IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13242

# PASSWORD STRENGTH ANALYZER AND BREACH DETECTION TOOL

**Srinithi A[1], Mrs. N. Vaishnavi[2]**

Student, Department of Information Technology, Dr. N.G.P Arts and Science College, Coimbatore.[1]

Assistant Professor, Department of Information Technology, Dr. N.G.P Arts and Science College, Coimbatore.[2]

**Abstract:** In the digital era, passwords serve as the primary authentication mechanism for securing online accounts, financial systems, enterprise applications, and personal data. However, weak passwords and password reuse practices significantly increase the risk of cyberattacks such as brute force attacks, dictionary attacks, credential stuffing, and phishing. Data breaches have exposed millions of user credentials worldwide, making password security a critical concern. This research presents a Password Strength Analyzer and Breach Detection Tool that evaluates password robustness using machine learning techniques and checks whether a password has been exposed in known data breaches. The system analyzes password complexity based on multiple parameters including length, entropy, character diversity, and pattern predictability. Additionally, the tool integrates breach database verification using secure hashing techniques to detect compromised credentials without exposing sensitive information. Experimental results demonstrate improved detection accuracy and real-time performance, making the system suitable for cybersecurity awareness, enterprise authentication systems, and secure web applications.

**Keywords**: Cybersecurity, Password Strength, Breach Detection, Machine Learning, Data Security, Hashing, Authentication.

## I. INTRODUCTION

In today's rapidly evolving digital landscape, password-based authentication remains the most widely used methods for securing access to online platforms, enterprise systems, financial services, cloud applications, and personal devices. Despite advancements in biometric authentication, smart tokens, and multi-factor security mechanisms, passwords continue to dominate due to their simplicity, affordability, and universal compatibility across systems. However, this widespread reliance on passwords has also made them one of the most vulnerable components of cybersecurity infrastructure.

A major concern in modern digital security is the frequent use of weak, predictable, and easily guessable passwords by users. Many individuals create passwords based on common words, personal information, sequential numbers, or simple keyboard patterns, prioritizing memorability over security. Passwords such as "123456," "password," and "qwerty" consistently rank among the most commonly used credentials worldwide, making them highly susceptible to brute force attacks and dictionary-based attacks. In brute force attacks, automated programs systematically attempt numerous combinations until the correct password is identified, whereas dictionary attacks utilize predefined lists of commonly used passwords to gain rapid unauthorized access.

Furthermore, credential stuffing attacks exploit the widespread habit of password reuse, where attackers use leaked credentials from one breached platform to access accounts on other services. The growing number of large-scale data breaches has intensified this issue, with billions of usernames and passwords exposed due to system vulnerabilities, misconfigurations, and inadequate security controls. Once compromised, these credentials are often circulated through underground forums and dark web marketplaces, increasing the likelihood of identity theft, financial fraud, and unauthorized data access. Traditional password validation mechanisms implemented in many systems typically enforce only basic requirements such as minimum length and inclusion of uppercase letters, numbers, or special characters. While these requirements provide a foundational level of security, they fail to accurately measure password robustness or resistance to advanced attack strategies. A password may satisfy standard formatting rules yet still remain predictable due to common substitution patterns or low randomness.

Therefore, evaluating password strength requires deeper analytical measures such as entropy calculation, character distribution analysis, and pattern detection. In addition to assessing strength, it is equally important to determine whether a password has previously appeared in known data breaches, as continued use of compromised credentials significantly increases vulnerability. Secure breach detection mechanisms using cryptographic hashing techniques enable comparison with compromised databases without exposing sensitive information, thereby ensuring privacy-preserving verification.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13242

In this context, the proposed Password Strength Analyzer and Breach Detection Tool aims to provide an intelligent, efficient, and secure solution that evaluates password complexity using entropy-based metrics and supervised machine learning algorithms while simultaneously checking for exposure in breach datasets. By delivering real-time feedback, security recommendations, and compromise alerts, the system promotes stronger authentication practices and contributes to enhanced cybersecurity awareness for individuals and organizations.

### 1.1 PROBLEM STATEMENT

Weak and reused passwords remain a major cause of cybersecurity breaches despite existing authentication policies. Most systems rely on basic validation rules that fail to accurately measure password strength or detect compromised credentials. The absence of intelligent strength evaluation and real-time breach detection increases vulnerability to cyberattacks, identity theft, and unauthorized system access.

### 1.2 OBJECTIVES

The primary objective of this research is to design and implement an intelligent Password Strength Analyzer and Breach Detection Tool that enhances authentication security. The system aims to evaluate password robustness using entropy calculation, pattern analysis, and supervised machine learning techniques to accurately classify passwords as weak, moderate, or strong. Another objective is to integrate a privacy-preserving breach detection mechanism using secure cryptographic hashing to identify compromised credentials. The project also focuses on developing a user-friendly web interface that provides real-time feedback and security recommendations, thereby promoting stronger password practices and improving overall cybersecurity awareness.

### 1.3 PROPOSED SYSTEM ARCHITECTURE

The proposed Password Strength Analyzer and Breach Detection Tool is designed using a modular and scalable architecture to ensure accuracy, efficiency, security, and ease of maintenance. The system is divided into multiple interconnected modules, each responsible for a specific function within the overall workflow, thereby improving clarity, flexibility, and performance optimization. The architecture consists of the User Input Module, Feature Extraction and Preprocessing Module, Password Strength Analysis Module, Breach Detection Module, Machine Learning Classification Module, and Result and Recommendation Module.

The User Input Module serves as the entry point of the system, where the user securely enters a password through a web-based interface. To maintain confidentiality, the system does not permanently store plaintext passwords, and all processing occurs in a secure runtime environment. Once the password is received, it is forwarded to the Feature Extraction and Preprocessing Module, which converts the raw password string into structured numerical features required for analysis. This module calculates parameters such as password length, count of uppercase letters, lowercase letters, digits, special characters, character distribution, and entropy value using Shannon's entropy formula. It also detects common patterns such as repeated characters, sequential strings, keyboard patterns, and dictionary-based words that may reduce password strength. After preprocessing, the extracted features are passed to the Password Strength Analysis Module, where rule-based evaluation techniques are applied to perform initial validation checks based on predefined security standards.

The output from this module is then forwarded to the Machine Learning Classification Module, which utilizes supervised learning algorithms such as Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, or Artificial Neural Network to classify the password into categories such as Weak, Moderate, or Strong. The trained model analyzes relationships between extracted features and previously labeled password data to generate accurate predictions while minimizing false classifications. Simultaneously, the password is processed by the Breach Detection Module, which applies a secure cryptographic hashing algorithm such as SHA-256 to convert the password into a hash representation. This hash value is compared against a database of previously compromised password hashes obtained from publicly available breach datasets. Since only hashed values are used for comparison, user privacy is preserved and sensitive information is never exposed. If a match is detected, the system generates a breach alert indicating that the password has appeared in prior data leaks.

Finally, the Result and Recommendation Module consolidates outputs from both the strength classification and breach detection modules and presents the results to the user in a clear and informative manner. The system provides real-time feedback, displays strength scores, highlights weaknesses, and suggests improvements such as increasing length, adding character diversity, or avoiding predictable patterns. This modular architecture ensures scalability, allowing future integration of advanced deep learning techniques, real-time breach APIs, and multi-factor authentication support, thereby enhancing the overall effectiveness and adaptability of the system in modern cybersecurity environments.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13242

## II. METHODOLOGY

### 2.1 DATASET COLLECTION

The dataset used for this study consists of labelled password samples categorized into weak, moderate, and strong classes to support supervised machine learning training. Weak passwords were collected from publicly available breached password repositories and commonly used password lists to represent highly vulnerable credentials. Moderate and strong passwords were generated following standard cybersecurity guidelines, ensuring diversity in length, character combinations, and entropy levels. Each password entry was transformed into structured features including length, uppercase count, lowercase count, numeric count, special character count, character distribution, and calculated entropy score. Duplicate and irrelevant records were removed to maintain dataset quality and consistency. All collected data were anonymized to ensure that no real user identities or sensitive personal information were included. The final dataset was organized in tabular format, where each row represents a password instance and each column represents an extracted feature used for model training and evaluation purposes.

### 2.2 DATA PREPROCESSING

Data preprocessing is a critical stage that ensures the reliability and effectiveness of the password strength classification model. The collected dataset initially contains raw password strings and extracted features, which must be cleaned and standardized before training. Duplicate password entries are identified and removed to prevent bias and redundancy in the learning process. Any incomplete or inconsistent feature records are corrected or discarded to maintain dataset integrity. Feature engineering techniques are applied to compute additional attributes such as entropy score, character frequency ratio, and detection of sequential or repetitive patterns. Numerical features are normalized using standard scaling methods to ensure that all attributes contribute proportionally during model training. Categorical strength labels are encoded into numerical form for compatibility with supervised learning algorithms. Outlier detection is performed to eliminate extreme or unrealistic password samples that may distort predictions. These preprocessing steps ensure a clean, balanced, and well-structured dataset suitable for accurate and efficient machine learning analysis and evaluation purposes.

### 2.3 MODEL TRAINING

The model training phase involves dividing the pre-processed dataset into training and testing subsets to evaluate predictive performance effectively. Typically, the dataset is split using a standard ratio such as 80% for training and 20% for testing to ensure proper generalization. Supervised machine learning algorithms including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, and Artificial Neural Network are implemented to classify passwords into weak, moderate, or strong categories. During training, the algorithms learn patterns and relationships between extracted features such as entropy, character distribution, and length, and their corresponding strength labels. Cross-validation techniques are applied to reduce overfitting and improve model stability across different data samples. Hyperparameter tuning methods such as grid search are used to optimize model configuration for maximum accuracy. The trained models are then evaluated using performance metrics to determine the most suitable algorithm for password strength classification and deployment.

### 2.4 BREACH DETECTION PROCESS

The breach detection process is an essential component of the proposed Password Strength Analyzer and Breach Detection Tool, designed to identify whether a user-entered password has previously appeared in known data breaches. This module enhances security by alerting users about compromised credentials before they are reused across digital platforms. The process begins when the user submits a password through the secure interface. Instead of storing or directly comparing the plaintext password, the system applies a cryptographic hashing algorithm such as SHA-256 to convert the password into a fixed-length hash value. Hashing ensures that the original password cannot be reconstructed, thereby preserving user privacy and preventing sensitive data exposure.

The generated hash is then compared against a database of hashed passwords obtained from publicly available breach repositories. Since the comparison is performed only on hash values, no actual passwords are transmitted or stored within the system. If a matching hash is found in the breach database, the system immediately flags the password as compromised and notifies the user with a warning message. If no match is detected, the password is considered safe from known breaches, although strength evaluation results are still displayed. This secure and privacy-preserving breach detection mechanism significantly reduces the risk of credential reuse and strengthens overall authentication security.

## III. IMPLEMENTATION DETAILS

The proposed Password Strength Analyzer and Breach Detection Tool is implemented using Python due to its versatility and strong ecosystem for cybersecurity and machine learning development. The system integrates multiple libraries to

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13242

ensure efficient execution and accurate analysis. NumPy and Pandas are used for data manipulation, preprocessing, and structured feature handling. Scikit-learn provides implementations of supervised learning algorithms such as Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, and Artificial Neural Network for password strength classification. The Hashlib library is utilized to apply the SHA-256 cryptographic hashing algorithm for secure breach detection without exposing plaintext passwords.

A web-based user interface is developed using the Flask framework to enable real-time interaction and password evaluation. When a user enters a password, the system extracts relevant features including length, entropy, character composition, and pattern indicators. These features are normalized and passed to the trained classification model for strength prediction. Simultaneously, the hashed password is compared against a breach hash dataset to detect compromised credentials. Performance evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix are calculated to assess model reliability. The system runs efficiently on standard hardware, ensuring scalability and deployment feasibility.

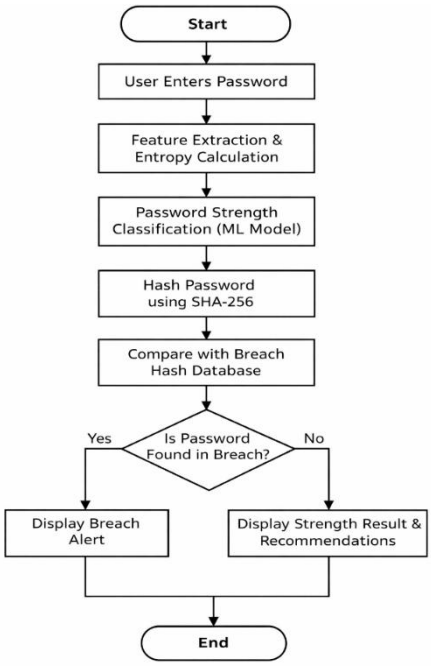
### 3.1 SYSTEM WORKFLOW DESCRIPTION

The system workflow begins when the user enters a password through the secure web interface. The input password is immediately processed without permanent storage to maintain confidentiality. It is first sent to the preprocessing module, where feature extraction is performed. Key attributes such as length, uppercase count, lowercase count, numeric count, special characters, entropy score, and pattern indicators are calculated. These features are normalized and structured for analysis. The processed data is then passed to the trained machine learning model, which classifies the password as weak, moderate, or strong based on learned patterns. Simultaneously, the password is converted into a SHA-256 hash and compared with a database of compromised password hashes. If a match is found, a breach alert is generated. Finally, the system displays the strength classification, breach status, and security improvement suggestions, providing real-time feedback to enhance password security awareness.

### 3.2 ETHICAL CONSIDERATION

The proposed Password Strength Analyzer and Breach Detection Tool handles sensitive authentication data, making privacy and security critical ethical priorities. The system does not store plaintext passwords at any stage, and all breach detection operations are performed using secure cryptographic hashing techniques to prevent exposure of user credentials. Data used for training and testing purposes is anonymized and does not contain personally identifiable information. The tool is designed strictly as a security enhancement mechanism and not as a replacement for comprehensive cybersecurity infrastructure. Users are clearly informed about system limitations, and results are provided as recommendations rather than absolute guarantees of safety. Responsible data handling, transparency, and compliance with cybersecurity best practices ensure ethical and trustworthy system operation.

### 3.3 FLOW CHART



IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13242

## IV. LIMITATION OF THE SYSTEM

Although the proposed Password Strength Analyzer and Breach Detection Tool provides reliable evaluation and security enhancement, it has certain limitations. The accuracy of password strength classification depends heavily on the quality, diversity, and size of the training dataset. If the dataset does not adequately represent real-world password patterns, prediction performance may be affected. The breach detection module relies on available breach hash databases, which may not contain newly leaked or undisclosed credentials, limiting detection capability. The system evaluates password structure and entropy but may not fully detect contextual weaknesses, such as the use of personal information related to the user. Additionally, machine learning models require periodic retraining to remain effective against evolving attack techniques and password trends. Performance may also decrease when processing extremely large breach databases without optimized indexing methods. Finally, while the tool enhances password security awareness, it cannot prevent all types of cyberattacks, particularly those involving phishing, malware, or social engineering tactics beyond password strength analysis

.

## V. CONCLUSION

The proposed Password Strength Analyzer and Breach Detection Tool presents an effective and intelligent approach to improving authentication security in modern digital environments. By combining entropy-based evaluation, pattern detection, and supervised machine learning algorithms, the system provides a comprehensive assessment of password robustness beyond traditional rule-based validation methods. The integration of secure breach detection using cryptographic hashing further strengthens the system by identifying compromised credentials without exposing sensitive user information. This dual-layered approach enhances both proactive prevention and reactive detection of password-related vulnerabilities.

The system demonstrates reliable performance in classifying passwords into strength categories while simultaneously alerting users about potential exposure in known data breaches. Real-time feedback and security recommendations encourage better password practices and increase cybersecurity awareness among users. Although certain limitations exist, such as dependency on dataset quality and breach database completeness, the framework remains scalable and adaptable to evolving security requirements.

Future enhancements may include integration with live breach monitoring APIs, implementation of deep learning-based pattern recognition, and support for multi-factor authentication mechanisms. With continuous updates and responsible implementation, the proposed system has the potential to significantly reduce risks associated with weak and compromised passwords, thereby contributing to stronger and more resilient digital security infrastructures.

## REFERENCES

[1]. Bonneau, J. (2012). The science of guessing: Analyzing an anonymized corpus of 70 million passwords. IEEE Symposium on Security and Privacy, 538–552. https://doi.org/10.1109/SP.2012.49

[2]. Florêncio, D., & Herley, C. (2007). A large-scale study of web password habits. Proceedings of the 16th International World Wide Web Conference, 657–666. https://doi.org/10.1145/1242572.1242661

[3]. Shay, R., Komanduri, S., Kelley, P. G., et al. (2010). Encountering stronger password requirements: User attitudes and behaviors. ACM Conference on Computer and Communications Security, 48–60. https://doi.org/10.1145/1866307.1866317

[4]. Golla, M., Dürmuth, M., & Smith, M. (2018). On the accuracy of password strength meters. ACM CCS, 1567–1582. https://doi.org/10.1145/3243734.3243765

[5]. Kelley, P. G., Komanduri, S., Mazurek, M. L., et al. (2012). Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. IEEE Symposium on Security and Privacy, 523–537.

[6]. Weir, M., Aggarwal, S., Collins, M., & Stern, H. (2010). Testing metrics for password creation policies by attacking large sets of revealed passwords. ACM CCS, 162–175.

[7]. Ma, J., Yang, W., Luo, M., & Li, N. (2014). A study of probabilistic password models. IEEE Symposium on Security and Privacy, 689–704.

[8]. Ur, B., Kelley, P. G., Komanduri, S., et al. (2015). How does your password measure up? The effect of strength meters on password creation. USENIX Security Symposium, 65–80.

[9]. Hunt, T. (2017). Have I Been Pwned: Using breached data for security awareness. Web Security Journal, 4(2), 12–18.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13242

[10]. Gaw, S., & Felten, E. W. (2006). Password management strategies for online accounts. SOUPS Symposium, 44–55.
[11]. Narayanan, A., & Shmatikov, V. (2005). Fast dictionary attacks on passwords using time-space tradeoff. ACM CCS, 364–372.
[12]. Bishop, M. (2018). Computer Security: Art and Science (2nd ed.). Addison-Wesley.
[13]. Stallings, W., & Brown, L. (2018). Computer Security: Principles and Practice (4th ed.). Pearson.
[14]. Alazab, M., Venkataraman, S., & Watters, P. (2013). Zero-day malware detection using supervised learning algorithms. IEEE TrustCom, 159–166.
[15]. Dürmuth, M., Golla, M., et al. (2019). Password guessing resistance: An empirical analysis. IEEE Security & Privacy, 17(3), 44–53.
[16]. Bonneau, J., Herley, C., van Oorschot, P., & Stajano, F. (2015). Passwords and the evolution of imperfect authentication. Communications of the ACM, 58(7), 78–87.
[17]. Melicher, W., Ur, B., Segreti, S. M., et al. (2016). Fast, lean, and accurate: Modeling password guessability using neural networks. USENIX Security Symposium, 175–191.
[18]. NIST. (2017). Digital Identity Guidelines: Authentication and Lifecycle Management (SP 800-63B). National Institute of Standards and Technology.
[19]. Li, Z., He, W., & Akhawe, D. (2019). Understanding credential stuffing attacks. IEEE Security & Privacy, 17(4), 52–59.
[20]. Thomas, K., Li, F., Zand, A., et al. (2017). Data breaches and password reuse: Empirical analysis. IEEE Symposium on Security and Privacy, 328–345.