack with a dictionary attack. In order to utilize a dictionary attack and simple brute force methods to gain access to the account in reverse, the hacker must first know the username.

d. Credential stuffing: After obtaining usernames and passwords, the attackers gather them and try them on different websites to see whether they grant them access to other accounts.

1.2.1.4 Keyloggers

Keyloggers are a software application that allows the hacker to know about every key click on your keyboard the user does.

The name is a shortened version of the keystroke logger. One of the main ways the app uses is to record what is typed on the keyboard. The software is installed on your computer and records everything you type.

1.2.1.5 Dictionary attack

One kind of brute force attack is a dictionary attack, which depends on our habit of creating a primary phrase as a password, like a birthplace, birth date, etc.

This is a simple hacking technique in which the attacker creates a target and then verifies the passwords. Although the assault technique is not regarded as a brute force attack per se, it can be quite useful in password cracking processes. Attackers substitute special characters for some words and letters.

Compared to other attack techniques, this kind of attack is more time-consuming and has lower success rates.

1.2.2 Password Strength Meters

1.2.2.1 Approximating strength

According to studies in [3], The names of a spouse or child accounted for almost 15% of the passwords, with football teams and pet names coming in second and third. According to one poll, 26.2 percent of respondents have more than 50 accounts on average. Furthermore, 50.9 percent of those polled said they frequently repeat passwords because they are simpler to remember. If a report was hacked, it might set off a chain reaction that allows other charges affiliated with that person to be hacked. This includes any statements one may have with their employment, raising the risk of a non-malicious influence.

Password strength meters are a helpful tool for assisting users in creating safe passwords. Meaningful information can only be obtained from strength meters provided their score truly reflects the strength of the password. A weak strength meter that is inaccurate could cause more harm than good by incentivizing users to use passwords with low security but high scores. Which strength meters are most beneficial in advising consumers since they offer high accuracy is unclear, despite the fact that numerous alternative strength meters have been put out in writing and applied in real life. Moreover, our understanding of comparing strength meter accuracy is lacking [3].

Passwords like “passw0rd” or “abc123” aren't necessarily unsafe (e.g., because they meet some "magical" feature). They are risky because they are frequently used, and if someone is trying to guess passwords, they will first guess those that are frequently used. Consequently, each password is given a probability by an optimal strength meter, which may be calculated using the relative rate of a large enough password corpus. While relative frequencies can be roughly enough for "reasonably likely" passwords—passwords that are particularly significant for online guessing attempts, for example—in actual use, this simple concept becomes challenging to put into reality. Since less common passwords are crucial for offline guessing attacks, estimating their frequencies is nearly hard because of the volume of information needed. Therefore, the real strength in practical strength meters should be approximated using compactly representable functions.

Other techniques based on Markov models or PCFGs [4, 5] are more accurate at the penalty of greater storage capacity. On the other hand, limited accuracy is associated with the traditional LUDS meter, which permits a very compact representation (of a few bytes) [6].

1.2.2.2 Measuring accuracy

One of the most critical aspects of PSMs is accuracy, and various PSMs have been proposed in recent years. However, little effort has gone into making a fair comparison of different meters, and there is no consensus on what constitutes a fair comparison. Comparing a strength meter to an ideal reference and assessing the preferred way to determine its accuracy is the degree to which the meter output and the reference are similar.

This concept is based on the assumption that frequent and previously used passwords are weak. On the other hand, previous work used ad hoc methodologies to compare reference

and tested meters, ranging from measures calculating overestimation errors to ranking the linkage metrics.

Designing a password-strength assessment method acceptable for password meters is a complex problem. Due to the complexity of passwords, even with the most modern password-guessing algorithms, there are still some differences in strength measures of the same password [7].

A practical, lightweight password meter, in our opinion, should be able to reliably evaluate the strength of a suggested password on the user's client machine.

There are latency and security problems with the password meters used on the server side [8, 9]. A password strength meter that uses minimum storage and operates in real-time is required if the password strength assessment is done on the client-side. To build a high-accuracy password checker, Carnavalet et al. at [10] recommend that numerous elements must be considered as intrinsic patterns in user choice, dictionaries utilized in cracking tools, exposure of big password datasets, and user adaption against password regulations.

It appears that creating such a password checker would be a challenging undertaking. It's impossible to determine which dictionaries the attackers employ or how many password databases have been exposed. To make these problems easier to solve, Carnavalet et al. at [11] propose that the primary purpose of password meters be to identify weak passwords using well-known password-cracking techniques and dictionaries. The most accurate approach to measure password strength is to use current password-cracking tools, such as probability-based guessing algorithms [11, 12].

Guess-ability, or the number of guesses an algorithm has to make to guess a password given a set of parameters and training sets, is used by them as a measure of password strength. Probability-based guessing techniques are not appropriate for determining the strength of a password on the client side, are computationally expensive, and require a large amount of storage space [8, 13]. Using a conventional password dictionary on the client-side is also not a smart idea, as it consumes storage space and slows down the password-evaluation algorithms' searching speed. Moreover, the most common phrases in passwords written in various languages are disregarded in a conventional password dictionary, which only includes terms in one language.

The NIST recommended that the password include at least one lowercase and uppercase letter and one special symbol with an eight-character minimum.

In [14], researchers examined the accuracy of password strength counters. They have demonstrated the inaccuracy of the current techniques used to assess the strength meters' accuracy. After comparing several similarity measures, they came to the conclusion that the most reliable and dependable way to estimate the accuracy of strength meters is to use weighted Spearmint correlation. Using this technique, they evaluated the accuracy of forty-five different strength meters both offline and online.

Researchers found that academic PSM proposals that use "Markov models," "PCFGs," and "RNNs" perform better than PSM proposals from businesses. Additionally, they discovered that a few password managers and websites had incredibly precise strength meters. Strength meters, however, are not as accurate as recommended by academics, and their study did not significantly improve meter accuracy over meters from five years prior. The accuracy of a password strength meter is one factor that influences its security.

Researchers in [15] conducted the first comprehensive analysis of password-strength meters and discovered that the meters did have an impact on security and user behaviour. People were urged to choose longer passwords by meters. Nevertheless, unless the meter carefully graded them, There was very little difference in the generated passwords' resistance to password-cracking attempts.

Users produced longer passwords with more capital letters, numbers, and symbols as a result of the stringent password rating meters. The pace at which simulated attackers could crack passwords using 500 million, 50 billion, and 5 trillion guesses was lower, despite the fact that they were not determined to be less memorable or useful.

Customers were upset by the most restrictive meter, which offered no further security advantages over less demanding meters. Text and a visual cue together performed better than either one by itself. On the other hand, the appearance of the visual indicator did not seem to have a major impact. They discovered that, in reality, many more lenient meters were employed, despite the greater strength these tougher meters broadcast. According to their research, if stricter meters are not overly onerous, security will increase.

The question of whether peer pressure affected a user's choice of password was investigated in [16]. The researchers tested 48 college students in order to provide an answer to this query. A normal password and a peer feedback password were the two kinds of password meters they utilized. Additionally, they experimented with two other requirements: clear orders and no costs that are explicitly stated.

The findings imply that when used with detailed instructions, the peer feedback meter produces more secure passwords than typical password meters. As a result, their hypothesis is confirmed. However, they did not uncover a statistically significant difference when no specific instructions were given. They concluded that prototype peer feedback meters and a method to remind the user to generate a unique password would be most beneficial for sites that currently rely on social relationships between users. “Facebook,” “LinkedIn,” “Google,” and “Outlook” are just a few examples. While some of these platforms' users' activity poses security and privacy concerns [17], preventing unauthorized users from accessing accounts remains a primary issue.

2. RELATED WORK

2.1 PASSWORD CHOICE

In general users create their passwords by using a small group of rules, such as the characters and symbols. The users have been found making their passwords depending on concepts relating to feelings, names, foods, etc. The relationship between how people generate their passwords and whether or not they have been verified as strong. They also report wrong conceptions on the consequence of creating passwords by using known phrases and digits [17].

A process called Password Composition Policies (PCP) is used to stop users from making passwords that are simple to figure out. Based on these policies “komanduri” has set up some recommendations for password Composition policies that enable users to create strong passwords, as he focused on reducing the number of guesses on the internet and blocking the most common passwords.

The password-cracking tool "John The Ripper" (JTR) [18] is available. It allows for the creation of password guesses in several modes. Dictionary mode generates guesses by obfuscating information using many dictionaries. Mangling rules can produce word variations from the input dictionary, enabling more effective guessing. They are used to evaluate user behavior when creating passwords. The mangling rule, for instance, might add a number to a word or change the "letter a" to @ or the "s letter" to $, etc.

2.2 PASSWORD STRENGTH

Trying to assess password strength to get rid of guessing attacks has a long history in 1949 "Morris and Thompson" [19] checked the power of a password by trying to crack a password, and the password which successfully broke was deemed to be "weak," and users had observed that, so they started to check their password strength before the system accepts it via password strength meters by some rules that exclude weak passwords [20,21].

In [22], researchers built an algorithm that measures the strength of the password, and one of these algorithms is the “zxcvbn” algorithm, which considered as one of the most important and best algorithms that determine the password's strength. This algorithm divides the

password into several patterns and then studies them separately. The approach that determines the power of the password is an approach that tests the most common passwords first, and the password guessed by this approach later is the strongest and most secure.

Lightweight Password-Strength Estimation (LPSE) is an algorithm to calculate the password's strength. It requires a small space in terms of storage, and its speed is sufficient and acceptable to measure the power of the client's password [23].

The "similarity assessment method" has been proposed as an effective password strength assessment method that the client can enable. It assesses password strength by evaluating the similarities between a password that the user provides and a strong password in multiple ways. The password strength is assessed using the similarity approach, which takes into account the password's "structure," "vector coefficient," and "distance." It operates by contrasting the password with a typical strong password. Three factors are compared between the two passwords in the article: the length of the password, its structure, and the quantity of insertions and replacements. The effect of password strength metrics on password selection have been studied, and found that when users have to change passwords in their important accounts, password metrics result in much more robust passwords [24].

2.3 PASSWORD GUESSING

Password guessing is a technique similar to password estimation that determines a password's strength from the standpoint of an attacker. The strongest password is the one that gets guessed later, because password guessing techniques multiply guesses in declining order of probability, testing the most popular passwords first. As a result, the quantity of guesses is frequently used by researchers to gauge password strength [25].

Password guessing is related to password strength. The best strategy for testing passwords is to rank the probability in descending order, i.e., putting the most common password first [26]. The researchers in [27] have built a new architecture called “camouflage,” as it builds programs that manage anti-theft passwords. This system forced one of the attackers to make a great effort and perform a large amount of work until he could steal the information of an electronic device.

Also, in [28] a new method was developed that provides the user to create a new password from a computer system which included the Universal Password Generator (UPG).

The UPG is a tool that gives password-based instructions specified by the user to create a strong password for each account. In [29], a mechanism was developed that depends on the linguistic and grammatical rules of passwords. This mechanism has been tried and found that the effectiveness in penetrating and cracking passwords does not exceed 25%.

Different definitions have been found to define "entropy". The definition of “entropy “is used as a measure of information. In [30], the researchers introduce the entropy to filter the weak passwords. The researchers used seven characters alphabet passwords. In [31], the researchers calculate the entropy by making it into parts. Password length, characters places, and the number of each character then estimate the entropy by summing the values.

In [32], while developing a password quality indicator (PQL) which is a framework to measure the password's strength, so the strength password depends on how much is different from the examples in the dictionary and how long and significant the password is.

Checking the password is to improve password security by preventing the users from creating an easy guessing password Dictionary-Based Proactive Password checker (DBPP) is a program that includes a list of possible passwords, then it searches for the password that the user creates if it is in the list or not. If the password is similar to a password in the dictionary, then the tester will block the password and will suggest a better password [33].

Microsoft password checker (MPC) is an online checker developed to show how good the password is; Microsoft depends on the type of the characters and if the password is found in the dictionary or not [34].

2.4 MACHINE LEARNING TO CLASSIFY THE STRENGTH

Organizations must establish stringent standards around the creation and use of passwords and be cognizant of the risks associated with password usage in order to prevent these risks from being exploited. Using supervised machine learning techniques, [36] approaches the prediction of a password's security level as a classification issue. The C 4.5 decision tree classifier, the multilayer perceptron, the naive Bayes classifier, and the support vector machine were a few of the most popular supervised machine learning algorithms that were

used to learn the model. The SVM performed exceptionally well when the models' results were compared. The outcomes of the models were also compared to those that might be acquired with the different password strength testing tools that are already available. The results of the investigation demonstrate that the machine learning method is very capable of categorizing the worst-case scenarios.

In [37], researchers present a system that can proactively assess how strong the character set of a password is. Filters and a support vector machine are utilized in order to do analysis on the selected password. The aforementioned framework is capable of being incorporated into the access control system in the capacity of a submodule.

The paper [38] applies a technology called machine learning to assess the strength of the password in order to assist enterprises in launching a complex defense against password breach and to offer an environment that is extremely secure. The categorization of passwords is accomplished by the application of a supervised learning method known as Support Vector Machine. In order to train the linear and nonlinear SVM classification models, the features retrieved from the password dataset are used. When subjected to tenfold cross-validation, the trained model demonstrates a prediction accuracy of around 98%.

Scholars in [39] describe how machine learning methods can be applied to produce forecasts that enable the anticipation of a hacking incident by accounting for the entropy of passwords and the actions of security experts. Techniques such as decision trees, multilayer perceptrons, and Naive Bayes are utilized in the process of developing and training prediction models. The effectiveness of these models is compared to one another in order to establish which of the models is more effective for the situation that is being investigated. It is becoming increasingly frequent for individuals to make attempts to steal information by breaking into internet accounts or violating password security. The majority of cyberattacks involve some kind of human interaction. Beginning to use artificial intelligence and placing certain human choices in the hands of computers is one approach to resolving these human oversights; nevertheless, these advances still have a long way to go before reaching their full potential. In order to close the gap between what technology can do and what humans need, researchers still require the application of human judgment. Here is where deliberate security practices come into play as a distinguishing factor between being the victim of a cyberattack and not being the victim of one. In light of password entropy and security expert behavior,

this study describes how machine learning techniques can be applied to produce predictions that enable the anticipation of a hacking event. Techniques such as decision trees, multilayer perceptron, and Naive Bayes are utilized in the process of developing and training prediction models.

In [40], researchers propose a comparative analysis of soft computing techniques for strong password classification. These techniques include Back propagation neural network (BPN), Logistic regression (LR), Hopfield neural network (HNN), Brain state in a box (BSB), and associative memory network (BAM), and Convolution neural network (CNN). The dataset is comprised of just one feature, which is passwords, and it has three categories for them: weak, medium, and strong. The analysis of the experimental results shows that for a dataset such as this, the straightforward logistic regression model produces superior results when compared to other models such as Convolution Neural Networks, Back Propagation Neural Networks, Logistic Regression, Hopfield Neural Networks, Brain State in a Box, and Bidirectional Associative Memory. It is possible to draw the conclusion that logistic regression is suitable for such an application and also functions as an improved method for making predictions for such a dataset. When compared with other approaches, LR has a higher accuracy than the other methods, which is 81%.

In [41], authors employ a recurrent neural network using back propagation technique to predict the characters of a password. A text password is a common type of authentication for today's computer security systems. Some of the principles that are helpful in modeling the password include the evaluation of the strength of the password and guessing assaults. Based on the methods used to train the network, researchers propose creating the text password using an artificial neural network. Only text data may be processed using this method, which is both quick and accurate.

3. THE PROPOSED APPROACH

3.1 INTRODUCTION

A password strength tester is a tool and/or an algorithm that checks how strong or secure a password is. This is done so that a password that an attacker is trying to crack or access is not easily guessable so that a system, account or network is secure. The approach password strength tester takes assists people to secure their accounts and create passwords that are hard to decipher or guess. This can assist organizations in enforcing password policies and detecting weaknesses within their systems. However, it is important to note that there is no infallible method for guessing a password's chronological strength so two steps authentication needs to put in place to improve safety. Some methods include rule based systems which are defined by terms that use denial and passwords created by words in the dictionary or trained systems algorithm. A rule based approach is evaluated according to pre-defined guidelines which include but are not limited to a password's complexity, length and character types.

The rule-based systems approach is a method that checks the strength of a password in accordance with a set of defined rules. Such an approach operates on the premise that certain requirements must be met in order for a password to be strong - it has to be long, complex, and random, among others. The rules are created or modified to meet specific practices within an organization. They can also be customized to fit the requirements at hand. For example, a rule-based system may specify that passwords have at least eight characters with each containing one capital letter, one lowercase letter, one number, and one special character. The system calculates how a password meets the expectations against stipulated rules and assigns a strength score or level. This is done in accordance to the criteria. The approach is simple, effective, and easy to adopt. These characteristics make it ideal for organizations trying to establish framework for gauging password strength.

3.2 THE PROPOSED METHODOLOGY

In this research, we introduce an improved approach for determining a password's strength using analyzing various evaluation factors. Our methodology is classified under the Rule-based systems of password strength tester. The evaluation factors include:

a. Strength score.

b. Efficiency.

c. Entropy.

d. Security Score.

e. Complexity Score.

We have implemented the methodology by Java programming language using NetBeans 8.2 IDE. Moreover, the conducted experiments include two scenarios of performing the proposed method. In the first scenario, we perform the method on some passwords we already suggested. In the upcoming sections, we will present more details about the implemented modules of the proposed approach. Where each module handles one of the evaluation factors of password strength tester.

3.2.1 Strength Score

Passwords are a vital component of our online security in the modern digital world. They act as our first line of protection against illegal access to private and confidential data. Nevertheless, many people still use weak and simple to figure out passwords because they don't think about how strong they should be. Password strength scores have been created to evaluate a password's resilience in order to overcome this problem. A password strength score is a measure of the complexity and unpredictability of a password. It is computed from the length of the password, how complex it is (in terms of uppercases, lower case letters, numbers and symbols), and how random the password is. The score increases as the randomness increases which means the password becomes harder to guess and more secure.

The proposition we suggest to determine a password strength score is illustrated in Algorithm 1. It uses the range of 1 to 5 to rate a password. 1 means the password is weak and can easily be guessed while 2 means the password is weak. 3 shows that the password given is moderate. 4 shows that the password is strong. 5 means this password is very strong and nearly impossible to break into.

Algorithm 1 describes a program to estimate a password's strength score given its length and character category. To start, the algorithm checks if the password contains at least 8 characters, in which case it grants a length score of 1. Next, the evaluation is done for each character in the password. Characters that are part of a category will score 1 in one of the following variables: digitScore, upperCaseScore, lowerCaseScore, or specialCharScore. Furthermore, there will be boolean variables set for an evaluation to verify if a password has at least one digit, one uppercase letter, one lowercase letter, and one special character. The assessment in the program attempts to derive a password score from the total of all the defined variables' scores. From my understanding, it looks like the author of this program tries to implement a simple mechanism for users to gauge the robustness of their passwords based on a number of characters and how they are structured.

Algorithm 1: Calculate the Strength Score

Input: A string that contains the password.

Output: The value that indicates the strength of the password in question.

Begin:

If the set criterion states the length of the password is more than or equal to 8

Set lengthScore to 1.

Let c be each character within the password. Then,

a. Assign boolean variable hasDigits true, digitScore to 1.

b. Assign boolean variable hasUpperCase true, upperCaseScore to 1.

c. Assign boolean variable hasLowerCase true, lowerCaseScore to 1.

d. Assign boolean variable hasSpecialChar true, specialCharScore to 1.

The cumulative score for lengthScore, digitScore, upperCaseScore, lowerCaseScore, and specialCharScore will yield the total password score.

Provide the password score as output.

End.

Our approach is to enforce that a password includes a mix of upper and lower case characters, numbers, symbols, is minimum eight characters long, and contains a considerable password strength score. To mitigate the risk of multiple accounts being overtaken due to a single hack, we highly recommend utilizing distinct passwords for every online account. Utilizing a password manager allows you to generate and securely store robust, unique passwords for all accounts. Creating strong and complex passwords significantly minimizes the likelihood of unauthorized access to private and sensitive information.

3.2.2 Efficiency

The efficiency of a password strength tester refers to how fast and how easily a password's strength is evaluated. With any users, it is undesirable to wait long to receive results or be made to undergo numerous steps to utilize the tool. Therefore, an ideal password strength tester must be easy to use, intuitive, fast, give immediate results, and have a user friendly design. Along with other tools used to identify weaknesses in passwords, an effective password strength tester is one of the most essential. Individual users are able to create robust procedures that help protect their private personal information using efficient aids and adhering to manifest guidelines.

In this section, we measure the performance of the approach using an algorithm added to the previous one. In the case at hand, verifying the time expenses of executing Algorithm 1, which evaluates the strength of a password, is described in detail in Algorithm 2. First, the system time in nanoseconds is captured. At this point in time, "currentTime" will serve as a moniker for "time". "password" is held captive by the variable and function "strength" is returned. After invoking the function, in Algorithm 1, the system time is captured again in nanoseconds and the new value of time is stored in "endTime". In "startTime" and "endTime" the time needed for the code block – the value of "startTime" is extracted from "endTime". The yield of modulus operation will be duration. 1000.0 duration divided by 1000.0 is converted from nanoseconds to microseconds, duration is defined. This permits the measurement of the time spent executing Algorithm 1 during restrictions imposed by efficiency optimization.

Algorithm 2: Calculate the Efficiency

Input: A string representing the password.

Output: The Efficiency of the given password.

Begin:

1. Get the current system time in nanoseconds and store it in the variable startTime.
2. Call Algorithm 1, passing in the variable password as an argument. Store the result in the variable strength.
3. Get the current system time in nanoseconds and store it in the variable endTime.
4. Calculate the duration of the code block by subtracting startTime from endTime, dividing the result by 1000.0 to convert the time from nanoseconds to microseconds, and storing the result in the variable duration.

End.

3.2.3 Entropy

In the context of passwords, entropy refers to their level of randomness. This means that the greater the entropy a password has, the tougher it is to crack due to the exorbitant computational resources needed to guess it through automated or brute force methods. One of the simplest yet accurate ways to measure password entropy is using the Shannon Entropy Formula.

The formula was made popular in the field of information theory as well as cryptology following its invention by mathematician and cryptologist Claude Shannon. This formula computes the amount of information that can be included or conveyed in a single message and is hence successful. When it comes to password entropy, the Shannon entropy formula is used to calculate how many character combinations can form a password. The formula for Shannon entropy is given as follows:

$$H = -\Sigma\, p(x)\, log2\, p(x) \quad (3.1)$$

In this equation, H denotes the entropy in bits, p(x) is the probability of a given character x appearing in the password, and log2 signifies the base-2 logarithm. Thus the entropy value would be the average number of bits of information per character of the password.

When analyzing password strength using the Shannon entropy theorem, we must establish the character set and the length of the password first. This set regard inclusively to all characters used within the password. For example, if the password solely consists of lowercase letters, then the character set is 26. However, if the password contains a mix of both upper and lower case letters, the character set grows to 52.

Nicholson outlines approaches to assess the password entropy estimation. 1. Identify the character set for a given password. 2. Identify the number of character combinations based on the password length and the character set. 3. Assess the likelihood of each character being used in a password.

Calculate entropy using Shannon's entropy formula.

For example, let's say we have a password with the following characteristics:

a. Length: 10 characters

b. Character set: uppercase and lowercase letters, numbers, and special characters (94 possible characters)

Using the Shannon entropy formula, we can calculate the entropy of this password as follows:

a. The number of possible combinations of characters is 94^10, or approximately 6.634 x 10^19.

b. The likelihood that every character in the password will appear is 1/94, or approximately 0.0106.

c. We can substitute these values into the Shannon entropy formula:

$$H = -\Sigma\, p(x)\, log2\, p(x) = -(10 * 0.0106 * log2\, 0.0106) \approx 47.61\, bits \quad (3.2)$$

Based on this calculation, the entropy of this password is approximately 47.61 bits, which is a relatively strong password.

We have implemented an algorithm to find the entropy of a password based on Shannon formula as illustrated in Algorithm 3. This algorithm is used to calculate the entropy of a password, which is a measure of the password's strength and randomness. The algorithm starts by initializing a variable "entropy" to 0.0. It then gets the length of the password and stores it in a variable "passwordLength". Next, it initializes a variable "charSetSize" to 0. The algorithm then checks the password against regular expressions to determine the extent of the employed character set to create the password. If the password contains only digits, the character set has a fixed size to 10. If it contains only letters (upper or lower case), the character set has a fixed size to 26. Otherwise, if the password is alphanumeric, the character set has a fixed size. to 62.

Finally, the algorithm calculates the entropy of the password by multiplying the password length by the logarithm base-2 of the character set size and stores the result in the variable "entropy". The entropy is a measure of the number of possible combinations that can be generated from the password's character set and length, and a higher entropy indicates a stronger and more random password. The algorithm returns the entropy value, which can be used to evaluate the strength of the password and compare it with other passwords.

Algorithm 3: Calculate the Entropy

Input: A string representing the password.

Output: The entropy of the given password.

Begin:

1. Initialize a variable *entropy* to 0.0.
2. Get the length of the password and store it in a variable *passwordLength*.
3. Initialize a variable *charSetSize* to 0.
4. If the password matches the regular expression "[0-9]+", set *charSetSize* to 10.

---

5. Else if the password matches the regular expression "[a-zA-Z]+", set *charSetSize* to 26.
6. Otherwise, set *charSetSize* to 62, since the password is alphanumeric.
7. Calculate the entropy of the password as *passwordLength* times the logarithm base-2 of *charSetSize*, and store the result in *entropy*.
8. Return *entropy*.

End.

In conclusion, the Shannon entropy formula is a powerful tool for calculating the entropy of passwords. By using this formula, a password's strength can be ascertained by looking at its length and character set. A password with high entropy is considered strong and is more resistant to brute-force attacks. It is recommended to use passwords with high entropy to enhance the security of online accounts and protect personal information.

3.2.4 Security Score

In the digital age, password is the gatekeeper of online security. We need them to keep confidential matters like personal information, online profiles, social accounts, and financial records safe. Weak online security is an all too common phenomenon due to simplistic password choices. In an effort to assist users secure their accounts better, developers alongside security professionals have come up with a scoring system that measures a password’s strength based on its length and complexity. A score reflects the overall strength of the password with a high score meaning stronger passwords.

Factors determining the overall password security score include its length, usage of capital and small letters, numbers, and special characters. The following list breaks down each aspect in detail:

a. Length: A crucial password characteristic for maintaining confidentiality and security. The longer the password the harder it is to guess or crack. The industry benchmarks a password to be strong if it is eight characters or longer.

b. Combination of uppercase and lowercase letters: The use of uppercase and lowercase letters in a password increases its strength. For example, ‘PaSsWoRd’ is stronger than ‘password’ as it contains uppercase letters.

c. Adding numeric characters: Including numbers in a password also makes it stronger. A password such as "P@ssw0rd," which mixes letters with numbers, is more secure than one that uses only letters like "password."

d. Inclusion of special characters: The use of punctuation marks and symbols make a password more complex, thus more difficult to guess. For example, alphabetic passwords ‘P@ssw0rd!’ are much stronger than using alphabets and numbers.

Different characteristics can enhance or detract from a password's strength. A scoring system that evaluates passwords based on these factors has been developed. For instance, ‘password’ has eight characters, thus it can score three if two uppercase letters and numerical digits are added. Special characters increase the score significantly.

We have created a new password scoring system which determines security level through an algorithm in ‘Algorithm 4’. In our approach, security starts from 0 and increases based on certain criteria. The first step is to store the numerical value of characters of the password in ‘length’ variable. The algorithm adds a score based on the password length. If password length is less than 8, score is incremented by 1. From `8` to `11` inclusive, score increases by `2`, while it goes up by `3` when it’s between 12 and 15. Lastly, it can increment up by `4` if its length is equal to or above 16.

Then, the algorithm determines if the password includes lowercase and uppercase letters, digits, and special symbols. If all lowercase letters are part of the password, score can go up by `1`. The same case applies for uppercase letters, however, adding a digit increases score by 1. On the other hand, including special characters can increase score by `2`.

The password strength is determined by the length and content of the password, and the algorithm ultimately yields the score variable. The password is regarded as stronger the higher the score. The algorithm can be helpful in determining a password's strength and in assisting users in coming up with stronger passwords.

Algorithm 4: Calculate the Security Score

Input: A string representing the password.

Output: The Security Score of the given password.

Begin:

1. Initialize the variable **score** to 0.
2. Get the length of the password and store it in the variable **length**.
3. If the length is less than 8

   add 1 to the **score** variable.
4. If the length is between 8 and 11 characters (inclusive)

   add 2 to the **score** variable.
5. If the length is between 12 and 15 characters (inclusive)

   add 3 to the **score** variable.
6. If the length is 16 characters or greater

   add 4 to the **score** variable.
7. If the password contains lowercase letters

   add 1 to the **score** variable.
8. If the password contains uppercase letters

   add 1 to the **score** variable.
9. If the password contains digits

   add 1 to the **score** variable.
10. If the password contains special characters

add 2 to the **score** variable.

11. Return the **score** variable.

End.

It's pretty common nowadays for websites and online services to offer a password strength meter, enabling users to receive prompt feedback regarding password strength, which in turn assists users in developing the strong passwords. This feedback can change the users' mental processes when choosing passwords and motivate them to create more robust and fortified passwords.

To summarize, score system in password security needs to be viewed as a smart gadget to help users create strong and secure passwords. With the system counting failure from the length and the complexity of the password, users can rationally decide on the strength of their passwords and safeguard their online security more efficiently.

3.2.5 Complexity Score

Password complexity score measures the difficulty an attacker would face in guessing a user’s password. The password's length, character set, and other elements that constitute a password are weighed. A strong password should have a high complexity score, while a weak password will have a low complexity score. Passwords that are easily guessable or crackable can result in security breaches, identity theft, and various forms of cybercrime.

A complexity score is typically computed by evaluating a password’s complexity based on its length, use of upper and lower case letters, numeric representation, and symbols. A higher score shows a stronger password while a lower score indicates a weaker password. Moreover, as passwords require a minimum of 8 characters, length serves as an important element in gauging the strength of the password. Furthermore, there is usually a greater level of strength associated with longer passwords when compared to shorter ones since longer passwords exponentially increase the number permutations an attacker would have to try in order to crack the password.

As given in Algorithm 5, we have put into place a method for calculating the complexity Score of a password. Such algorithm computes the complexity score of a password by analyzing the password's length and its character composition (uppercase, lowercase letters, digits and symbols). The complexity of a password is also analyzed, and corresponding bonus points are assigned based on the criteria defined for the algorithm. The password's length and complexity yield a maximum score of 100. Each component score was calculated based upon the weighting assigned to it and the component scores summed to yield the total score. The complexity score is then derived by dividing the total score by the total possible score. This method helps measure the effectiveness of a password and can be utilized to motivate users to adopt more complex passwords.

Algorithm 5: Calculate the Complexity Score

Input: A string representing the password.

Output: The Complexity Score of the given password.

Begin:

1. Calculate the length of the password.
2. Loop through each character in the password and determine its type (uppercase, lowercase, digit, or symbol).
3. Keep track of the count and presence of each type of character.
4. Determine the number of character types present in the password and the bonus count based on various criteria (password length, presence of both upper and lowercase characters, presence of both digits and symbols, and presence of a number or symbol in the middle of the password).
5. Assign a total possible score of 100 to the password based on its length and complexity.
6. Calculate a score for each component of the password (length, character type count, etc.) using predefined weightings.

---

7. Add up the component scores to obtain the total score.
8. Divide the total score by the total possible score to obtain the complexity score.
9. Return the complexity score.

   End.

The use of different character types, such as uppercase and lowercase letters, numbers, and symbols, also increases the complexity of a password. The more varied the character set, the harder it is for an attacker to guess the password. Password complexity scores may also take into account other factors, such as whether the password contains dictionary words or commonly used phrases, which can make it easier for attackers to guess the password. It is important to note that a high complexity score does not guarantee that a password is completely secure. Passwords should be changed regularly, and users should avoid using the same password for multiple accounts. Additionally, other security measures, such as two-factor authentication, can help protect against password-based attacks.

In conclusion, a password complexity score is an important tool for assessing the strength of a password. By taking into account various factors that contribute to password strength, it can help users create stronger passwords and better protect their accounts from cyber threats.

4. EXPERIMENTAL RESULTS

4.1 EXPERIMENTS

We have implemented the proposed method for password strength tester in Java programming language. The NetBeans 8.2 IDE is used to edit and run the code as a console application. As we mentioned in chapter 3, the following evaluation factors are implemented and tested: Strength Score, Efficiency, Entropy, Security Score and Complexity Score. Our conduct experiments include six password that have manually selected with various composition patterns. The results of testing the strength of the selected passwords and the values of evaluation factors are presented in Table 4-1.

Table 4.1: The Evaluation Factors Values For Six Selected Passwords.

Password	Strength Score	Strength Level	Efficiency	Entropy	Security Score	Complexity Score
“hi12”	2	Weak	13.512 ms	23.8168	3.0	0.42
“#welcome1”	4	Strong	15.62 ms	53.5877	6.0	0.78
“Mypassword”	3	Moderate	14.281 ms	47.0044	4.0	0.68
“My@pass123”	5	Very Strong	19.19 ms	59.5419	7.0	1.02
“Hi1234@#”	5	Very Strong	10.711 ms	47.6336	7.0	1.04
“0001”	1	Very Weak	5.355 ms	13.2877	2.0	0.52

4.2 PASSWORD STRENGTH TESTER BY SERVICE VENDORS

Service providers employ a variety of meters with somewhat different algorithms to gauge the strength of passwords. A comprehensive assessment and analysis of the meters was provided in [42], which enumerated these suppliers' fundamental criteria and is displayed in Table 4-2.

Table 4.2: Password Requirements At Various Vendors.

Service	Strength Scale	Minimum Length	Maximum Length
Apple	Weak, Moderate, Strong	8	32
Dropbox	Very weak, Weak, So-so, Good, Great	6	72
Drupal	Weak, Fair, Good, Strong	6	128
eBay	Invalid, Weak, Medium, Strong	6	20
FedEx	Very weak, Weak, Medium, Strong, very strong	8	35
Google	Weak, Fair, Good, Strong	8	100
PayPal	Weak, Fair, Strong	8	20
Skype	Poor, Medium, Good	6	20
Twitter	Invalid/Too short, Obvious, Not secure enough, could be more secure, Okay, Perfect	6	1000
Yahoo	Weak, Strong, very strong	6	32

4.3 COMPARISON STUDY

These vendors' algorithms produced widely different strength scores for the identical passwords. For instance, as seen in [1], Table 4-3 shows the password "password$1"'s score, which ranges from extremely weak to very strong.

Table 4.3: Password Multi-Checker Output For Password$1.

Service	Strength Level	Strength Score
Our proposed approach	Strong	4 out of 5
Apple	Moderate	2 out of 3
Dropbox	Very weak	1 out of 5
Drupal	Strong	4 out of 4
eBay	Medium	4 out of 5
FedEx	Very weak	1 out of 5
Google	Fair	3 out of 5
PayPal	Weak	2 out of 4
Skype	Poor	1 out of 3
Twitter	Perfect	6 out of 6
Yahoo	Very strong	4 out of 4

4.4 EVALUATE "ROCKYOU" DATASET

The RockYou dataset is a public data breach compilation that was released in 2009. It contains over 32 million passwords that were leaked from the social networking site RockYou. The passwords were stored in plain text and therefore were easily readable for anyone who accessed the breach. The RockYou dataset has become a popular benchmark for password cracking algorithms and security researchers who want to analyze password strength and usage patterns. However, because it contains real user passwords, it is important to handle the dataset with care and use it only for legitimate research purposes.

It's worth noting that using passwords from the RockYou dataset or any other public data breach for personal or business use is highly discouraged. These passwords are likely to have already been compromised and can lead to security vulnerabilities. Rather, it is advised to create strong, one-of-a-kind passwords for every account and to change them on a regular basis. An additional degree of protection can be added by turning on two-factor authentication.

We have apply our proposed method of password strength tester on RockYou dataset with calculating the average Efficiency per a password, Average Entropy and Average Security Score. Moreover, we observed the number of each the five strength levels as shown in Table 4-4.

Table 4.4: The evaluation result on rockyou dataset.

	Level	**Number of passwords**
Strength levels	Weak	1562528
	Very weak	4132485
	Moderate	3959188
	Strong	338595
	Very strong	7204
Efficiency per a password	1.306404 milliseconds	
Average Entropy	47.33214	
Average Security Score	3.5140265	

5. CONCLUSION

This Work introduced the ability to remember a password and whether it is strong enough that it is easy to crack or not. We concluded that most of the users get weak passwords. Most of the article was about how to gain knowledge about information security. Most users have a few passwords that they use repeatedly, and they share them with others when creating a new password. It should consider if the password is weak or strong. Remembering the password and its strength is essential to obtaining a strong and secure password.

In this research, we have described the importance of password in data security in addition to describing how to do password strength assessment. They consist of them, which are the approximate strength of the password and the accuracy of the password. To guess by the attackers or not.

Password strength testers are an essential tool for ensuring that users create secure passwords that are resistant to various types of attacks. The current password strength testers rely on a mix of heuristic evaluation, mathematical operations, and even some machine learning algorithms to evaluate the complexity of a password. With new technology and an evolving threat landscape, there is a need for always ongoing refinement and innovations for better password strength testers. This chapter aims to elaborate on future works on password strength testers.

Advanced Machine Learning Algorithms: Password strength testers can utilize new methodologies focused on the implementation of advanced machine learning algorithms. Nowadays, machine learning has only been limited to classifying passwords into different categories based on identified patterns. With developments in the machine learning algorithms, it becomes possible to create more sophisticated models that can detect new patterns and make better predictions.

Password Strength Assessment using User Behavioral Analytics: Another aspect of work that needs to be addressed in the future is the application of user behavioral analytics as a means to evaluate the strength of passwords. User behavioral analytics analyzes user behaviour through the tracking of certain workflows such as how frequently users change their passwords, the use of trivial passwords, and the cross-clustering of passwords in different user accounts. Patterns like these can be analyzed so that password strength

evaluators can come up with more tailored suggestions for the users and also uncover other weaknesses within the users' systems.

Incorporation of Biometric Data in Password Strength Assessment: The growing use of biometric attributes calls for the inclusion of biometrics in password strength assessment. The use of biometric information like fingerprints or face images allows password strength evaluators to analyze various features and offer better safeguards regarding the security of passwords under different passwords strength evaluation frameworks.

Tailored Password Strength Assessment Based on Risk Factors: Risk-Based Assessments: these deal with other risks such as the type of information being guarded and the chance of a password being compromised. Incorporation of these criteria into analyses increases the chances of offering better tailored suggestions aimed at users and organizations adopting a specific risk strategy.

In closing remarks, further developments for testers of password strength will include the addition of sophisticated machine learning capabilities, user behavior analysis, biometric identification, risk-based evaluations, and systems that do not rely on passwords. By implementing these technologies, password strength evaluations can be more precise in their assessments, tailored suggestions can be offered to various users and institutions, and the security of authentication systems will be heightened. It is important to adapt password strength testers in the face of emerging challenges to defend against possible exploitation and weaknesses.

REFERENCES

[1] https://www.certmike.com/confidentiality-integrity-and-availability-the-cia-triad/

[2] https://www.simplilearn.com/tutorials/cyber-security-tutorial/types-of-cyber-attacks

[3] Claude Castelluccia, Markus Dürmuth, and Daniele Perito. 2012. Adaptive Password-Strength Meters from Markov Models. In Symposium on Network and Distributed System Security (NDSS ’12). The Internet Society, San Diego, California, USA.

[4] Ding Wang, Debiao He, Haibo Cheng, and Ping Wang. 2016. Fuzzy PSO: A New Password Strength Meter Using Fuzzy Probabilistic Context-Free Grammars. In Conference on Dependable Systems and Networks (DSN ’16). IEEE Computer Society, Toulouse, France, 595–606

[5] Matt Weir, Sudhir Aggarwal, Breno de Medeiros, and Bill Glodek. 2009. Password Cracking Using Probabilistic Context-Free Grammars. In IEEE Symposium on Security and Privacy. IEEE Computer Society, Oakland, California, USA, 391–405

[6] Matt Weir, Sudhir Aggarwal, Michael Collins, and Henry Stern. 2010. Testing Metrics for Password Creation Policies by Attacking Large Sets of Revealed Passwords. In ACM Conference on Computer and Communications Security (CCS ’10). ACM, Chicago, Illinois, USA, 162–175

[7] Ur B, Segreti S M, Bauer L, et al. Measuring real-world accuracies and biases in modeling password guessability[C]// Proceedings of the 24th USENIX Security Symposium (USENIX Security). 2015: 463-481.

[8] Melicher W, Ur B, Segreti S M, et al. Fast, lean and accurate: Modeling password guessability using neural networks[C]// Proceedings of the 25th USENIX Security Symposium (USENIX Security). 2016, 175-191.

[9] Van Acker S, Hausknecht D, Joosen W, et al. Password meters and generators on the web: From large-scale empirical study to getting it right[C]//Proceedings of the 5th ACM Conference on Data and Application Security and Privacy. ACM, 2015: 253-262.

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

[11] Narayanan A, Shmatikov V. Fast dictionary attacks on passwords using time-space tradeoff[C]//Proceedings of the 12th ACM conference on Computer and communications security. ACM, 2005: 364-372.

[12] Ma J, Yang W, Luo M, et al. A study of probabilistic password models[C]//Proceedings of the IEEE Symposium on Security and Privacy. IEEE, 2014: 689-704.

[13] Kelley P G, Komanduri S, Mazurek M L, et al. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms[C]//Proceedings of the 2012 IEEE Symposium on Security and Privacy. IEEE, 2012: 523-537

[14] Wheeler D L. zxcvbn: Low-budget password strength estimation[C]//Proceedings of the 25st USENIX Security Symposium (USENIX Security). 2016, 157-173.

[15] Your B, Segreti S M, Bauer L, et al. Measuring real-world accuracies and biases in modeling\ password guessability[C]// Proceedings of the 24th USENIX Security Symposium (USENIX Security). 2015: 463-481.

[16] Hristo Bojinov, Elie Bursztein, Xavier Boyen, and Dan Boneh. Kam- camouflage: Loss-resistant password management. In European symposium on research in computer security, pages 286-302. Springer, 2010.

[17] Hristo Bojinov, Elie Bursztein, Xavier Boyen, and Dan Boneh. Kam- ouflage: Loss-resistant password management. In European symposium on research in computer security, pages 286-302. Springer, 2010

[18] OpenWall. John the Ripper, 2012. http://www.openwall.com/john

[19] Robert Morris and Ken Thompson. 1979. Password Security: A Case History. Commun. ACM 22, 11 (1979), 594–597.

[20] Matt Bishop and Daniel V. Klein. 1995. Improving System Security via Proactive Password Checking. Computers & Security 14, 3 (1995), 233–249. 1990. “Foiling the Cracker”: A Survey of, and Improvements to Password Security. In USENIX Security Workshop. Berkeley, California, USA, 5–14.

[21] Eugene H. Spafford. 1992. Observing Reusable Password Choices. In USENIX Security Symposium (SSYM ’92). USENIX, Berkeley, California, USA, 299–312.

[22] Golla, M., & Dürmuth, M. (2018, October). On the accuracy of password strength meters. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 1567-1582).

[23] Yimin Guo and Zhenfeng Zhang. 2018. LPSE: Lightweight Password-Strength Estimation for Password Meters. Computers & Security 73 (March 2018), 507–518.

[24] Daniel Lowe Wheeler. 2016. zxcvbn: Low-Budget Password Strength Estimation. In USENIX Security Symposium (SSYM ’16). USENIX, Austin, Texas, USA, 157–173.

[25] Ur B, Segreti S M, Bauer L, et al. Measuring real-world accuracies and biases in modeling password guessability[C]// Proceedings of the 24st USENIX Security Symposium (USENIX Security). 2015: 463-481.

[26] Ur, B., Kelley, P. G., Komanduri, S., Lee, J., Maass, M., Mazurek, M. L., ... & Cranor, L. F. (2012). How does your password measure up? The effect of strength meters on password creation. At the 21st USENIX security symposium (USENIX Security 12) (pp. 65-80).

[27] Dupuis, M., & Khan, F. (2018, May). Effects of peer feedback on password strength. In 2018 APWG Symposium on Electronic Crime Research (eCrime) (pp. 1-9). IEEE.

[28] Dupuis, M., Khadeer, S., & Huang, J. (2017). “I Got the Job!”: An exploratory study examining the psychological factors related to status updates on Facebook. Computers in Human Behavior, 73, 132-140.

[29] Philip G Inglesant and Martina Angela Sasse. The true cost of unusable password policies: password used in the wild. ACM, 2010. Edward E Kelley, Franco Motika, and James B Webb. Universal password generation method, January 30 2007. US Patent 7,171,564.

[30] J. Yan. “A note on proactive password checking,”ACM New Security Paradigms Workshop, New Mexico, USA, 2001.

[31] R. Shay, S. Komanduri, P. G. Kelley, P. G. Leon, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor, and S. Egelman. “Of passwords and people: measuring the effect of a password- composition policies,” ACM, 2011.

[32] W. Ma, J.Campbell, D.Tran, and D.Kleeman. “A conceptual framework for assessing password quality,” International Journal of Computer Science and Network Security (IJCSNS), VOL.7 No.1, January 2007.

[33] M.Bishop., “Proactive password checking,”4th Workshop on Computer Security Incident Handling, August 1992.

[34] Microsoft, Available: https://www.microsoft.com/engb/security/ pc-security/password-checker.aspx [Oct.13,2012].

[35] Ray, S. (2019, February). A quick review of machine learning algorithms. In 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon) (pp. 35-39). IEEE.

[36] Vijaya, M. S., Jamuna, K. S., & Karpagavalli, S. (2009, December). Password strength prediction using supervised machine learning techniques. In 2009 international conference on advances in computing, control, and telecommunication technologies (pp. 401-405). IEEE.

[37] Suganya, G., Karpgavalli, S., & Christina, V. (2010). Proactive password strength analyzer using filters and machine learning techniques. International Journal of Computer Applications, 7(14), 1-5.

[38] Jamuna, K. S., Karpagavalli, S., & Vijaya, M. S. (2009). A novel approach for password strength analysis through support vector machine. International Journal of Recent Trends in Engineering, 2(1), 79.

[39] Taveras, P., & Hernandez, L. (2018). Supervised machine learning techniques, cybersecurity habits and human generated password entropy for hacking prediction.

[40] Rathi, R., Visvanathan, P., Kanchana, R., & Anand, R. (2020, February). 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) (pp. 1-3). IEEE.

[41] Dahiya, M. (2017). Password Predictability Using Neural Networks. International Journal of Advanced Research in Computer Science, 8(1).

[42] de Carné de Carnavalet, X., Mannan, M.: From very weak to very strong: analyzing password-strength meters. In: Network and Distributed System Security Symposium (NDSS 2014). Inter-net Society (2014).