Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 12, December-2020

Real Time Password Strength Analysis on a Web Application Using Multiple Machine Learning Approaches

Umar Farooq
Department of Computer Science & Technology,
Central University of Punjab,
Bathinda, India

***Abstract*— Passwords, being the most common mechanism for authentication for its easy implementation, pave a way for attackers to break into accounts by guessing passwords. This happens due to predictable patterns that humans usually set these passwords to like dictionary words, known phrases, names of people and places, keyboard patterns, etc. Many password cracking tools have been made to guess passwords either online or offline that mostly results in cracking such accounts with weak passwords or common password patterns. The proposed model provides a common and efficient way to defend against these online and offline attacks by forcing the users to choose a strong password by implementing multiple machine learning algorithms such as Decision Tree (DT), Naïve Bayes (NB), Linear Regression (LR), Random Forest (RF), and Neural Network (NN) on a web application over real time. This results in logging in into user's account only if the password strength from all algorithms happens to be strong. However, while testing the models over the test set the best results were evaluated by Decision Tree with an accuracy of 99% and the lowest by Naïve Bayes with an accuracy of 87%. Using Burp Suite software we performed three types of password cracking attacks like Brute Force Attack, Dictionary Attack, and Reverse Brute Force Attack on our web application. We made 250 accounts wherein 150 accounts contain strong passwords and other left 100 accounts contain weak passwords. The strong passwords that were set for 150 accounts were not cracked with any of these three types of attacks whereas 86 passwords out of all 100 weak passwords were cracked.**

Keywords*—Passwords, password strength, password analysis, machine learning.

I. INTRODUCTION

A Web Application is programming that utilizes web associated internet browsers and has picked up high significance for performing various tasks in social, business, scholarly, and different stages. These web applications are associated with back-end databases that hold an immense measure of data like usernames, passwords, and so on, and are utilized for correspondence, online exchanges, information stockpiling, getting to interpersonal organizations, and so forth. Regardless of all the significances of these web applications it gives an approach to programmers and wafers to assault these information bases. Securing the web information should be of the absolute significance for engineers of these web applications. Almost 98% of web applications are prone to various attacks such as SQL Injection Attack (SQLI), Cross-Site Scripting (XSS), XML External Entities (XXE), Broken Authentication, etc. [1].

The very first and basic technique, in the field of cyber security, to help secure the web information is Password based authentication [2]. Its reason for being popular is its easy implementation that requires neither any kind of high paid software nor any special hardware [3, 4]. Despite the fact that there are a lot of vulnerabilities in this technique, it still is implemented more often [5]. A password is a secret/mystery word or series of characters that is utilized for verification, to demonstrate identity or access an asset. An ordinary PC client may require passwords for some, reasons: signing in to accounts, recovering email from multiple servers, moving assets, shopping web based, getting to programs, information bases, organizations, sites, and in any event, accessing the morning paper on the web. Passwords are not only critical for login identification, but also in more sophisticated service-granting systems, like Kerberos [6]. For various reasons, including evident security concerns, clients need to utilize various passwords for various frameworks or systems, making it harder to remember and ensure users' passwords [7]. The users then usually choose such a password that is quite simple, easy to remember, easy to guess, and hence ignoring the strength of the password. In the course of recent years, many elective confirmation components (e.g., graphical passwords [8] and multifaceted validation [9] have been recommended, yet passwords obstinately endure and duplicate with each new web systems. This circumstance is to a great extent because of the way that every one of these options has its own noticeable shortcomings when contrasted with passwords [10]. In order to properly comprehend the password security, researchers have gone through many phases and techniques be it heuristic method, Markov-based [11], probabilistic context-free grammar [12, 13] or other algorithms that use personally identifiable information (PII) [14]. These approaches were sparked by the data leakage occurrences that caused a boundless worry in cyber security platform [15, 16, 17].

Password strength is a measurement of defense that it puts against guessing and other kinds of password attacks like brute force and dictionary attacks. The length and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 12, December-2020

complexity are the main factors for making a password strong where length represents the number of characters used and complexity represents the use of combination of different types of characters like numerals, uppercases, lowercases, and symbols. Using a long and complex password lessens the risk of being cracked however the safety is not guaranteed. Any password can be cracked but the thing is some passwords take less time and some take a lot of time to be cracked. In the field of checking the strength of passwords many commercial password strength tools have been made that are based on lexical rules like Google Password Meter (GPM, 2008), Microsoft Password Checker (MPC, 2008), Password Meter (PM, 2008), etc. [18]. In addition to these approaches Decision Trees [19], Enfilter [20] and other tools were made to check the strength of the passwords. Many password guessing tools were also created like HashCat, John the Ripper, PassGAN [3], TarGuess I-IV [2, 21], etc.

II. RELATED WORK

Multiple studies and researches have been carried out so far on the field of password strength analysis and it's detection by using various approaches have been summarized here. In 2012, it was made into account that those password strength meters efficiently increase the strength of passwords that profoundly provide accurate remarks on the passwords [22]. However, majority of users assume that their passwords can provide them with security that actually is in contrast and this was observed in 2015 [23]. This mostly happens due to the inaccurate remarks by the password strength meters [24, 25]. Based on heuristic rules, entropy based approach is an ad hoc password strength meter that happened to be influential [26]. Gmail, Yahoo, PayPal, Microsoft, and other online PSMs use the principle of NIST password strength meter [25]. However, it was demonstrated that this entropy-based methodology just gives an unpleasant estimation of password strength as estimated as far as true speculating obstruction [27, 28]. The more compelling measurement for secret phrase strength is "guessability" [22, 28], which identifies with genuine security and describes the time complexity needed for a password breaking calculation to recuperate a record. This guessability is further classified into two types: online and offline guessing [29]. Further trawling guessing and targeted guessing are two types of guessing that depends upon whether involving the user information or not [24].

III. METHODOLOGY

The main motive of the proposed model is to analyze the password strength in real time web application by implementing multiple machine learning approaches. The whole procedure is carried out in four stages as depicted in Figure (Fig. 2.).

1. The first phase focuses on collecting the meaningful dataset that contains a huge amount of passwords that are used to check the strength of passwords. For this phase, we have collected a dataset that contains passwords with three types of strength as weak, medium and strong.
2. The second phase focuses on preprocessing methods and extracting features from the dataset followed by the TF-ID Vectorizer. The labelling of the dataset is done in this stage. The models are then trained with 70% of the dataset (a.k.a. Training Part).
3. The third phase focuses on using the 30% of the dataset that we separated from the actual collected dataset for testing and evaluating the proposed model (a.k.a. Testing Part).
4. The fourth phase focuses on implementing the proposed model on the web application that we also designed for the said problem in order to analyze the strength of the passwords in real time. The main motive of the proposed model is to force the users to select a strong password.

A. *Dataset*

The most important factor in analyzing the strength of the passwords using machine learning is collecting a meaningful dataset that contains a huge amount of passwords with strength as weak, medium, and strong. The main contribution in this paper is a labelled dataset that we manually created for the said problem. The dataset contains weak, medium, and strong passwords with labelling as 0, 1, and 2, respectively. The dataset is created in three phases: 1) the weak passwords are generated in this phase; 2) the medium passwords are generated in this phase, and 3) the strong passwords are generated in this phase. We generated these passwords in the text (.txt) format and then applied certain preprocessing methods and converted them into a csv (.csv) file. The dataset contains a total of 700000 passwords. The dataset has 3 categories and each has been created with different conditions:

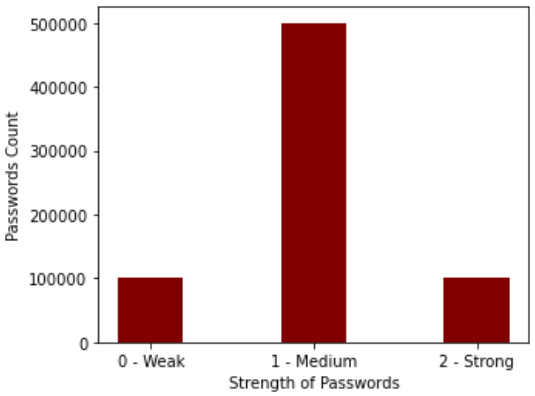


Fig. 1. Class count

a) Weak Passwords

We started with creating the weak passwords at the first place by using own python code, where in we created 4 sets of weak passwords. The first set of weak passwords was generated by using numerals (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9) only and we saved it as a text file under the name "weak set 1.txt". The second set of weak passwords was generated by using lowercase characters (a-z) only and we saved it as a text file under the name "weak set 2.txt".

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 12, December-2020

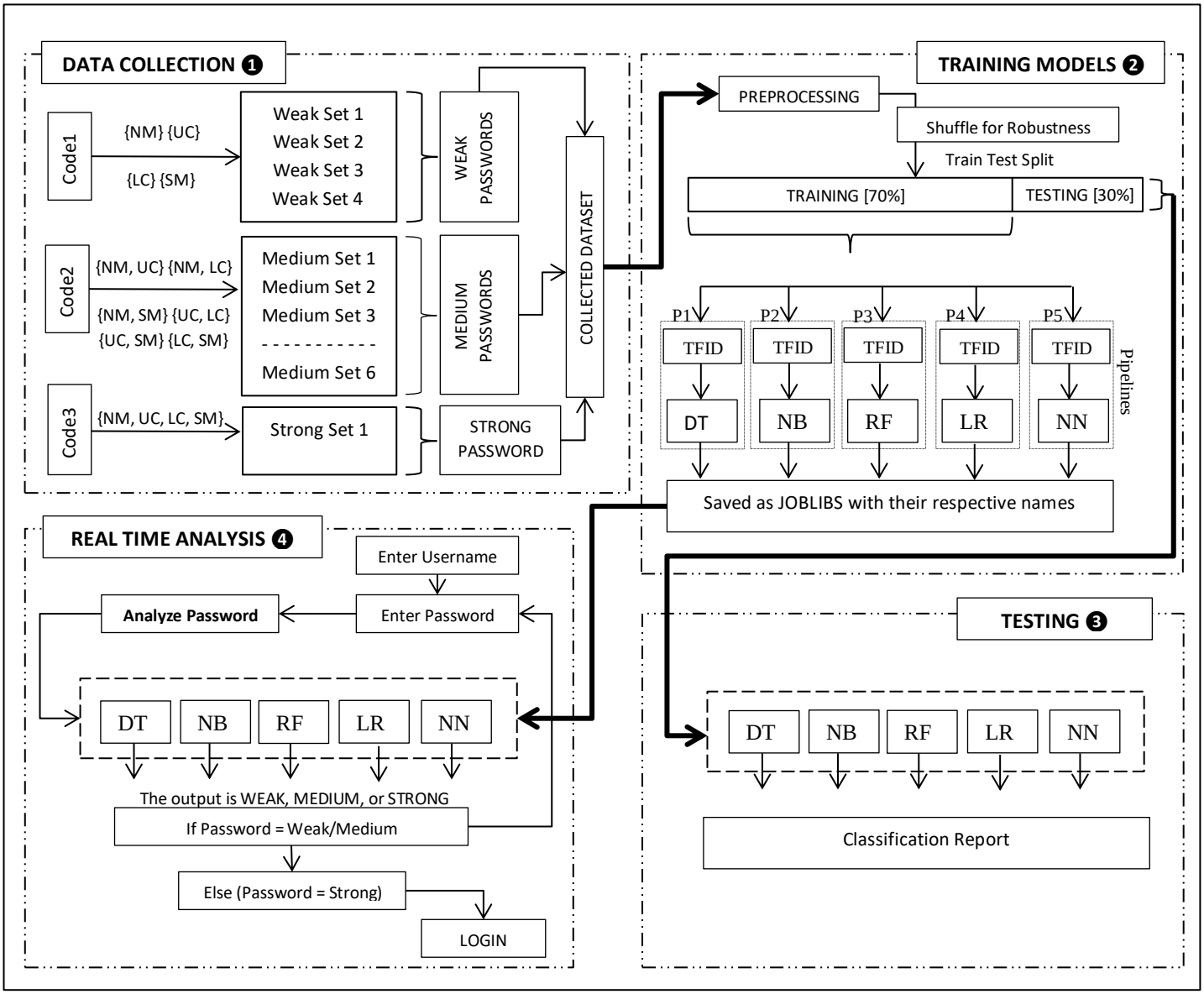


Fig. 2. Overall methodology and workflow of our proposed model

The third set of weak passwords was generated by using upper case characters (A-Z) only and we saved it as a text file under the name "weak set 3.txt". The fourth set of weak passwords was generated by using symbols (!, @, #, $, %, ^, &, *, (,), <, >, /, and ?) only and we saved it as a text file under the name "weak set 4.txt". We then joined these text files and saved it as a new text file under the name "Weak_Set.txt".

TABLEI. COUNT AND RATIO OF LABELS

Class	Labelled as	Count	Ratio
Weak	0	100000	14.286%
Medium	1	500000	71.428%
Strong	2	100000	14.286%
Total		**700000**	

b) Medium Passwords

The second phase in the creation of dataset was creating medium passwords. For this issue we generated 6 set of medium passwords. The first set contains passwords that are generated with numerals and lowercase characters only, the second set with numerals and uppercase characters only, the third set with numerals and symbols only, the fourth set with uppercase and lowercase characters only, the fifth with uppercase and symbols only, and the sixth set with lowercase and symbols only. These six sets were saved as "medium set 1.txt", "medium set 2.txt", "medium set 3.txt", "medium set 4.txt", "medium set 5.txt", and "medium set 6.txt", respectively. Finally we joined them all and then saved them as a text file under the name "Medium_Set.txt".

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 12, December-2020

c) Strong Passwords

The third phase in the creation of dataset was creating strong passwords. We generated a single set of strong passwords that contains a mixture of numerals, uppercases, lowercases and symbols with length greater than 8 characters. This set was also saved as a text file under the name "Strong_Set.txt".

B. Training Models

The main phase is to train the machine learning algorithms for the detection of password strength over the manually collected dataset and then applying the models for the real time analysis of password strength. But before that there are some preprocessing steps that are to be done on the dataset. The three text files that contain weak, medium, and strong passwords are then joined and converted to a csv (.csv) file. The labels are encoded with one hot encoding. Then the missing data is handled, we could have replaced the missing data with mean but we decided to delete the missing data and shuffled it for robustness. We then used pipelines where each pipeline starts with TF-ID Vectorizer for assigning a numeric value to each password and ends with fitting the model. The machine learning algorithms that we used for the proposed work are Decision Tree (DT), Naïve Bayes (NB), Linear Regression (LR), Random Forest (RF), and Neural Network (NN). To have a better understanding of how the machine learning models would perform over the testing data we applied 10-fold cross-validation where we split the dataset into 10 parts. The advantage of cross validation is that all the observations are utilized for both training and testing the models, and each observation is used for testing exactly once. The trained models after evaluating the training accuracy are saved as joblib files (.joblib) that are to be served while implementing the proposed model over real time on a web application. The training accuracy and training time is depicted in the figure below (Fig. 3.).

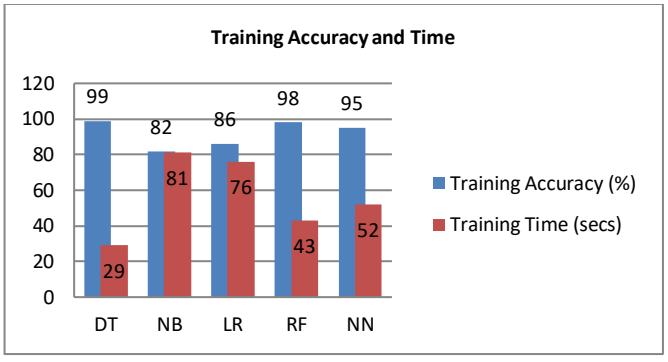
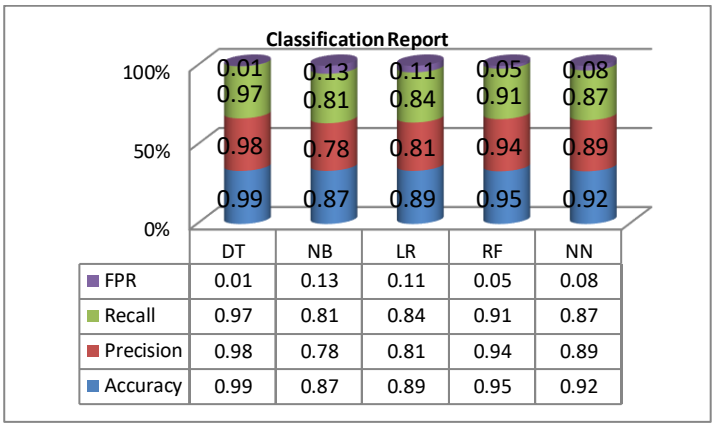


Fig. 3. Training accuracy and training time by proposed models

IV. RESULTS AND DISCUSSION

A. Testing

As per the experiments that we conducted, we come to conclusion that our proposed system is enough to detect and analyze the strength of passwords. To evaluate the performance of our proposed model we applied the algorithms on the testing data (30% of the original dataset). The overall classification model is depicted in the below figure (Fig. 4.).



	DT	NB	LR	RF	NN
FPR	0.01	0.13	0.11	0.05	0.08
Recall	0.97	0.81	0.84	0.91	0.87
Precision	0.98	0.78	0.81	0.94	0.89
Accuracy	0.99	0.87	0.89	0.95	0.92

Fig. 4. Classification report of our proposed model

B. Real Time Analysis

The proposed system is implemented by using multiple programming languages like HTML5, CSS3, JavaScript, PHP, and also Python. HTML5 and CSS3 are used to create a webpage with some back-end functions with JavaScript. The back-end database is connected by using PHP. The Python language is used to implement the real time analysis of passwords. The overall design of the web application is depicted below (Fig. 5.).

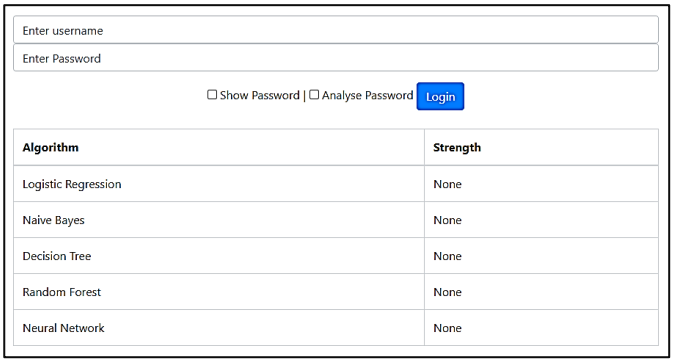


Fig. 5. Snapshot of the web application built for the proposed model

The user cannot login before analyzing the strength of passwords, and if the strength from all algorithms is strong then only can user login to the account. We tried the model for some passwords and received the results as depicted in below table (Table II).

TABLEII. STRENGTH EVALUATED BY PROPOSED MODEL

Password	Strength by Algorithms				
	DT	NB	LR	RF	NN
Umar1234	Weak	Weak	Weak	Weak	Weak
Umar@123qwe	Medium	Medium	Medium	Medium	Medium
Umar#!&$#Far0o@Q	Strong	Strong	Strong	Strong	Strong

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 12, December-2020

C. *Password Guessing*

In order to properly analyze how our model performed, we planned to perform certain attacks by using Burp Suite [30] software on our web application for cracking accounts. For this issue we manually made 250 accounts with usernames to be from "user1" to "user250" and passwords with different patterns. The accounts were divided and created from weak and strong passwords wherein 150 accounts contain strong passwords and 100 accounts contain weak passwords. The attacks we performed are:

a) Brute Force Attack

In this attack the attacker tries many passwords against a single username until it matches and gets access to the account (Fig. 6.). However, the attack can be done automatically on multiple usernames with auto generated passwords wherein all passwords will be checked against the first username and afterward correspondingly against others. The first payload that we used in Burp Suite is list of usernames and the second one is list of passwords.

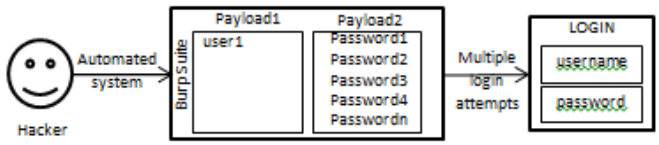


Fig. 6. Brute force attack

b) Reverse Brute Force Attack

In this attack the attacker tries a single password against multiple usernames until it matches and gets access to the account (Fig. 7.).

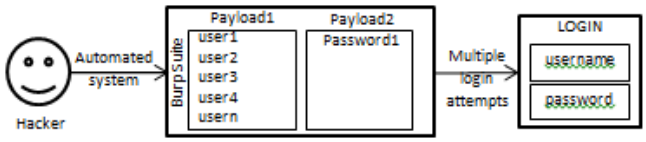


Fig. 7. Reverse brute force attack

c) Dictionary Attack

It is a type of brute-force attack where an attacker tries to crack a password with a dictionary list that contains words and phrases (Fig. 8.). We created a text file (dictionary list) that contains numerous random passwords, names, places, phrases, passwords from data breaches, etc.

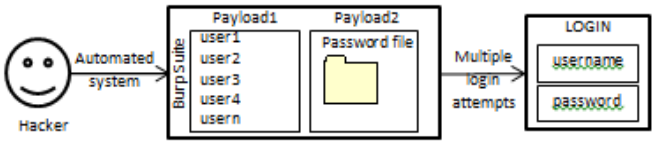


Fig. 8. Dictionary attack

Upon performing all three attacks on our web application, not a single account from 150 accounts with strong passwords were cracked while as 86 accounts from 100 accounts with weak passwords were cracked. However, there is no surety that it will be secure from other attacks or vulnerability not yet discovered/made public as our main motive was to analyze the strength of passwords. The attacks if prolonged can result in cracking the accounts with strong passwords also, however that can defended by limiting the number of login attempts. Moreover other mechanisms can also be implemented along with the proposed system like multi-step authentication, authentication by verification code sent to phone number or email, etc. This indicates the efficiency of our proposed system and capability for implementing it in real time systems.

V. CONCLUSION

In this research work we proposed a password strength detection and analysis system by implementing multiple machine learning approaches on a web application over real time. The main target of our system was mainly to force the users to choose strong passwords with complex patterns and without any known or common patterns in order to make it difficult for hackers and crackers to crack the passwords. The foremost step that we took to deal with this issue was creating a dataset that contains numerous passwords with different patterns. The machine learning models that we implemented performed well whereas the best results were evaluated by Decision Tree with an accuracy of 99%. The other models such as Naïve Bayes, Linear Regression, Random Forest, and Neural Network achieved accuracy of 87%, 89%, 95%, and 92%, respectively. In order to check how defensive role our system plays in protecting the accounts from being cracked, we performed brute force, reverse brute force, and dictionary attack on our web application. The results were quite interesting as none strong passwords were cracked by these three attacks while as many weak passwords were been cracked. From the results, we come to conclusion that our proposed model is enough to be implemented in real time detection and analysis systems in order to force the users to choose strong passwords and give a hard time to hackers to crack the passwords.

REFERENCES

[1] "OWASP," 2020. [Online]. Available: https://owasp.org/www-project-top-ten/. [Accessed 11 12 2020].

[2] Z. Xie, M. Zhang, Y. Guo, Z. Li and H. Wang, "Modified Password Guessing Methods Based on TarGuess-I," *Wireless Communications and Mobile Computing*, pp. 1-2, 2020.

[3] B. Hitaj, P. Gasti, G. Ateniese and F. Perez-Cruz, "PassGAN: A Deep Learning Approach for Password Guessing," in *17th International Conference, ACNS 2019, Bogota, Colombia*, 2019.

[4] T. Hunt, "Here's why [insert thing here] is not a password killer," 2018. [Online]. Available: https://www.troyhunt.com/heres-why-insert-thing-here-is-not-a-password-killer/.

[5] D. Wang, D. He, H. Cheng and P. Wang, "fuzzyPSM: A New Password Strength Meter Using Fuzzy Probabilistic Context-Free Grammars," in *46th Annual IEEE/IFIP International Conference on*

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 12, December-2020

Dependable Systems and Networks, 2016.

[6] E. Zouave, M. Bruce, K. Colde, M. Jaitner, I. Rodhe and T. Gustafsson, "Artificially intelligent cyberattacks," *FOI*, 2020.

[7] S. G, K. S and C. V, "Proactive Password Strength Analyzer Using Filters and Machine Learning Techniques," *International Journal of Computer Applications (0975 – 8887)*, vol. 7, no. 14, pp. 1-2, 2010.

[8] Z. Zhao, G.-J. Ahn and H. Hu, "Picture gesture authentication: Empirical analysis, automated attacks, and scheme evaluation," *ACM Trans. Inform. Syst. Secur.*, vol. 17, no. 4, p. 1–37, 2015.

[9] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE Trans. Depend. Secur. Comput*, 2016.

[10] J. Bonneau, C. Herley, P. Oorschot and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *IEEE S&P*, 2012.

[11] J. Ma, W. Yang, M. Luo and N. L, "A study of probabilistic password models," in *2014 IEEE Symposium on Security and privacy*, San Jose, CA, USA, 2014.

[12] M. Weir, S. Aggarwal, B. D. Medeiros and B. Glodek, "Password cracking using probabilistic context-free grammars," in *2009 30th IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, 2009.

[13] R. Veras, C. Collins and J. Thorpe, "On semantic patterns of passwords and their security impact," in *Proceedings 2014 Net work and Distributed System Security Symposium*, San Diego, CA, USA, 2014.

[14] Y. Li, H. Wang and K. Sun, "A study of personal information in human-chosen passwords and its security implications," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, San Francisco, CA, USA, 2016.

[15] S. Ji, S. Yang, X. Hu, W. Han, Z. Li and R. Beyah, "Zero-sum password cracking game: a large-scale empirical study on the crackability, correlation, and security of passwords," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 5, pp. 550-564, 2017.

[16] Z. Li, W. Han and W. Xu, "A large-scale empirical analysis of chinese web passwords," in *23rd USENIX Security Symposium (USENIX Security 14)*, San Diego, CA, USA, 2014.

[17] M. K. L. Gong-Shen, Q. Wei-Dong and L. Jian-Hua, "Password vulnerability assessment and recovery based on rules mined from large-scale real data," *Chinese Journal of Computers*, vol. 39, no. 3, p. 454–467, 2016.

[18] J. KS, K. S and V. MS, "A Novel Approach for Password Strength Analysis through Support Vector Machine," *International Journal of Recent Trends in Engineering*, vol. 2, no. 1, pp. 79-82, November 2009.

[19] F.Bergadano, B.Crispo and G.Ruffo, "Proactive password checking with decision trees," in *Proc. of the 4th ACM conference on computer and communications security*, Zurich, Switzerland, 1997.

[20] G. Ruffo and F. Bergadan, "EnFilter : A Password Enforcement and Filter Tool Based on Pattern Recognition Techniques," *Springer Berlin / Heidelberg*, vol. 3617, no. 1611-3349 (Online), 2005.

[21] D. Wang, Z. Zhang, J. Y. P. Wang and X. Huang, "Targeted online password guessing: an underestimated threat," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna Austria, 2016.

[22] B. Ur, P. G. Kelley, S. Komanduri, M. M. J. Lee, M. Mazurek, T. Passaro, R. Shay, T. Vidas and L. Bauer, "How does your password measure up? the effect of strength meters on password creation," in *Proc. USENIX SEC 2012*, 2012.

[23] B. Ur, F. Noma, J. Bees, S. M. Segreti, R. Shay, L. Bauer, N. Christin and L. F. Cranor, "I added '!' at the end to make it secure: Observing password creation in the lab," in *Proc. SOUPS 2015*, 2015.

[24] D. Wang and P. Wang, "The emperor's new password creation policies," in *in Proc. ESORICS 2015*, 2015.

[25] X. Carnavalet and M. Mannan, "A large-scale evaluation of high-impact password strength meters," *ACM Trans. Inform. Syst. Secur*, vol. 18, no. 1, pp. 1-32, 2015.

[26] W. Burr, D. Dodson, R. Perlner, S. Gupta and E. Nabbus, "NIST SP800-63-2: Electronic authentication guideline," *National Institute of Standards and Technology, Reston, VA, Tech. Rep.,*, August 2013.

[27] M. Weir, S. Aggarwal, M. Collins and H. Stern, "Testing metrics for password creation policies by attacking large sets of revealed passwords," in *Proc. ACM CCS 2010*, 2010.

[28] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor and J. Lopez, "Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms," in *in Proc. IEEE S&P 2012*, 2012.

[29] D. Florencio, C. Herley and P. C. V. Oorschot, "Password portfolios and the finite-effort user: Sustainably managing large numbers of accounts," in *Proc. USENIX SEC 2014*, 2014.

[30] "PortSwigger," PortSwigger Ltd., 2020. [Online]. Available: https://portswigger.net/burp. [Accessed 13 12 2020].