Vol.15, Issue No 2, 2025

Improving Password Robustness with Supervised Learning and Deep Learning Models Classification

[1] Bhaskara Sai Venkata Sandeep, [2] Karri Rakesh, [3] Gurujula Divya Sri, [4] Kucherlapati Roopesh Vinay Varma, [5] Mr. Ch. Venkata Reddy

[1,2,3,4] Students, Dept. of CSE, DNR College of Engineering & Technology, Balusumudi, Bhimavaram, India.

[5] Assistant Professor, Dept. of CSE, DNR College of Engineering & Technology, Balusumudi, Bhimavaram, India.

Abstract—

It's safe to assume that text-based passwords will continue to dominate the authentication market. Nevertheless, developers may evaluate their defenses and anticipate their susceptibility to brute-force assaults with the use of machine learning and deep learning algorithms, since these passwords are usually composed of meaningful sequences. By studying user behaviors, advanced approaches like LSTM and GAN may learn to build lists of predicted text passwords that are close to or identical to the ones they are asked to make. In this study, we investigate the possibility of classifying passwords as either strong, moderate, or weak using techniques derived from machine learning. We also assess the potential of deep learning and machine learning to understand the patterns used by hashing algorithms. In addition, we have created a model for password creation that utilizes Gated Recurrent Unit (GRU) to generate new passwords according to discovered patterns. Our goal is to make password creation and management easier for users while also increasing password security. Password-guessing, GRU, LSTM, GAN, and RNN are keyword terms.

I. INTRODUCTION

Many different types of security systems rely on authentication methods, which have many different uses. Electronic commerce platforms, digital computer systems, and cutting-edge mobile phones are just a few of the many businesses that make heavy use of them. Passwords stored in text format, tokens, cards, and even distinctive facial and fingerprint traits may all be used for authentication purposes. To construct text-based passwords, one uses strings that include both alphanumeric characters and extra special characters. Typically used in multi-factor authentication, a token might take the form of a portable USB or smart card. Because of their minimal implementation cost, high availability, and reusability, text passwords are the most used authentication technique [1]. It remains the most significant authentication technique and will remain valid in the future, with all past features being text-based.

Creating robust passwords is all about prioritizing security and making sure you adhere to all requirements. Updated password guidelines [2] were released by the National Institute of Standards and Technology (NIST). These standards include minimum character length, character kinds, password verification, and tries limits. Users often reuse passwords across multiple platforms since remembering them all may be a real pain for online accounts. Millions of user credentials have been stolen due to data breaches that have happened in the last several years. As an example, 2019 saw the theft of more than one billion passwords. Table 1 lists a few of these breaches.

Table 1: A List of famous password breaches globally in 2019. Source: https://haveibeenpwned.com/

Source	*Total of breached passwords*
CafePress	23,205,290
europa.jobs	226,095
Canva	137,272,116
Club Penguin Rewritten	4,007,909
Collection #1	772,904,991
Cracked. to	749,161
EatStreet	6,353,564
EpicBot	816,662

A major security risk to the password-based authentication mechanism is an attack that uses the password-guessing technique. By combining comparable data with repeated passwords, attackers may train a model to understand the pattern of passwords. Unfortunately, attackers are able to discover these patterns due to the large number of compromised credentials. In a brute-force assault, these patterns of passwords may be used as guides to generate possible passwords. Popular rule-based password guessing programs include HashCat and John the Rippe [3]. In order to guess a password correctly, rule-based password-guessing systems may need to try a few different combinations based on pre-established patterns or rules. Password guessing is still a possibility, but it's far less likely with strong, unique passwords and two-factor authentication. One kind of machine learning is neural networks, which may be taught new information about a dataset without requiring any human intervention or prior knowledge. They are now able to identify pictures and comprehend audio or spoken language. Neural networks can learn a lot of characteristics and patterns from compromised passwords, and text passwords are like little sentences. New research shows that neural networks function better when combined with other methods for guessing passwords. In this research, we conducted extensive experiments with several machine learning algorithms to determine the password's strength. Additionally, we proposed a novel model that utilizes deep learning and GRU to create probable passwords. Section II of the remaining article provides a list of the most current relevant papers in the field of password guessing. Section III explains the methods and outcomes of using machine learning to evaluate passwords. Section IV provides the specifics of the proposed GRU model along with the anticipated outcomes. In part V, we wrap up the findings and talk about what's next.

Table 2. Summarization of deep learning models used to guess passwords by generating a new set after learning the patterns.

Model Name	*PassGAN*	*SSPG & DPG*	*[12]*	*GENPass*	*TPGXNN*	*[15]*	*BiLSTM RNN*	*PG-RNN*
Technique used	GAN	Transfer Learning & GAN	bidirectional LSTM	PCFG and LSTM	LSTM and VDCNN	LSTM and a semantic analysis	RNN and LSTM	RNN and LSTM

II. BACKGROUND AND RELATED WORKS

Users may create more secure passwords with the aid of password strength meters, which provide recommendations for factors including complexity, length, and the usage of special characters (e.g., capital letters, numbers, and symbols). In order to ascertain the robustness of a password, the majority of conventional password strength meters are rule-based. Pattern recognition and computer vision problems have both been successfully tackled by deep learning [4]. A GAN, which is comprised of a discriminative and a generative model, was suggested by Ian Goodfellow [5]. The generator is responsible for producing synthetic samples that mimic actual data, while the discriminator is responsible for detecting and rejecting these samples. The generator's job is to make samples that the discriminator can't tell apart from actual data, and the discriminator's job is to pinpoint the phony samples with pinpoint accuracy. In order to teach the generator to create realistic samples of high quality, the discriminator and

generator are adjusted according to their performance. Instead of beginning with a blank slate, a pre-trained model may be used as a foundation for a new job using the transfer learning approach in deep learning. By using a pre-trained model that already has learnt features and representations relevant to the current job, time and resources may be saved in comparison to training a model from beginning. In order to make the pre-trained model work better for the new job, it is common practice to add or modify layers to the model. When dealing with sequential input, a recurrent neural network (ANN) is the way to go since each unit's output is dependent on its prior state. RNNs imitate the behavior of memory. Long short-term memory (LSTM) RNNs [6] overcome the vanishing gradient issue that crops up when regular RNNs are trained on lengthy sequences. Long short-term memory (LSTM) neural networks are able to circumvent the vanishing gradient issue by controlling the influx and outflow of information via memory cells. For this reason, LSTMs excel in environments where language modeling, machine translation, and voice recognition are necessities. To create potential passwords, one uses PassGAN [7]. The method is based on training two networks: one to create password candidates that are comparable to a dataset of stolen credentials, and another to differentiate between generated and actual passwords. PassGAN is a popular tool in the field of security research for assessing password strength and creating novel attack vectors. A powerful tool for password cracking and security assessment, PassGAN learns from real-world passwords to provide password candidates that people are more likely to use. A representation-learning approach to GAN-based password guessing was presented by Dario Pasquini et al. [8]. The approach consists of two parts: substring password guessing (SSPG) and dynamic password guessing (DPG). In response to input from specific sets of passwords, the DPG technique dynamically adjusts its guessing approach. Experimental results demonstrate that representation learning outperforms more conventional approaches when used to password guessing. To model passwords and execute password guessing, Li and colleagues [9] introduced a deep neural network that combined an extended long short-term memory (LSTM) with a bidirectional language model. The bidirectional language model improves the deep neural network's performance, and the extended LSTM is used to model the password and extract features. One such use of deep learning is GENPass [10], a password guessing system that trains and generates candidate passwords using PCFG and LSTM. Through adversarial creation, it enhances efficacy. To train the model using passwords and personal data, TPGXNN [11] used the efficacy of LSTM and VDCNN. In their paper on password guessing, Fang et al. [12] propose a hierarchical semantic model that combines LSTM with a semantic analysis model. The performance and efficiency of the hierarchical semantic model are enhanced by the semantic analysis component, which aids in the avoidance of non-meaningful substring generation. Using structural partitioning and BiLSTM RNN, Zhang et al. [13] suggests a solution for password guessing. Understanding the user's password generating habits is achieved by the structure partitioning module, which models the passwords in the training set. It then generates a collection of common structures and a sorted string dictionary by likelihood. For the BiLSTM module, this string dictionary is used as the training set. When it comes to guessing passwords, RNN is just as frequent as LSTM. For example, PG-RNN [14] uses RNNs to automatically allocate distribution features and character rules derived from compromised password datasets. In Table 2, we compiled a summary of all the models that were described before along with the tools and methodologies used for deep learning to construct the answer. Since memory is a property within LSTM layers, most models use it as a foundation.

III. MACHINE LEARNING AND PASSWORDS STRENGTH

There are a number of security solutions that have made heavy use of machine learning methods in the literature. These include the detection of harmful and phishing websites, the defense against dales injection attacks, and the identification of darknet tor traffic [15]. Passwords should be secure enough to withstand brute force assaults and similar methods. Passwords may be cracked via brute-force attacks, which include attempting every conceivable combination of characters until the right one is identified. Either a human being can try out various combinations by hand, or a computer program may go through a large number of options rapidly. Whether it's for email, social networking, or online banking, a brute-force assault may break the password. A brute-force assault would normally attempt every conceivable combination of characters within a certain duration of time when it comes to passwords. Because of this, the length of a password is directly proportional to the number of potential combinations, and thus the time required to break it.But, even very strong passwords may be broken reasonably fast because to the availability of strong software and hardware cracking tools and the general ease with which brute force assaults can be executed. Passwords should be lengthy and complicated, and users should use a separate password for each account. To further increase the difficulty of an attacker gaining access to an account application, multi-factor authentication should be used. This technology, together with the graphical diversity of goods like spectacles and haircuts, may benefit users.

A) A Model to measure Password Meters:

We use Scikit-Learn into our model to construct a password categorization. The model will provide the password strength on a scale from 0 (the weakest) to 2 (the strongest). The dataset was compiled from several online sources using web scraping techniques. Approximately 6,770,000 passwords of varying strengths are included in the collection. First, we checked for missing values and removed them from the dataset if found. Second, we shuffled the dataset to better understand its patterns and correlations. Finally, we tokenized the dataset.

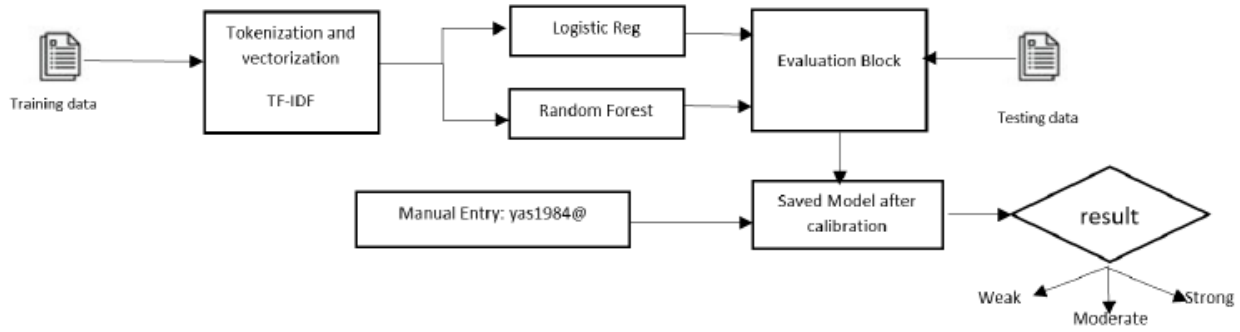

Figure 1: machine learning model for classifying passwords

B) Model to Learn Hashing type

The term "password hashing" refers to the practice of creating a unique fixed-size string of characters by applying a one-way mathematical function on a plaintext password. Instead than keeping the original, plaintext password in a database, this hash is used instead. Whenever a user tries to log in, the system takes the password they entered, runs it through the same procedure, and then compares the hashes. Login succeeded if the two values are identical. Due to the one-way nature of password hashing, it is not feasible to recover the original plaintext password by reversing the process. The significance of this lies in the fact that it renders the password hash database unusable for system login or any other purpose, even in the event that an unauthorized individual were to have access to it.

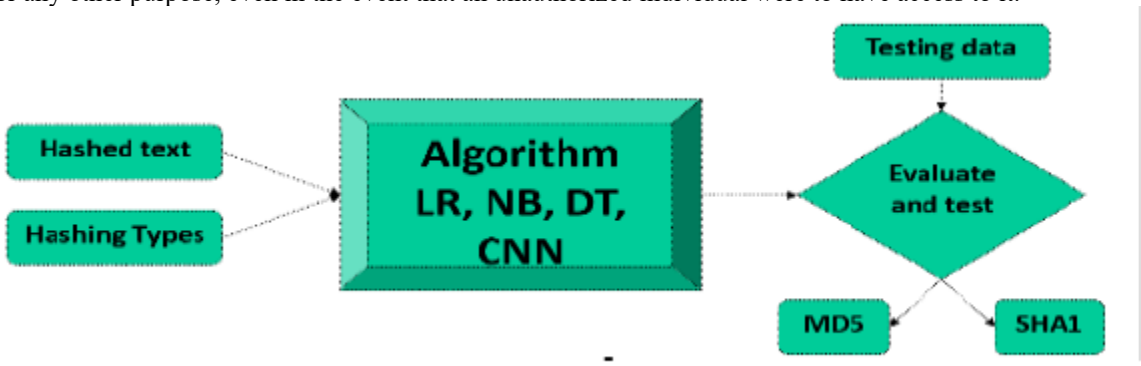

Figure 2: Converting plain text passwords to hashed value

Logistic Regression, Naïve Bayes, and Decision Tree Algorithm are some of the machine learning methods that we use to construct the model. In addition, we inputted data into a Convolutional Neural Network (CNN) to test a deep learning approach. It is used in TFIDF, or term frequency inverse document frequency. In order to adapt machine learning algorithms for prediction, this widely used approach converts the text to a meaningful numerical representation. The primary component of our model is shown in Figure 1. Two methods, Random Forest and Logistic Regression, were used for data training. To find out how likely it is that a target variable would be used as a password in this scenario, we use logistic regression, a supervised learning classification approach. The obtained accuracy rate is 81%. One ensemble learning approach to categorization is Random Forest, often known as random choice Forest. When compared to Logistic Regression's 94.22% accuracy, the results obtained using Random Forest Classifier are much superior. We then ran several standalone predictions for each system, with Random Forest doing

very well (95 percent accuracy out of 200 unique tests). Numerous hashing algorithms are at your disposal, including SHA-256, SHA-512, bcrypt, scrypt, argon2, and many more. The necessary degree of security and the system's performance are two of the application-specific considerations that should be considered while choosing an algorithm. Make sure the hashed passwords aren't easy to break by using a safe and current hashing technique. The model begins by hashing the passwords in plain text. To ensure accuracy, as seen in Figure 2. An attempt was made to construct a model that accepts the hashing value as input and labels it with the hashing type. In the first stage, MD5 and SHA1 are used as hashing algorithms. Afterwards, we used SHA2 and SHA512 to hash the same collection. When we first started, we only tested two hashing algorithms; of these, Logistic Regression proved to be the most effective, with an accuracy rate of 70%. Not even CNN gets close to 40% accuracy. We included four more hashing methods in the second stage. Since no algorithm is more than 25% accurate, the situation is deteriorating. Table 3 shows the total accuracy results for both the first and second stages of all methods.

Table 3: Accuracy results while trying to learn hashing type

ML / DL technique	*The first phase (%)*	*Second Phase (%)*
Logistic Regression	70.2	25.1
Naïve Bayes	27	20
Decision Tree Algorithm	40.1	24
CNN	37	24

IV. SUGGESTED GRU MODEL GENERATES PASSWORDS

Identifying the pattern of password formation was the primary emphasis of our prior study stage. We can learn more about the pattern and the process by which people generate passwords by analyzing the massive amounts of data associated with compromised accounts. Due to its two gates—the "update gate" and the "reset gate"—that regulate the flow of data, GRU was chosen. The update gate determines the amount of data to add to the hidden state and the amount to remove from the previous hidden state. The network is then able to efficiently remember and retrieve the input sequence's long-term dependencies. When fresh input is received, the amount by which the prior concealed state should be reset is determined by the reset gate. The next stage in the process uses the GRU's output—which is determined by combining the current input with the prior concealed state—as input. The GRU is very effective in language modeling, text production, and voice recognition because of this method, which enables it to handle sequential input data. With fewer parameters and less vanishing gradient issue, GRUs are computationally more efficient than LSTM- RNNs.

V. CONCLUSION AND FUTURE WORKS

A text password is a string of characters that consists of letters and numbers. When it comes to handling text data, deep learning models are superior. Both password guessing and measurement of password strength use deeper learning models, according to recent research. Included in this category are representation learning, transfer learning, LSTM, RNN, and GAN. We discovered that our model achieves extremely good accuracy results for password strength analysis, and that other tokenizers may further improve these findings. For password guessing, we proposed a novel model that makes use of deep learning but relies only on GRU. The model has the ability to detect trends in passwords and enhance its guessing capabilities. This model is capable of efficiently generating sets of candidate passwords and evaluating their strength. When compared to more conventional approaches, they perform better and are more useful for text guessing. On the other hand, training dataset quality could affect how well deep learning-based systems perform when it comes to password guessing and strength analysis. We want to keep using the proposed model going forward and fine-tune it using optimal hyper parameters.

REFERENCES

[1]. Melicher W, Kurilova D, Segreti SM, Kalvani P, Shay R, Ur B, Bauer L, Christin N, Cranor LF, Mazurek ML. Usability and security of text passwords on mobile devices. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems 2016 May 7 (pp. 527-539).

[2]. Lemmon EW, Bell IH, Huber ML, McLinden MO. NIST standard reference database 23: reference fluid thermodynamic and transport properties-REFPROP, Version 10.0, National Institute of Standards and Technology. Standard Reference Data Program, Gaithersburg. 2018.
[3]. Hitaj B, Gasti P, Ateniese G, Perez-Cruz F. Passgan: A deep learning approach for password guessing. InApplied Cryptography and Network Security: 17th International Conference, ACNS 2019, Bogota, Colombia, June 5–7, 2019, Proceedings 17 2019 (pp. 217-237). Springer International Publishing.
[4]. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Bengio Y. Generative Adversarial Networks, 1–9. arXiv preprint arXiv:1406.2661. 2014.
[5]. Ayub S, Kannan RJ, Alsini R, Hasanin T, Sasidhar C. LSTM-based RNN framework to remove motion artifacts in dynamic multi-contrast MR images with registration model. Wireless Communications and Mobile Computing. 2022 May 4, 2022.
[6]. M. A. Fauzi, B. Yang, and E. Martiri, "PassGAN Based Honeywords System for Machine-Generated Passwords Database," 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), Baltimore, MD, USA, 2020, pp. 214-220, doi 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00046.
[7]. Pasquini D, Gangwal A, Ateniese G, Bernaschi M, Conti M. Improving password guessing via representation learning. In2021 IEEE Symposium on Security and Privacy (SP) 2021 May 24 (pp. 1382-1399). IEEE.
[8]. Li T, Jiang Y, Lin C, Obaidat MS, Shen Y, Ma J. Deepag: Attack graph construction and threats prediction with bi-directional deep learning. IEEE Transactions on Dependable and Secure Computing. 2022 Jan 18;20(1):740-57.
[9]. Xia, Zhiyang, Ping Yi, Yunyu Liu, Bo Jiang, Wei Wang, and Ting Zhu. "GENPass: a multi-source deep learning model for password guessing." IEEE Transactions on Multimedia 22, no. 5 (2019): 1323-1332.
[10]. Zhou H, Liu Q, Zhang F. Poster: An analysis of targeted password guessing using neural networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (S&P) 2017.
[11]. Fang Y, Liu K, Jing F, Zuo Z. Password guessing based on semantic analysis and neural networks. InTrusted Computing and Information Security: 12th Chinese Conference, CTCIS 2018, Wuhan, China, October 18, 2018, Revised Selected Papers 12 2019 (pp. 84-98). Springer Singapore.
[12]. Zhang, Yi, Hequn Xian, and Aimin Yu. "CSNN: Password guessing method based on Chinese syllables and neural network." Peer-to-Peer Networking and Applications 13 (2020): 2237-2250.
[13]. Bai Q, Zhou J, He L. PG-RNN: using position-gated recurrent neural networks for aspect-based sentiment classification. The Journal of Supercomputing. 2022 Feb;78(3):4073-94.
[14]. Abu Al-Haija, Q., Al-Fayoumi, M. An intelligent identification and classification system for malicious uniform resource locators (URLs). Neural Comput & Applic (2023). https://doi.org/10.1007/s00521-023-08592-z
[15]. Bai Q, Zhou J, He L. PG-RNN: using position-gated recurrent neural networks for aspect-based sentiment classification. The Journal of Supercomputing. 2022 Feb;78(3):4073-94.