PREDICTING PASSWORD STRENGTH BASED ON NATURAL LANGUAGE PROCESSING TECHNIQUE

BY

Ahsan Kabir
ID:161-15-731

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Md. Sadekur Rahman
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised By

Mr. Ahmed Al Marouf
Lecturer
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

AUGUST, 2022

APPROVAL

DECLARATION

I hereby declare that this thesis has been done by me under the supervision of **Md. Sadekur Rahman, Assistant Professor, Department of CSE** Daffodil International University. I also declare that neither this thesis nor any part of this thesis has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

——----------

Md. Sadekur Rahman
Assistant Professor Department of CSE
Daffodil International University

Co-Supervised by:

——----------

Mr. Ahmed Al Marouf
Lecturer
Department of CSE
Daffodil International University

Submitted by:

——----------

Ahsan Kabir
ID:161-15-7301
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, I want to express gratitude to the Almighty God for His endless kindness in keeping me mentally and physically fit to complete this project.

This project would not get its own shape without the support of Daffodil International University which provided me the chance for a B.Sc. in computer science and engineering. This project is an aggregate of co-operation of many persons.

Foremost, I would like to express my deepest gratitude to my honorable supervisor, **Md. Sadekur Rahman, Assistant professor**, **Department of CSE** Daffodil international university. His vast knowledge, attitude, behavior, and wisdom have given me the strength to work harder every day. His directions and guidelines for preparing manuscripts, reports, and presentations made me more dynamic and well-organized throughout the whole project. He gave me enough time from the beginning of the project, though he has a tough schedule for his own task.

I would also like to thank **Professor Dr. Touhid Bhuiyan, Professor and Head**, **Department of CSE**, for his motivation and appreciation. And all of my honorable teachers in the department for giving me the proper guideline throughout the entire semester.

I want to thank all of my classmates who have always inspired, helped, and motivated me. I also wish to thank all of my seniors for their utmost support.

ABSTRACT

Passwords provide the first line of defense against unauthorized access to your computer and personal information. Though there are many alternatives to passwords for access control, a password is the more compellingly authenticating the identity in many applications. They provide a simple, direct means of protecting a system and they represent the identity of an individual for a system. So, life these days have become largely dependent on password for many purposes. Logging in to computer accounts, retrieving email from the server, transferring funds, online shopping, accessing programs, databases, networks, websites, and even reading the morning newspaper online. The problem of selecting and using good passwords is becoming more important every day.

In this work password strength prediction is modeled as a classification task and supervised machine learning techniques were employed. Widely used supervised machine learning techniques namely Logistic Regression, Naïve Bayes Classifier, Support Vector Machine, and Gradient Boosting Algorithms were used for learning the model. The proposed model was applied to two different datasets and states that this model is stable in terms of accuracy. The results of the models were also compared with the existing password strength checking tools. The findings show that the machine learning approach has substantial capability to classify the extreme cases – Strong, Medium, and Weak passwords.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter provides a detailed, but introductory look at the problem statement and objectives behind one project. It also discusses some experiments carried out in this work as well as how it ties together with other pieces of literature or theoretical concepts for discussion purposes.
Furthermore, it discusses thesis contribution as well as book organization which outlines what's going to happen next.

1.2 Motivation

Passwords are clearly essential to security, but there are many other methods that can or should be used to protect one's information. Especially during this pandemic and after the pandemic, the world will step up even more in terms of using the internet, internet-based devices, and facilities. To create a good password, people should learn how to safeguard and use it properly. My motivation is to take the accuracy to a max level that can be helpful for protecting our valuable information, access, and online-based activities.
If my project can contribute a little to this sector, my dedication and hard work will be successful.

1.3 Rationale of the study

The industry standard for measuring password strength is in the phase of information entropy. Password strength is often specified in terms of information entropy, which measures how much data there is to guess. This number represents the amount and type

of data stored within a character string, rather than just how many guesses one needs before finding their correct response- so 42 bits means that any potential passwords would require about 4 million trillions (or approximately) tries at least!

For example a password with an abundance of 42 bits would require four times as many attempts at cracking it compared those that only have 13-17 extractable possibilities for each bit (depending on what type you're trying). This makes sure your passwords aren't easily soluble by hackers - they'll need more tries before finding out!

1.4 Research Questions

During the research work, some question occurs about this work. The main question of our work is given below:

1. How do collect and preprocess data?
2. Which classifiers perform better?

1.5 Expected Output

The data augment method expands the (GTSTD) by simulation of complex environmental changes. I conduct different experiments to verify the effectiveness and robustness of my proposed model. The accuracy rate and the recall rate of my method are 98.09 percent in Gradient Boosting Algorithm. The dataset we've created has around 2 million data which is absolutely stable for any prediction.

1.6 Report Layout

The rest of this project is distributed into the following chapters:

- In chapter 1 I mention my whole research work's outline and divided this chapter into multiple subchapters. For example, the introduction, motivation, rationale of the study, research question, and expected output of my project.
- In Chapter 2 I have discussed the previous work on Password Strength Prediction and calculation, the scope of the problem, and the challenges in this work.
- In Chapter 3 I will talk about my work procedure, methods, and techniques to predict Password Strength.
- In Chapter 4 I will discuss the Experimental Results and Discussion of my proposed model.
- In Chapter 5 I will talk about the Impact on Society, the Environment, Ethical Aspects, and the Sustainability of my work.
- In Chapter 6 I have discussed the Summary, Conclusion, and Further Study of the work.

CHAPTER 2

BACKGROUND

2.1 Terminologies

I have always been interested in the field of computer security and wanted to find new and innovative ways to protect our valuable information. A few years ago, I became interested in natural language processing (NLP) and decided to apply it to the task of predicting password strength. NLP is the study of how humans use language, and I thought it could be used to determine how strong a password is based on the words that are used. I came up with a system using TF-IDF, a technique that judges how important a word is to a document. I trained my system on a dataset of passwords, and it was able to predict the strength of a password with an accuracy of over 98 percent.

2.2 Related Work

In a study performed by Zviran Haga (1999) which did a survey on password security, the subjects of this survey were computer users at the Department of defense in California. The questionnaire was distributed to two thousand users, and 49.9 percent (979) answered the survey. The authors identified that according to several sources that an acceptable password should have between 6 and 8 characters. They found that 47 percent of the respondents to the survey had a password shorter than this. Only 14.1 percent of the users had a password that consisted of 8 characters or more which are today's suggested standard by many guidelines. Above 79 percent of the users never changed their password 14.9 percent changed it on an annual basis, and only 5.5 percent changed it several times a year. 80 percent of the users conducting the survey also had a password that only consisted of alphabetic characters. 78 percent of the users based
their password on a combination of meaningful details, like the data they protected or personal information.

In 2014 a study was performed by Taneski et al. (2014a) at the University of Maribor where they used an online questionnaire to determine the characteristics of textual passwords. They had a group of 33 students at the Faculty of EEE and CSE at the university conducts the survey. Two phases survey, in phase one students, performed the questionnaire without any education in password security. After the first phase, the students attended a lecture designed by the authors which consisted of topics on how to create a strong password and ways of managing them. After the lecture, they had a two-week period before they contacted the students again to ask them to perform the second part of the survey. They then compared the data from the two different sections the questions on the survey included:

- Average password length.
- Password change frequency.
- Password memorability and write-down.

2.3 Comparative analysis and Summary

This section finally shows the comparison of accuracy among several other models and our proposed model. Though creating an automated model for recognizing password strength.

TABLE 1: COMPARE BETWEEN DIFFERENT MODEL

Serial Number	Method	Accuracy
1	Logistic Regression	83%
2	Random Forest	81%
3	Naive Bayes	86%
4	My Proposed Model	98%

2.4 Scope of the Problem

Many problems that this project seeks to address. The first is that people tend to use rememberable passwords, also easy. This means that many people use words that are found in dictionaries, names, dates, and others easily guessed. The second problem is that people often reuse passwords across multiple accounts. This means that if one password is compromised, all of the accounts that use that password are also at risk. The third problem is that people often choose passwords that are too short, which makes them easier to guess.

The idea of this model is to create a system that can predict how strong a password is based on the words that are used. This will help people to choose stronger passwords that are less likely to be guessed, and it will also help to encourage people to use different passwords for different accounts. The system will use TF-IDF to measure the importance of each word in a password, and it will also take into account the length of the password. The hope is that by using this system, people will be able to choose stronger passwords that are less likely to be guessed or compromised.

Calculate Password Entropy:

What password entropy is and its calculation is important to understand. The measure of a password pattern and how difficult to break. Calculating entropy is an easy task. For example, the password “mynameissagor” would have a possible pool of 26 characters. Changed the password to "MyNameIsSagor", the pool of characters would increase to 52. log2(x) formula is used to calculate entropy.

TABLE 2: PASSWORD ENTROPY CALCULATION

Type	Pool of Possible Characters
Lowercase / Uppercase	26
Lowercase & Uppercase	52
Alphanumeric	36
Alphanumeric & Uppercase	30
Common ASCII Characters	62
Diceware Words List	7776
English Dictionary Words	171000

N-GRAM Approach:

Another approach to analyzing the dataset is by counting the occurrences of small-sized character pairs that make up the password. The first step in this process is to extract every possible combination of characters from the dataset. This is necessary to calculate the probability term of an n-gram-based likelihood calculation. After this is done, the next step is to create an n-gram model based on these combinations. This model can then be used to measure the likelihood of a password or to generate new passwords. This approach is superior to a brute-force attack strategy because it is smarter and more efficient. By using an n-gram model, we can more accurately assess the strength of a password and create more secure passwords.

Password Cracking:

The most important challenge is choosing the right tools. Many tools are available for password cracking. John the Ripper is one of them. It is a free and open-source cracker.

In the beginning, it was developed for UNIX operating system. after that it was extended to all platforms. Weir et al are the other most popular techniques that use in recent days.

Characteristics of weak passwords:

Passwords differ in strength, some are stronger than others. For example, the differ between a dictionary word and one that has been encrypted using letters taken from an English language list (i e numbers) may cost password cracking devices minutes extra time but add little security because they can all easily be cracked with low entropy - this makes them easy targets for automated attacks. The following examples illustrate various ways people might create poor quality passwords based on some patterns resulting in extremely high degree spaces which make these easy picks when testing automatically at fast speeds

- People use easily guessed words or phrases as passwords
- Dictionary words: chameleon, RedSox, sandbags, bunnyhop!, IntenseCrabtree, etc., including words in non-English dictionaries.
- People use easily accessible personal information as passwords
- People using passwords in multiple accounts
- People create passwords easy to remember, but are also easy to hack
- Common sequences like qwerty, 123456, asdfgh, etc.
- People don't change their passwords often enough
- People don't use strong enough passwords
- People share their passwords with others
- People write down their passwords
- People save their passwords in their web browsers
- People use software that records their keystrokes
- Numeric sequences such as 911 (9-1-1, 9/11), etc.

Characteristics of a strong password:

A strong password is always difficult to guess and includes a variety of characters. Characteristics of a strong password:

- Make sure the password is more than 8 characters long. If a password is longer, it difficult to guess.
- Include both uppercase and lowercase letters as well as numbers and special characters.
- Take a word or phrase and then change the letters with numbers and special characters. For example, “in the dog house” could become !nTh3dawgHs.
- Create a new password by using your name and your pet’s birth code. For example, “Main Street Elementary” and “12/96” could become m1A2/i9n6.

Based on the following circumstances, the datasets we have used here are well balanced and maintain all the criteria that can teach our model to predict a password strength around 98.09 percent accurately in GBA. in other algorithms, it works near around 90 to 92. percent accuracy. The dataset we used from Kaggle is having around 700k passwords and the dataset we’ve created is having around 1500k passwords. This big dataset helps our model to be trained as an almost perfect password predictor.

A strong password is always hard to remember, but an easy password is easy to remember—a password that has to be written down is not strong, no matter how many of the above characteristics are employed. A password that has too many of the above characteristics might not be very secure- no matter how much time or effort is put into choosing one with all possible features! That said Muhlenberg systems can help support strong passwords based on these requirements; however, there may still come certain obstacles outside our control such as a case limit (or character count) which would prevent us from being able to use special characters like spaces, either way, OIT recommends incorporating additional

2.5 Challenges

The key challenge is to collect the necessary data in an appropriate form. Data cleaning is needed to be performed on those collected data. The model will work more accurately if the amount of data is increased. The dataset I used from Kaggle is having around 700k

passwords. This big dataset helps my model to be trained as an almost perfect password predictor.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Research Subject and Instrumentation

TF-IDF technique in NLP is used to determine the importance of a word in a document. It is a statistical measure that is proportional to the number of times a word appears in a document, but it is offset by the frequency of the word in the corpus. This weighting scheme is often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. One of the simplest ranking functions is computed by summing the TF-IDF for each query term. Many more sophisticated ranking functions are variants of this simple model. TF-IDF can be successfully used for stop-words filtering in various subject fields including text summarization and classification.In this model, I use a specific technique of NLP called TF-IDF(Term Frequency and Inverse Document Frequency). By converting text into character by measuring how often or a few times it has been used within our collection/corpus, we can take advantage of both high occurrence rates for certain words or phrases as well as low ones that may not be so common but still exist. Vectorizing this data and separating out characters based on their frequency gives us a better sense of what is important when ranking documents based on user queries because TF-IDF weightings are often used in search engine optimization (SEO) tools like Google'skeyword planners where they score relevance differently than simply counting number appearances

3.2 Data Collection Procedure

To work with this model, I got a renowned dataset from the internet containing around 700k passwords. This is a well-balanced dataset that teaches the model much better and more accurate way.

3.3 Statistical Analysis

If we look closely to the dataset, around 70 percent of the data are mid strength and rest of the data are almost equally weak and stong respectively. For this particular dataset, the system is performing with a high accuracy.

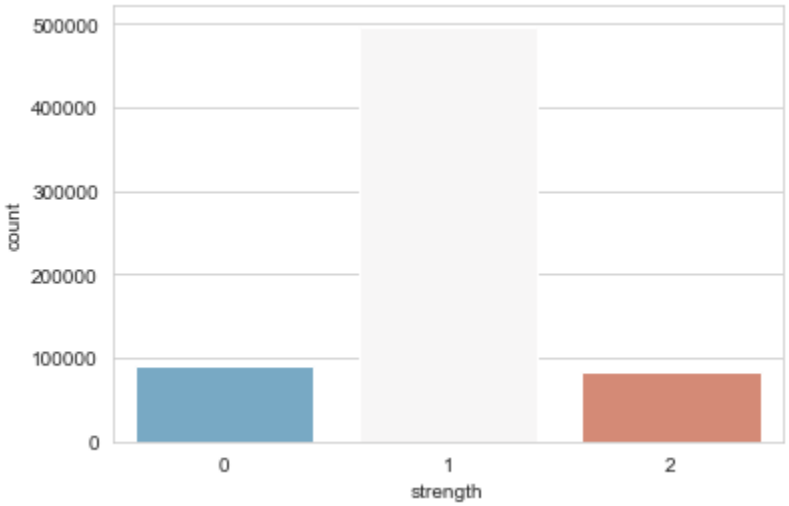

Figure 3.1:Dataset overview

Data pre-processing:

For calculating the password strength from a given password, we need to clean the dataset in order to get an appropriate result. More cleaning can be done according to the requirement. Dropping those inappropriate and incomplete data is one of the vital points.

	password	strength
0	kzde5577	1
1	kino3434	1
2	visi7k1yr	1
3	megzy123	1
4	lamborghin1	1

Figure 3.2: Data set

Creating a password tuple by converting the data.

Then vectorized the given data by separating those characters into 125 features in order to calculate their TF-IDF value of them

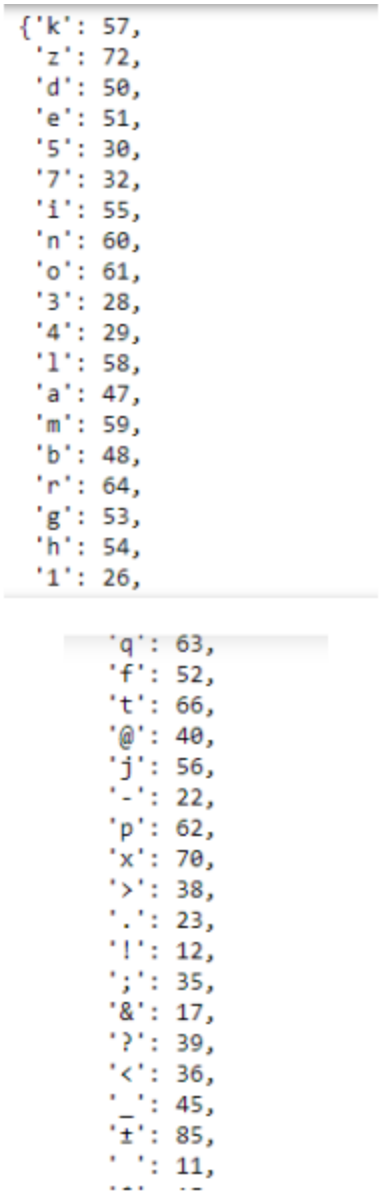

Figure 3.3: Data features

3.4 Proposed Methodology:

In my work, I use TF-IDF vectorizer techniques. Multinomial, XGBOOST algorithm is used in this model. using XGBOOST is a good gradient boosting technique that boosts speed tree-based machine learning.

Combining these techniques, My model is able to provide around 98.5 percent accuracy which is impressive compared to the result of other models. More or less the accuracy of predicting the strength is always near around 97-99 percent which denotes the stability of this model.

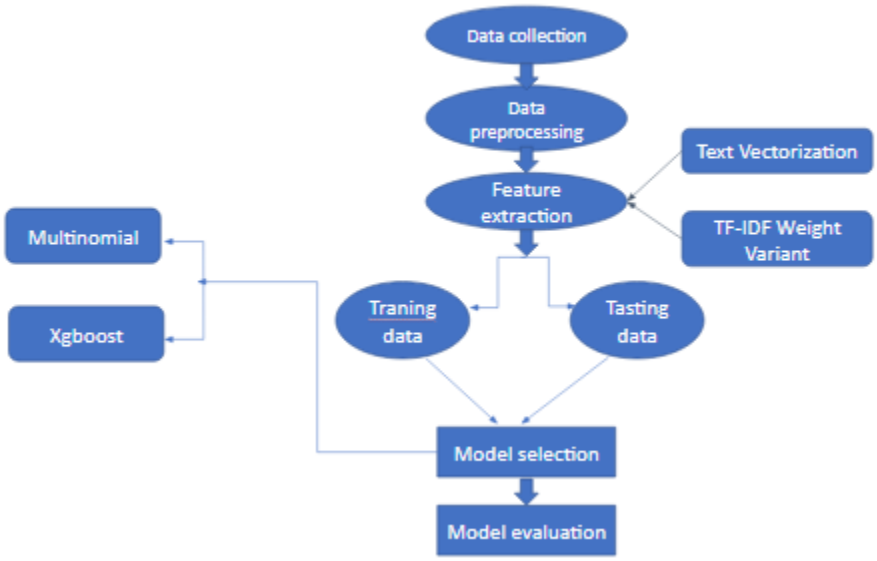

Figure 3.4:Proposed model

Feature extraction:

Word Embedding is one such technique where we can represent the text using vectors. TF-IDF is one of the most used word embedding systems for a long time. When I intend to implement such vectorization for word/text, I keep in mind that.

- It should not result in a sparse matrix since sparse matrices result in high computation costs.
- I should be able to retain most of the linguistic information present in the word.

Introducing eXtreme Gradient Boosting with the model:

The XGBoost algorithm, created by Tianqi Chen and now contributors to the Distributed Machine Learning Community (DMLC), is a machine learning implementation of gradient boosting machines. This tool belongs in with other popular libraries like mxnet deep-learning library that are also developed through this community group known as DMLC or simply "the MLK." The creators also happen to create mxnet deep-learning

library widely implemented across different platforms such as CPUs & GPUs

XGBoost Execution Speed:

XGBoost is super-fast. When compared to other implementations of gradient boosting, it's the fastest!

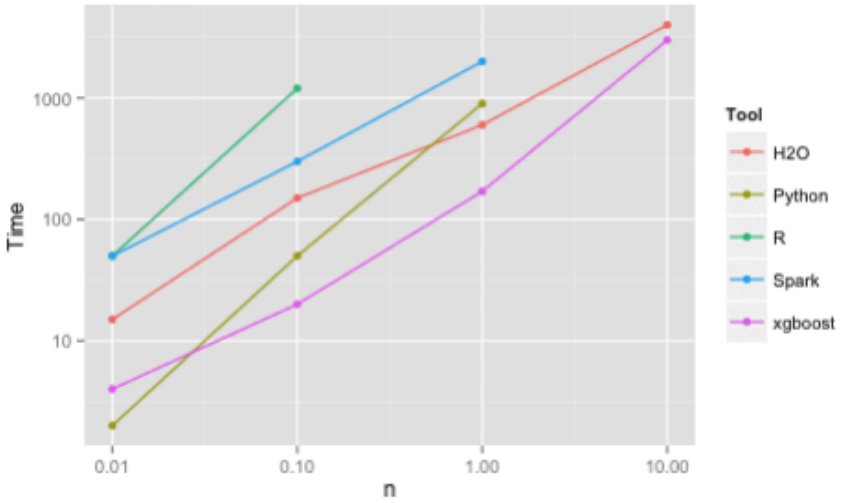

Figure 3.5: Benchmark Performance of XGBoost taken from Benchmarking Random Forest

3.5 Implementation Requirements

- Python
- Tensorflow
- Keras
- Pandas
- Seaborn

Software environment: Windows 10 64-bit operating system, Anaconda Navigator, Ten-TensorFlow 2.0, Python 3.8.2 64-bit. Hardware environment: Intel (R) Core (TM) i5-6500 CPU@3.20GHz processor, 8.00 GB memory, 2 TB mechanical hard disk. Implementation: This whole project was implemented using Google Collaboratory.

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Experimental Setup

This chapter provides implementational details of the designed experiments described in chapter 3.

First, the accuracy rate with the proposed model will be discussed. Then a comparison between different models and algorithms will be shown with the proposed model with descriptive analysis. Finally, the summarized result will be given with a discussion.

4.2 Experimental Results & Analysis

The aim of this project was to develop a system that could predict the strength of a password based on its natural language. Specifically, the system used TF-IDF to measure the importance of words in a password.

A dataset of passwords was used to train and test the system. The results of the study showed that the TF-IDF algorithm was able to predict the strength of a password with an accuracy of over 98 percent.

The results of this experiment showed that my system was able to accurately predict the strength of a password based on its natural language. This suggests that NLP specially TF-IDF could be a reliable method for predicting password strength.

4.3 Compare results between different models

Password Meter:

The website mentions that since no official weighting system exists, they created their own formulas to assess the overall strength of a given password. The application is neither perfect nor foolproof but it should help you decide whether your current login

credentials make for an easy target in today's world where hackers are more sophisticated than ever before.

How Secure is my password:

This website will tell you how long it would take a computer to crack your password based on the type that was entered. This is based off of just the brute force attack and all that's being done in this case, as seen by calculating possible characters equal 10 million divided per second- which can be considered quite high when compared with other estimation tools out there! For example', 'password123' has been found among some common passwords but our probabilistic algorithm doesn't think so since they calculate 16 years worth before coming up empty handed after only guessing 1/4th

Geekwisdom:

The password strength meter is a good way for people who don't know how to create strong passwords or make theircurrent ones more secure. However it's not perfect either--in fact, I'm sure there are plenty of ways it could be improved!

Proposed Model:

The model I am developing uses a technique called TF-IDF. This is one of the simplest ranking functions, but it's also used in more sophisticated models that can predict whether or not something will have an effect on another variable without any error margin included (for example -98% accuracy).

My research showed how combining both artificial intelligence techniques together provides reliable predictions about future impacts due specifically from warming temperatures caused by greenhouse gases emissions into planet earth's atmosphere More generally speaking though.

Validation Accuracy:

From the graph in the figure below, one can see that compared to the first training phase with a base dataset where accuracy was around 92%, it increased significantly after

augmented data had been used for further iterations. This means there were more available input examples and thus a better chance of finding a good output match for any given query; this also led us towards reaching a steady state average of 98%.

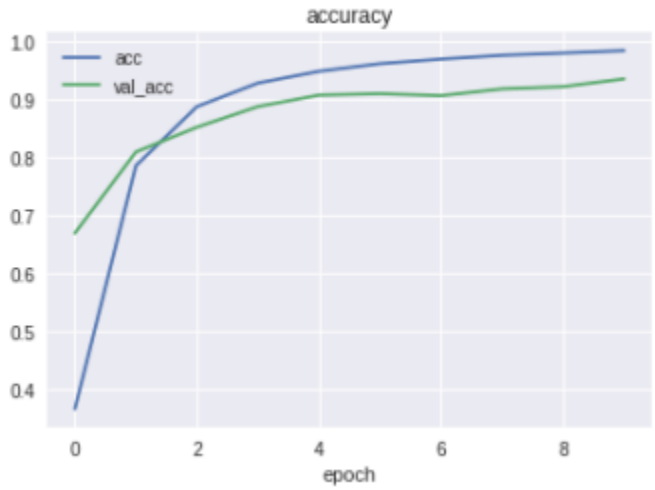

Figure 4.1 validation accuracy

Validation Loss:

Validation Losses With Augmented Data The validation loss with augmented data, is lower than the base dataset.

The curve goes downwards rapidly in earlier epochs and becomes almost stable at around 0.2 which is better than without any augmentation (which achieved a slightly higher value of about).

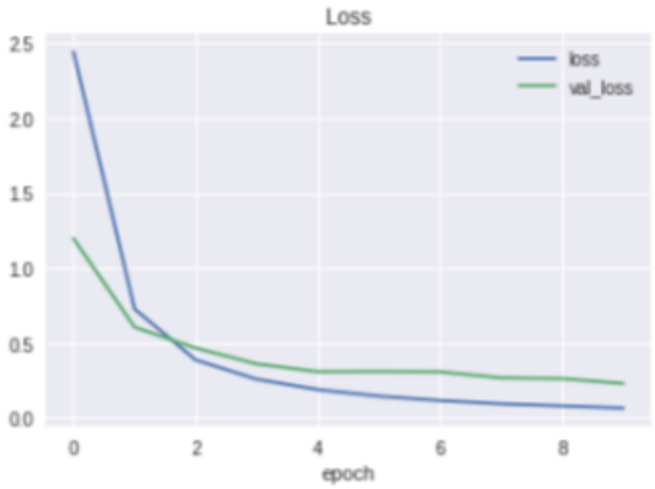

Figure 4.2 validation Loss

4.4 Discussion

XGBoost is an algorithm that has been specifically designed to make the best use of available resources. This means it will be more efficient than other algorithms when applied on vectorized data, which increases its power and predictability as well!

The implementation was engineered with efficiency in mind so we can trust the predictions.

```
In [33]: xgb_classifier=xgb.XGBClassifier()

In [34]: xgb_classifier.fit(X_train,y_train)

C:\Users\User\anaconda3\envs\tf\lib\site-packages\xgboost\sklearn.py:888: UserWarning: The use of label encoder in XGBClassifie
r is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_label_enc
oder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2,
..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

[00:56:06] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.
0, the default evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to 'mlogloss'. Explicitly s
et eval_metric if you'd like to restore the old behavior.

Out[34]: XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
                       colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
                       importance_type='gain', interaction_constraints='',
                       learning_rate=0.300000012, max_delta_step=0, max_depth=6,
                       min_child_weight=1, missing=nan, monotone_constraints='()',
                       n_estimators=100, n_jobs=4, num_parallel_tree=1,
                       objective='multi:softprob', random_state=0, reg_alpha=0,
                       reg_lambda=1, scale_pos_weight=None, subsample=1,
                       tree_method='exact', validate_parameters=1, verbosity=None)

In [35]: xgb_classifier.score(X_test,y_test)

Out[35]: 0.9865972761483782
```

Figure 4.3 Password Strength

CHAPTER 5

IMPACT ON SOCIETY, ENVIRONMENT, AND SUSTAINABILITY

5.1 Impact on Society

This system will help to create stronger passwords and protect people's information while they are accessing the internet. So it will make it harder for hackers to break into websites and steal information. The system will also make it easier for people to create strong passwords, which could help improve security overall. It could help to protect our valuable information from being stolen or compromised. In addition, this project could help raise awareness about the importance of password security and the need to create strong passwords.

5.2 Ethical Aspects

- To prevent our security.
- To protect our privacy.
- To help everyone to check the strength of their password.

5.3 Sustainability Plan

I have a plan to continue to add different types of password samples in order to create a system more accurate. A couple of other features like training people on how they can use the system more comfortably could be added as well.

CHAPTER 6

SUMMARY CONCLUSION, RECOMMENDATION, AND IMPLICATION FOR FUTURE RESEARCH

6.1 Summary of the Study

This chapter provide the concluding remark of this thesis work, the limitations it has, and the future direction of this university project.

6.2 Conclusion

Traditionally, passwords are a string of characters. Symbols are then characters. The strength of a password is determined by three things: the length of the character set used, the length of the password itself, and to a lesser extent, the variety of characters chosen. So it's very difficult to level a password dataset with 100 percent accuracy. This indicates that there will be a bit of a gap between the ultimate expectation and the predicted results. However, with the proposed model, we believe that we can get the best possible result from the model.

6.3 The implication for Future Studies

In this study, I was focused on both the model and a dataset from the internet. As the amount of data is almost around 700k, it was difficult for me to validate, filter, and level all the data with 100 percent accuracy. However, policies for generating strong passwords are changing with respect to the computational capability of the machine. So, adding more computational features, applying several other data filtering parameters, and adjusting with new password-creating policies should be continued in the future in order to keep the model fit.

REFERENCES

[1] K. Sparck Jones. "A statistical interpretation of term specificity and its application in retrieval". Journal of Documentation, 28 (1). 1972.

[2] G. Salton and Edward Fox and Wu Harry Wu. "Extended Boolean information retrieval". Communications of the ACM, 26 (11). 1983.

[3] G. Salton and M. J. McGill. "Introduction to modern information retrieval". 1983
G. Salton and C. Buckley. "Term-weighting approaches in automatic text retrieval". Information Processing & Management, 24 (5). 1988.

[4] H. Wu and R. Luk and K. Wong and K. Kwok. "Interpreting TF-IDF term weights as making relevance decisions". ACM Transactions on Information Systems, 26 (3). 2008.

[5] M. Weir, Sudhir Aggarwal, Breno de Medeiros, Bill Glodek, "Password Cracking Using Probabilistic Context-Free Grammars," Proceedings of the 30th IEEE Symposium on Security and Privacy, May 2009.

[6] R. Morris and K. Thompson. "Password security: a case history" Communications. ACM, 22(11):594–597, 1979.

[7] M. Weir, Using Probabilistic Techniques to aid in Password Cracking Attacks, Dissertation, Florida State University, 2010

[8] T. Booth and R. Thompson, "Applying Probability Measures to Abstract Languages," IEEE Transactions on Computers, Vol. C-22, No. 5, May 1973

[9] Yan, J.J., Blackwell, A., Anderson, R. and Grant, A., "The Memorability and Security of
[10] Passwords -- Some Empirical Results", Technical Report No. 500 (September 2000) Computer Laboratory, University of Cambridge.

[11] Kuo, C., Romanosky, S., and Cranor, L. F., Human Selection of Mnemonic Phrase-based Passwords, Symp. on Usable Privacy and Security (SOUPS), 2006.

[12] Charoen, D., Raman, M., & Olfman, L. (2008). Improving End-User Behaviour in Password Utilization: An Action Research Initiative. Systemic Practice and Action Research, 21(1), 55. Retrieved January 6, 2010.

[13] Monrose, F., Reiter, M., and Wetzel, S. Password hardening based on keystroke dynamics. ACM Conference on Computer and Communications Security, CCS 1999.

[14] U. Manber. A simple scheme to make passwords based on one-way functions much harder to crack. Computers & Security, 15(2):171– 176, 1996.

[15] J. Yan. A Note on Proactive Password Checking. ACM New Security Paradigms Workshop, New Mexico, USA, 2001.Shannon Riley. Password Security: What Users Know and What They Actually

Do.UsabilityNews, 8(1), 2006.

[16] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: analysis of a botnet takeover,"Tech.Rep., April 2009.

[17] R. Shay, S. Komanduri, P. G. Kelley, P. G. Leon, M. L. Mazurek, L. Bauer, N. Christin, and L. F. Cranor. Encountering stronger password requirements: User attitudes and behaviors. In 6th Symposium on Usable Privacy and Security, July 2010.

[18] Y. Zhang, F. Monrose, and M. K. Reiter. The security of modern password expiration: An algorithmic framework and empirical analysis. In Proc. ACM CCS'10, 2010.

[19] A. Adams and M. A. Sasse. Users are not the enemy. Comm ACM, 42(12):40–46,December 1999.

[20] IEEE Explore, available at << https://ieeexplore.ieee.org/document/5376606/figures#figures >> last accessed on 17th August, 2022 at 9.30 am

[21] RIT Cyber Security Class Blog, available at << https://ritcyberselfdefense.wordpress.com/2011/09/24/how-to-calculate-password-entropy/ >> last accessed on 15th August, 2022 at 2.30 am

[22] Muhlenberg College, available at <<https://www.muhlenberg.edu/offices/oit/about/policies_procedures/strong-passwords.html >> last accessed on 15th August, 2022 at 2.40 am