ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

Strengthening Identity Protection with Supervised Learning

[1] Shereen Uzma, [2]MOHD NASEER UDDIN, [3] SYED WALEED UDDIN AHMED, [4] RAYYAN QUADRI.
[1]Assistant Professor, Department of CSE-AIML, Lords Institute of Engineering & Technology.
[234] Student, Department of CSE-AIML, Lords Institute of Engineering & Technology.

Abstract

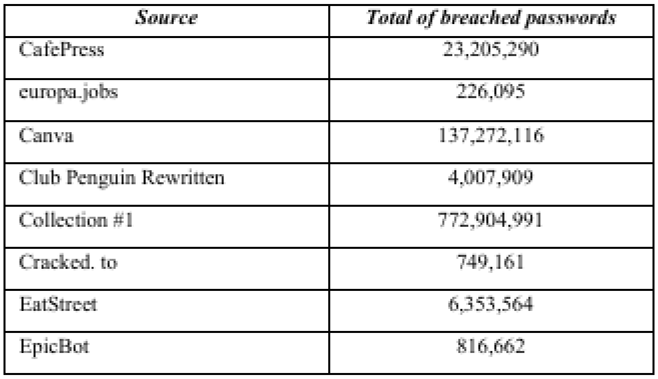
There's no doubt that text-based passwords will continue to dominate the authentication market. Nevertheless, developers may evaluate their defenses and anticipate their susceptibility to brute-force assaults with the use of machine learning and deep learning algorithms, since these passwords are usually composed of meaningful sequences. By studying the patterns that users use when creating and selecting passwords, advanced approaches like LSTM and GAN may provide lists of text passwords that are comparable to and anticipated by users. In this research, we investigate the feasibility of classifying passwords as either strong, moderate, or weak using machine learning methods. We also assess the potential of deep learning and machine learning to understand the patterns used by hashing algorithms. Additionally, we have created a model for password creation that utilizes Gated Recurrent Unit (GRU) to generate new passwords according to patterns that have been learnt. Our goals in using this approach are to make password creation and management easier for users and to make password security better overall.

Keywords—password-guessing, GRU, LSTM, GAN, RNN.

INTRODUCTION

Authentication methods have many uses and are an essential part of many security systems. 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. Authentication may be done in a variety of ways, such as via text-based passwords, tokens, cards, or even just one's unique fingerprint or visage. To construct text-based passwords, one uses strings that include both alphanumeric characters and extra special characters. In the context of multi-factor authentication, a token can be a portable USB drive or a smart card. Because of their minimal implementation cost, high availability, and reusability, text passwords are the most used authentication technique [1]. Because it relies on text for all of its characteristics, it has been and will continue to be the gold standard for authentication. The aim is security when establishing strong passwords, and there are rules to follow. 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 many systems since remembering them is a major difficulty for application and internet accounts. Millions of user credentials have been stolen due to data breaches that have happened in the last several years. Consider that in 2019, hackers gained access to more than one billion credentials. See Table 1 for a list of some of these violations.

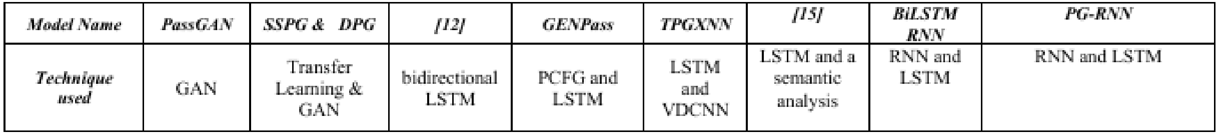
Table 1: A List of famous password breaches globally in 2019. Source: https://haveibeenpwned.com/



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 serious security risk to the password-based authentication mechanism is an attack that uses the password guessing technique. Using patterns in passwords and comparable data, attackers may train a model to guess passwords. Unfortunately, attackers are able to discover these patterns due to the large number of compromised credentials. Using these patterns as a starting point, a brute-force assault may generate passwords. As an example, two well-known programs for rule-based password guessing are HashCat and John the Rippe [3]. It may take more than one try for a rule-based password-guessing program to correctly guess a password since it uses previously established patterns or rules to create potential passwords. But, the likelihood of a successful password guessing attack may be significantly mitigated by using strong and distinct passwords in conjunction with multi-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. Their picture recognition and speech/natural language processing capabilities are fully operational. Passwords in text form are like little sentences, and neural networks can pick up a lot of information from stolen passwords. New research shows that neural networks function better when combined with other methods for guessing passwords. In this research, we proposed a novel model to create possible passwords using deep learning and GRU, and we tested several machine learning methods to determine the password's strength. The rest of the article is structured as follows: Section II provides a summary of current research on password guessing and related topics. Section III explains the methods and outcomes of using machine learning to evaluate passwords. The proposed GRU model is described in depth, along with the anticipated outcomes, in section IV. Part V concludes the findings and discusses next research.

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

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]. Generative Adversarial Networks (GANs) were 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 correctly detect the phony samples. teaching involves making adjustments to the discriminator and generator depending on how well they perform, with the goal of teaching the generator to create realistic examples of high quality. 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. Since the pre-trained model already has learnt characteristics and representations suitable for the current job, it may save time and resources compared to training a model from scratch. In order to adapt the pre-trained model to the new job, it is common practice to insert new layers or modify existing ones. LST and a semantic evaluation The use of RNNs with LSTM A deep neural network for password modeling and guessing was introduced by Li and colleagues [9] using an expanded long short-term memory (LSTM) and a bidirectional language model. To model the password and extract features, we use the extended LSTM. To improve the deep neural network's performance, we use the bidirectional language model. As an additional usage of deep learning, GENPass [10] trains and generates candidate passwords using PCFG and LSTM, making it a deep learning approach for password guessing. It generates adversaries more effectively, which boosts efficacy. When training the model using sensitive information and passwords, TPGXNN [11] made advantage of LSTM and VDCNN's efficacy. A hierarchical semantic model combining LSTM with a semantic analysis model is proposed by Fang et al. [12] as a means of password guessing. In order to improve speed and efficiency, the hierarchical semantic model uses the semantic analysis component to prevent the generation of non-meaningful substrings. Using structural partitioning and BiLSTM RNN, Zhang et al. [13] suggests a solution for password guessing. In order to learn how the user often creates passwords, the structure partitioning module models the training set's passwords. It then produces a set of common structures and a probability-sorted string dictionary. For the BiLSTM module, this string dictionary is used as the training set. 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 have characteristics similar to memory. When training regular RNNs on lengthy sequences, the vanishing gradient problem arises; LSTM [6] RNNs solve this issue. In order to preserve long-term memory and circumvent the vanishing gradient issue, LSTMs use gates to regulate the influx and outflow of information from the memory cell. Linguistic modeling, machine translation, and voice recognition are all areas where LSTMs excel because of this. A password candidate generator is PassGAN [7]. In order for it to function, a generator network is trained to mimic a dataset of compromised passwords and a discriminator network is trained to differentiate between the two. When doing security research, PassGAN is often used to generate new password candidates and assess the strength of existing ones. One practical application of PassGAN is in the field of security assessment and password cracking, since it learns from actual passwords to provide password candidates that users are more likely to really use. Substring password guessing (SSPG) and dynamic password guessing (DPG) are two components of a representation learning technique for password guessing using a GAN that was presented by Dario Pasquini et al. [8]. Feedback from engaging with specific password sets allows the DPG approach to dynamically adjust its guessing strategy. When compared to more conventional approaches, experiments reveal that representation learning performs better when used to password guessing. When it comes to guessing passwords, RNN is just as frequent as LSTM. One such model is PG-RNN [14], which uses RNNs to automatically identify distribution features and character rules in compromised password datasets. In Table 2, we compiled a summary of all the models that were described before together with the deep learning approaches or techniques that were used to construct the answer. Because LSTM layers have the memory attribute, most models employ them as a foundation.

MACHINE LEARNING AND PASSWORDS STRENGTH

The detection of harmful and phishing websites, defense against dales injection attacks, and identification of darknet tor traffic are only a few examples of the many security solutions that have made heavy use of machine learning methods in the literature [15]. A strong password can withstand brute force assaults and other similar threats. To break a password using a brute-force assault, one must methodically attempt every conceivable combination of characters until one finds the right one. One way to do this is by hand, by attempting several combinations, while another is to use a computer program that can rapidly test numerous possibilities. Passwords for all kinds of encrypted accounts, such as email, social media, and online banking, may be cracked using a brute-force approach. When it comes to passwords, a brute-force assault usually involves trying every conceivable combination of characters up to a certain length. 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. However, brute force assaults are becoming easier to execute due to the availability of sophisticated software and hardware cracking tools; as a result, even very strong passwords may be broken reasonably fast. Passwords should be lengthy and complicated, and users should use a separate password for each account. Furthermore, use multi-factor authentication to significantly increase the difficulty of an attacker gaining access to an account application. Products like spectacles and haircuts provide a visual variation that users may take advantage of thanks to this technology. Part A: A Password Meter measurement model: Using Scikit Learn, we construct a model for password categorization. From 0 (the weakest) to 2 (the strongest), the model will display the password strength. 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. As part of the preprocessing steps, the dataset was shuffled to better understand its patterns and correlations, checked for missing values and removed them if found, and then tokenized using Term Frequency Inverse Document Frequency (TFIDF). In order to adapt machine learning algorithms for prediction, this widely used approach converts the text to a meaningful numerical representation. We used two algorithms—Random Forest and Logistic Regression—to train the data, and Figure 1 shows the major block of our model. 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. An accuracy rate of 81% was attained. The Random Forest, often known as the Random Decision Forest, is a categorization ensemble learning algorithm. The Random Forest Classifier outperformed Logistic Regression, which only managed a 94.22% success rate. We then ran several standalone predictions for each system, with Random Forest doing very well (95 percent accuracy out of 200 unique tests).

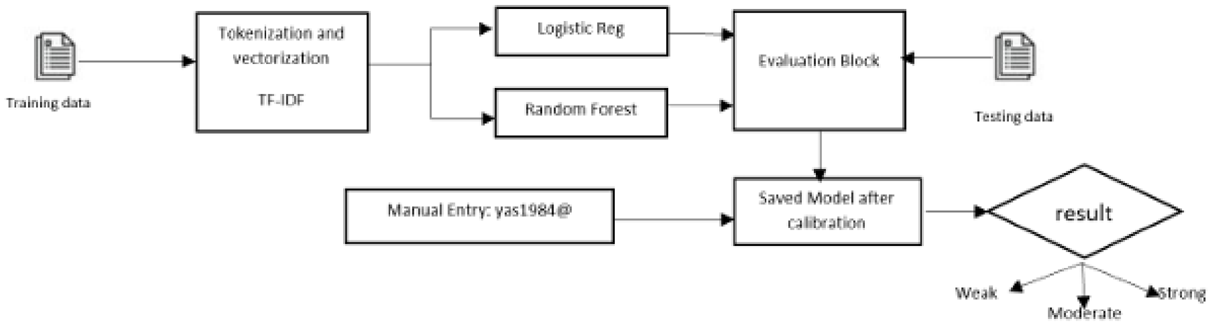


Figure 1: machine learning model for classifying passwords

B) Hashing Type Modul for Learning A one-way mathematical function is applied to a plaintext password in order to generate a unique fixed-size string of characters called a hash. This procedure is known as password hashing. Instead of storing the original password in plaintext, this hash is saved in a database. Whenever a user tries to log in, the system performs the same procedure on the entered password and compares the result to the stored hash. The login was successful if they match. Due to the one-way nature of password hashing, it is not feasible to recover the original plaintext password by reversing the process. This ensures that even if an unauthorized user has access to the database containing password hashes, they will still be unable to access the system or use the plaintext passwords for any other reason. For example, SHA-256, SHA-512, bcrypt, scrypt, argon2, and many more are at your disposal. Considerations like system speed and required security levels dictate the algorithm to be used. Make sure the hashed passwords aren't easy to break by using a safe and current hashing technique. We begin by hashing the raw passwords in our model. Figure 2 illustrates the need for more testing. We made an effort to construct a model that accepts a hashing value and a label indicating the kind of hashing as input. In the first stage, MD5 and SHA1 are used as hashing algorithms. After that, we used SHA2 and SHA512 to hash the identical collection.

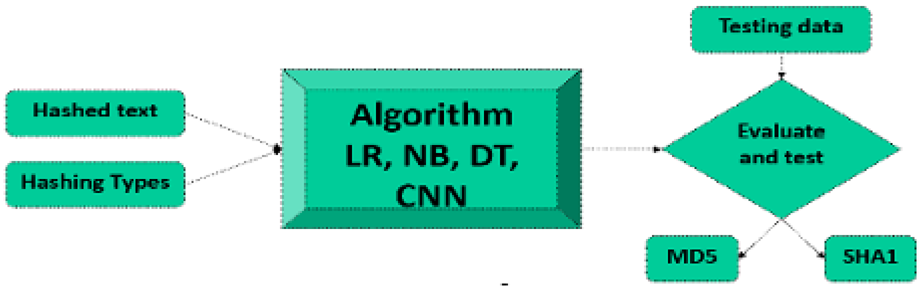
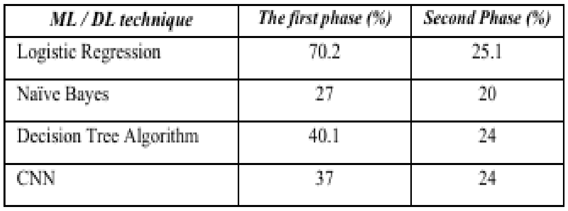


Figure 2: Converting plain text passwords to hashed value

Use of machine learning methods such as Logistic Regression, Naïve Bayes, and Decision Tree Algorithm is used in the model's construction. In addition, we inputted data into a Convolutional Neural Network (CNN) to test a deep learning approach. With 70% accuracy, Logistic Regression emerged victorious from our first trial of only two hashing techniques. Not even CNN gets close to 40% accuracy. We went from two hashing algorithms in the first phase to four in the second. With no algorithm improving its accuracy over 25%, the situation is becoming more dire. All of the algorithms' accuracy results from the first two stages are given in Table 3.

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

SUGGESTED GRU MODEL GENERATES PASSWORDS

We learned the pattern of password generation in our earlier study stage. Knowing the pattern and how people generate passwords will be helpful, as there is a mountain of data on stolen credentials. Since GRU controls the flow of information using two gates—the "update gate" and the "reset gate"—we choose to employ it. The amount of data to be added to the concealed state and the amount to be removed from it are both determined by the update gate. The network is then able to efficiently remember and retrieve the input sequence's long-term dependencies. In order to respond to fresh input, the reset gate decides how much of the concealed state from before should be reset. Next, the current input and the prior hidden state are combined to determine the GRU's output, which is then utilized as input for the next phase in the sequence. 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.

CONCLUSION AND FUTURE WORKS

You may think of text passwords, which are string values that consist of characters and numbers, as text sequences. When it comes to handling text data, deep learning models are superior. The use of deeper learning models in password guessing and strength analysis has been recently shown in research. Among them are transfer learning, representation learning, LSTM, RNN, and GAN. Our approach produces extremely accurate results for password strength analysis, and further improvement is possible with other tokenizers. For the purpose of password guessing, we proposed a novel model that makes use of a deep learning approach using just GRU. The model is able to enhance guessing performance by detecting patterns in passwords. It is possible to construct sets of potential passwords and evaluate their strength using this methodology. They outperform more conventional approaches in terms of accuracy and practicality when it comes to 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 hyperparameters.

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.

[8]. 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.

[9]. 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.

[10]. 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.

[11]. 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.

[12]. 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.

[13]. 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

[14]. 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