Predicting The Strength of Password Using ML

RAJESH AREPALLI[1], G. SUPRIYA[2], SK. SHEEMA[3], A.L.S. DEVI[4], D. GOVARDHAN[5], G. SUBBARAO[6]

[1] *Sr. Asst Prof., CSE Dept, Sri Vasavi Engineering College, Tadepalligudem.*

[2, 3, 4, 5, 6] *Student, CSE Dept, Sri Vasavi Engineering College, Tadepalligudem, A.P, India*

Abstract— Now-a-days the technology was immensely expanding, so the secure authentication methods become more prominent. Although there are many alternatives to passwords for accessing control, but password is a more convenient way to authenticate the identity. Passwords are used for numerous functions in daily life like accessing programs, networks, websites etc. Passwords serve as a primary means of user authentication method but they are vulnerable to various attacks due to the prevalence of weak and easily guessable passwords. Every day, the need of choosing and utilising strong passwords was increasing. The goal of this project is to develop an accurate system which is capable of determining the strength of a password based on its linguistic properties. We are going to build a model to predict the password strength using machine learning Algorithm and Natural language processing in python. The outcome of this project can contribute significantly to the field of password security. By accurately predicting the password strength using NLP techniques we can assist users in creating stronger passwords and enhances overall cybersecurity.

Index Terms- Machine Learning, Algorithm, Python, Natural Language Processing

I. INTRODUCTION

Password acts as a primary key for login activities into any websites, networks etc. Password is a combination of characters, numbers and special symbols. The strength of the password depends on the length of its length, special symbols used. Passwords that are long and complex reduce the risk of being cracked, but they do not guarantee safety. A password's strength measures how well it defends against guessing and other kinds of attacks. Cybersecurity begins with the basic technique of password-based authentication in order to help secure information on the internet. In order to create a password strength checker, a machine learning model is trained on labelled datasets of different password combinations. This ml model is loaded into the web application where users can able to check the strength of their passwords.

II. LITERATURE SURVEY

[1] Umar Farooq proposed a paper "Real Time Password Strength Analysis on a Web Application Using Multiple Machine Learning Approaches" from Central university of Punjab, this paper is all about developing ml models and real time analysis of passwords. Multiple programming languages are used in this paper, including HTML5, CSS3, JavaScript, PHP, and Python. The back-end database is connected by using PHP.

[2] Sony Kuriakose, G Krishna Teja, Sravan Dugi, A Hershel Srivastava, Venkat Jonnalagadda these people proposed a paper "Machine Learning Based Password Strength Analysis" in this paper model was explained by using UML diagrams and Data Flow Diagrams

[3] Gong Zhu Hu proposed a paper "On Password Strength: A Survey and Analysis" in this paper Brute-Force, Dictionary, Table Lookup, and Rainbow Table Attacks are used in ML models for password strength prediction to generate data, extract features, label data, simulate real-world scenarios, evaluate model performance, and enhance password security.

III. METHODOLOGY

The main goal of this model is that we are developing is used to analyse the strength of real time passwords given by the users. This is developed by using numerous machine learning algorithms. Machine Learning model development involves different phases: -

1. DATASET COLLECTION: Model development begins with the collection of a dataset with two attributes - passwords and strengths. A dataset contains n numbers of passwords with different strengths, such as weak, medium, and strong. Strengths represented as 0,1 and 2 for the passwords. Collecting a relevant dataset is the most important factor in analysing the strength of a password using

machine learning which are labelled as 0,1 and 2. Where 0 is weak,1 is medium and 2 is for password. Taking the huge amount of dataset contains all types of passwords which has 3 categories and each one is created with different conditions are in text format in initial stage and using several pre-processing methods for converting them into csv file for usage.
https://www.kaggle.com/datasets/bhavikbb/password-strength-classifier-dataset?select=data.csv

- Password Strength Analysis:

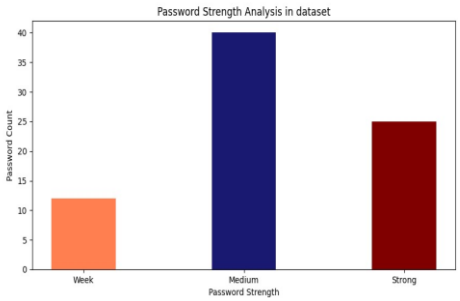

Fig 1

2. DATA PREPROCESSING: In next phase the focus is on the preprocessing methods and extracting the required features form the dataset using Tfidf vectorizer.

In our dataset, which contains passwords and their corresponding strengths, it's possible that we encounter missing values and inconsistent data. These issues need to be addressed using various techniques, collectively known as data preprocessing.
data isna (). sum ()
data. isna (): - This part generates a Boolean data frame. If there is a missing value in the dataset then it returns true otherwise false.
. sum () is applied to the data frame to calculate them number of true values.

In cases where there are a few missing values, we opt to remove the corresponding rows from the dataset. However, if the missing values are more prevalent, we employ an imputation strategy. Specifically, for the 'strengths' column, we fill the missing values with the mean values derived from the entire column. Conversely, for the 'passwords' column, we utilize the mode values extracted from the entire column to address missing values.

Steps for Machine Learning Model Training:

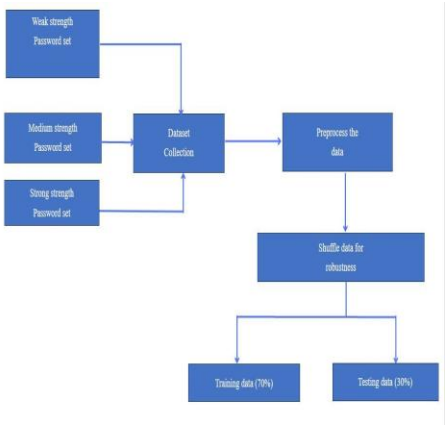

Fig 2

3.TOKENIZATION
Tokenization is the process of breaking down of text (passwords) into individual units.

```
def word_divide_char(inputs):
    character= []
    for i in inputs:
        character. append(i)
    return character
```

here word_divide_char function is tokenizing the passwords into individual characters. Breaking down of passwords can help to extract the features.

4.FEATURE EXTRACTION
Count Vectorizer and TfidfVectorizer are methods of scikit-learn library in python. These two methods are used to extract the features from the text data given by the users. These Vectorizers converts the text data into numerical format by which machine leaning algorithms work with.

TF-IDF is a text Vectorization technique, mostly used in natural language processing. It is also used in data analysis. It is used to transform the passwords into numerical feature vectors. These feature vectors represents the importance of individual characters .word_divide_char function is passed to the tokenizer as an argument which defines the breaking down of text data into individual elements. A variable vectorizer is initiated to store the instance of TfidfVectorizer. Later this vectorizer is used to transform the passwords into TF-IDF feature vectors.

```
vectorizer=
TfidfVectorizer(tokenizer=word_divide_char)
```

All the passwords are passed are converted into numerical format using the fit_transform method.
X=vectorizer.fit_transform(x)
here x is the list of passwords. X is the variable which is used to store the transformed data in a matrix format.

5.TRAINING AND TESTING: The dataset is divided into two parts .70% part is used for training the model (Training part) and another 30% parts is used for testing the model (Testing part). Testing dataset is used to evaluate the proposed model. Scikit-learn is a popular machine learning algorithm. train_test_split is the function provided by scikit-learn algorithm which is used to divide the data into training and testing sets.

IV. LOGISTIC REGRESSION

Logistic Regression is a supervised machine learning algorithm. It is commonly used for classification tasks.
clf=LogisticRegression (random_state=0, multi_class='multinomial')
clf is a variable to store the logistic regression classifier which uses the machine learning model. There are two parameters passed to Logistic Regression:
1)random state: It is used to make the results reproducible that means it is used to generate the same results every time when we run the code.
2)multiclass: It specifies the methos used to handle multiple classes. Here there are three classes - weak, medium, strong.
Now the model will be trained using the fit method ,Here the model will learn to predict the password strength from training data based on the provided features.
Accuracy and Prediction:

```
print("Accuracy :",clf.score(X_test, y_test))
```

```
Accuracy : 0.8192536288154829
```

```
X_predict = ['password@1123']

X_predict = vectorizer.transform(X_predict)
y_Predict = clf.predict(X_predict)
print(y_Predict)
```

```
['strong']
```

Fig 3

6.DEPLOYMENT: The trained model will be deployed into web application using the frameworks like flask, Django etc.The trained model will be loaded into web application using pickle or joblib libraries.

V. PYTHON LIBRARIES

PANDAS: Panda's library is imported for data manipulation and analysis.
NUMPY: NumPy is called as numerical python. It is a fundamental library for scientific computing with Python. In this code NumPy is imported to work with numerical data, arrays and mathematical functions
RANDOM: Random module is imported for generating the randomness. Random module is used to shuffle the dataset to increase the robustness.
MATPLOTLIB: Pipilotti's library is imported for data visualization. A bar chart can be drawn to visualize the distribution of password strength.

VI. SYSTEM ARCHITECTURE

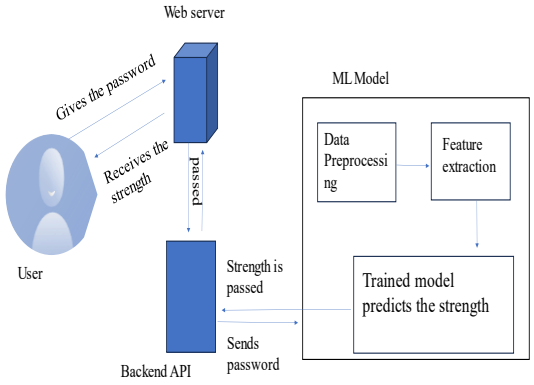

Fig 4

This is the system architecture for the project. Here user will give the password through user interface. It was received by web server; it passes the password to the backend API. It passes to the model where data pre-processing, feature extraction takes place. The trained model predicts the strength and passes the strength to the API, it again passes to the webserver. Finally, the strength can be displayed to the user through UI.

VII. USER INTERFACE

user interface will be developed using HTML5 and CSS3.Here user can enter the real time password to check its strength. Passwords are passed to the trained model, then we predict the strength and displays the output to the user on interface.

INTERFACE:

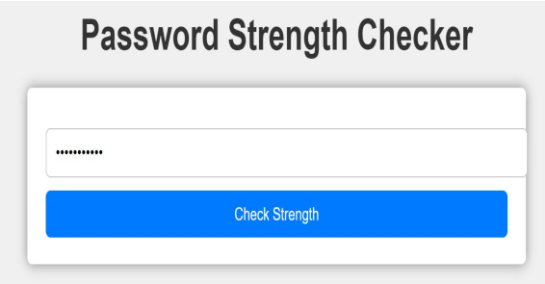

Fig 5

A. TESTING

On the report of our tests, the proposed system done by us has the ability of detecting and analysing the strength of a password.

B. REAL TIME ANALYSIS.

Python, HTML5, Flask framework and CSS3 are the multiple languages of computer used to prepare proposed system. We also have languages like HTML5, CSS3 which are used to create webpage including some flask web framework also.

C. RESULT:

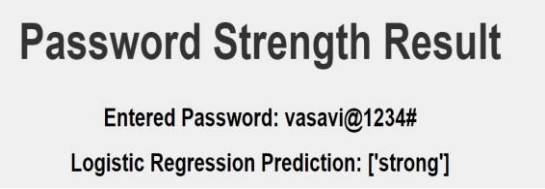

Fig 6

And now if the terms are not satisfied, users can come up with new passwords that are supportable by the model in their terms and conditions. And now it is difficult for hackers to attack on someone's accounts as this increases the complexity and security for their accounts.

CONCLUSION

From this paper we are going to propose that password strength predictor, implemented by using machine learning approaches on a web application. Our main target is to keep everyone's account safe and getting rid of attackers this is possible by a strong password which is not possible to hack as weak is easily cracked by the hackers and at the end of this the user will benefitable and this will help them from cyberattacks.

REFERENCES

[1] Sony Kuriakose, G Krishna Teja, Sravan Dugi, A Hershel Srivastava, Venkat Jonnalagadda published a paper Machine Learning Based Password Strength Analysis under International Journal of Innovative Technology and Exploring Engineering (IJITEE)ISSN: 2278-3075 (Online), Volume-11 Issue-8, July 2022

[2] Darbutaite, E.; Stefanoviˇc, ˙P.; Ramanauskaite, S. ˙Machine-Learning-Based Password-Strength-Estimation Approach for Passwords of Lithuanian Context. Appl.Sci.2023,13, 7811.

[3] Umar Farooq Department of Computer Science & Technology, Central University of Punjab, Bathinda, India published a paper Real Time Password Strength Analysis on a Web Application Using Multiple Machine Learning Approaches under International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 9 Issue 12, December-2020

[4] Bhavani Gorla* GMR Institute of Technology Razam, Kinneravada, Toguri, Saravakota, Srikakulam, Andhra Pradesh, PIN-532427, India published a paper Password Strength Analyzer under International Journal of Research Publication and Reviews ISSN 2582-7421

[5] M. Weir, S. Aggarwal, B. d. Medeiros and B. Glodek, "Password Cracking Using Probabilistic Context-Free Grammars," 2009 30th IEEE Symposium on Security and Privacy, Berkeley, CA, 2009, pp. 391-405, Doi: 10.1109/SP.2009.8.

[6] Grassi P.A., Garcia M., Fenton J. NIST Special Publication 800–63–3 Digital Identity Guidelines National Institute of Standards and Technology, Los Altos, CA (2020)

[7] Zhou Huan, Liu Qixu, Cui Xiang, Zhang Fang jiao. Research on Targeted Password Guessing Using Neural Networks. Journal of Cyber Security. 2018. 3(5): p. 25-37

[8] 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

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