

Password Strength Analyzer Based on AI

Author: Neha Faresh **Programme:** BCA — Final Year Project, AI Foundation Programme

Institution: SJ Global PU & Degree College, in collaboration with Aider Digital Pvt Ltd

Abstract

Passwords remain a widely used authentication mechanism, but human-chosen passwords can be vulnerable to guessing attacks because users often select secrets that are predictable or easy to remember [1], [2]. A password-strength analyzer attempts to estimate how resistant a password is to guessing and provide feedback during password creation [1], [3]. This paper proposes a practical, AI-based password strength analyzer suitable for a one-semester BCA project. The proposed system combines basic security checks with a lightweight machine-learning model trained to recognize patterns associated with password guessability. Unlike a simple rule-based checker that mainly counts length or character categories, the proposed analyzer would examine patterns such as common words, repetitions, sequences, substitutions, and learned password structures, while producing an understandable strength category and feedback. Neural-network research has shown that learned models can be used to model password guessing more accurately than several traditional approaches and can be made compact enough for client-side use [4], [5]. The project is intentionally scoped as a defensive educational prototype rather than a production authentication service. Its limitations include dependence on training data, possible dataset bias, uncertainty in estimating real attacker behavior, and the inability of any password meter to guarantee that a password will never be guessed or compromised [3], [6].

1. Background

A password is a memorized secret used as an authentication factor. Current NIST guidance states that passwords must have sufficient effective strength and secrecy to make guessing or discovery impractical [1]. NIST also recommends checking newly selected passwords against a blocklist containing commonly used, expected, or compromised values, including passwords from previous breach corpora and context-specific words [1]. This reflects an important distinction between password length and actual resistance to guessing: a long password can still be predictable if it follows a common human pattern [1], [3].

Weak passwords remain a security concern because attackers can attempt guesses using dictionaries, previously exposed passwords, patterns, and transformations. Research on password guessability has shown that different guessing methods are systematically better at

guessing some password features than others, and that relying on a single guessing algorithm can give a misleading estimate of security [3]. NIST therefore emphasizes blocklists, secure password storage, rate limiting, and other controls rather than relying only on composition requirements [1].

Traditional composition rules often require combinations of uppercase letters, lowercase letters, digits, and symbols. However, NIST's current guidance recommends against imposing additional composition rules on memorized secrets because users tend to satisfy such rules through predictable transformations [1], [2]. For example, research has documented user beliefs that adding a symbol can make an otherwise predictable password secure, illustrating why the presence of a symbol alone is not a reliable measure of password strength [7].

2. Review of Existing Password Strength Checking Approaches

2.1 Rule-Based Methods

Rule-based password checkers use predefined conditions such as minimum length, character classes, repeated characters, sequential characters, dictionary words, or known-compromised passwords. They are straightforward to implement and easy to explain to users. NIST recommends checking passwords against lists of commonly used, expected, or compromised values [1]. However, a rule-based score can overestimate strength when a password satisfies formal requirements but follows a predictable pattern. Research on password meters found that the strictness and scoring behavior of meters can substantially affect the security of passwords users create [8].

2.2 Entropy-Based Methods

Entropy is an information-theoretic concept that can be useful when a secret is generated from a known probability distribution. For human-chosen passwords, however, estimating entropy is difficult because password choices are not uniformly random. NIST explicitly notes that estimating entropy for user-chosen passwords is challenging and discusses password length and guessing resistance as more practical considerations [1]. A simple calculation such as $\log_2$ of a theoretical character-space size can therefore produce an overly optimistic result when it assumes random selection that users do not actually make.

2.3 Machine-Learning-Based Methods

Machine-learning methods attempt to learn statistical patterns from password data rather than relying only on manually written rules. Melicher and colleagues demonstrated that artificial neural networks could model password guessing and, in their evaluation, could

outperform probabilistic context-free grammars and Markov models [4], [5]. Their work also showed that a neural-network model could be compressed to a small size and used for practical client-side password checking with sub-second latency [4]. Later work combined neural-network strength estimation with data-driven heuristics to produce human-understandable feedback rather than only a numerical score [9].

Machine learning is not automatically more accurate simply because it is AI. Its performance depends on the training data, model design, evaluation method, and threat model. Research on password guessability has demonstrated that different guessing approaches have systematic biases and that a single algorithm should not be treated as a universal representation of attacker behavior [3]. More recent evaluation work similarly concludes that there is no single accuracy metric or single password-strength measure that is sufficient for every guessing scenario [6].

3. Proposed System

The proposed Password Strength Analyzer Based on AI is a small defensive web or desktop application designed for a final-year BCA project. Its purpose is to accept a password locally, analyze characteristics and learned patterns, and return an interpretable strength category with improvement feedback. The system should not store the entered password or send it to an external server. The project can use a lightweight machine-learning classifier rather than attempting to build a large-scale password-cracking system.

3.1 What Makes It Different from a Rule-Based Checker

A conventional checker may assign points for length, uppercase letters, digits, symbols, and other manually specified conditions. The proposed system would retain a small set of security rules but add a learned component that estimates whether the overall structure resembles patterns associated with weak or stronger passwords in the training data. This means that two passwords with similar length and character categories could receive different assessments when their learned structural patterns differ. This design follows the research direction in which neural networks are used to model password guessing rather than merely counting character classes [4], [5].

3.2 Main Modules

Module	Purpose
User Interface	Accept a password and display strength and feedback without exposing or storing the password.
Preprocessing	Convert the password into model features or an appropriate representation while keeping processing local.
Rule-Based Feature Extractor	Measure length, character categories, repetition, sequences, and selected common-pattern indicators.
ML Model	Classify or estimate password strength using a lightweight trained model.
Feedback Engine	Translate model/rule results into understandable suggestions.
Evaluation Module	Measure accuracy, precision/recall or similar classification metrics on a held-out test set.

3.3 Feasible AI Workflow

A feasible semester-scale implementation is to prepare a labeled dataset of passwords or password-like strings from lawful, publicly available research datasets, remove or protect sensitive material as appropriate, and extract non-reversible structural features. A supervised classifier such as logistic regression, a decision tree/random forest, or a small neural network can then be trained to predict predefined strength categories. The final system can combine the model output with security rules and a blocklist check. The research literature supports the general idea that learned models can estimate password guessability, while the proposed project deliberately uses a smaller model and narrower objective so that one student can implement and evaluate it within a semester [4], [5].

The output should preferably be expressed as categories such as Weak, Moderate, Strong, and Very Strong, together with reasons such as excessive repetition, common pattern, short length, or predictable substitution. A categorical output is more suitable for a student project than claiming an exact number of guesses because real-world guessability depends on the attack model and training data. Research has specifically warned that password-meter accuracy depends on the guessing scenario and that there is no single universal strength metric [3], [6].

4. Discussion and Limitations

The proposed analyzer should not be presented as a guarantee of password security. A password meter estimates resistance under assumptions; it does not prove that an attacker cannot guess the password. Research has shown that automated guessing approaches can differ in effectiveness and can systematically favor or miss particular password features [3]. Consequently, a model trained on one dataset may give an inaccurate estimate for passwords created by a different population or under a different service context.

Dataset bias is another important limitation. Password datasets can reflect the users, websites, languages, time periods, and password policies represented in the source data. A model trained mainly on one population may learn patterns that do not generalize well to other populations. This is a reason to report the dataset composition and evaluate the model on data that was not used during training rather than presenting training accuracy as evidence of real-world performance. The broader research literature on password guessability has shown that the choice and configuration of guessing methods can influence conclusions about particular password characteristics [3].

The analyzer also cannot detect every form of compromise. Password strength does not protect against phishing, malware, credential theft, or reuse of the same password on another service. NIST explicitly notes that passwords are not phishing-resistant [1]. A strong-looking password can therefore still be compromised through an attack that does not involve guessing.

False confidence is a further usability risk. A user who receives a “strong” label may interpret it as a guarantee. The interface should therefore describe the result as an estimate of resistance to guessing and should avoid claiming that any password is unbreakable. Research on password-strength meters also shows that feedback can influence user behavior, meaning that the wording and strictness of the meter are part of the security design rather than merely visual features [8], [9].

Finally, the project should avoid collecting users’ real passwords for experimentation. A defensive prototype can be evaluated with controlled test cases and appropriately handled research datasets. The system’s objective is password-strength estimation, not password recovery or credential collection.

5. Conclusion

An AI-based password strength analyzer is a realistic and academically meaningful BCA capstone when its objective is limited to estimating password resistance to guessing and providing understandable feedback. Existing evidence shows that simple composition rules

do not fully represent password strength, while machine-learning approaches can model password-guessing behavior more directly [1], [4], [5]. The proposed system therefore combines a small rule-based feature layer, a lightweight machine-learning classifier, a common/compromised-password check, and a feedback module. Its main contribution is not to claim perfect prediction, but to demonstrate how learned password-pattern analysis can complement conventional password meters. Because password strength depends on the attacker model, training data, and context, the system must communicate uncertainty and must not be treated as a security guarantee [3], [6]. Within a one-semester BCA project, this scope is technically feasible while still allowing meaningful experimentation, evaluation, and discussion of the limitations of AI-based security tools.

References

- [1] National Institute of Standards and Technology (NIST), *Digital Identity Guidelines: Authentication and Lifecycle Management*, NIST Special Publication 800-63B, 2025. <https://pages.nist.gov/800-63-4/sp800-63b.html>
- [2] National Institute of Standards and Technology (NIST), *NIST Special Publication 800-63B, Digital Identity Guidelines: Authentication and Lifecycle Management*, 2017. <https://pages.nist.gov/800-63-3/sp800-63b.html>
- [3] Blase Ur, Sean M. Segreti, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Saranga Komanduri, Darya Kurilova, Michelle L. Mazurek, William Melicher, and Richard Shay, “Measuring Real-World Accuracies and Biases in Modeling Password Guessability,” *Proceedings of the 24th USENIX Security Symposium*, 2015, pp. 463–481. <https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/ur>
- [4] William Melicher, Blase Ur, Sean M. Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor, “Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks,” *Proceedings of the 25th USENIX Security Symposium*, 2016, pp. 175–191. <https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/melicher>
- [5] William Melicher, Blase Ur, Sean M. Segreti, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor, “Better Passwords through Science (and Neural Networks),” *login:*, Vol. 42, No. 4, Winter 2017. <https://www.usenix.org/publications/login/winter2017/melicher>
- [6] Ding Wang, Xuan Shan, Qiying Dong, Yaosheng Shen, Chunfu Jia, “No Single Silver Bullet: Measuring the Accuracy of Password Strength Meters,” *Proceedings of the 32nd USENIX Security Symposium*, 2023, pp. 947–964. <https://www.usenix.org/conference/usenixsecurity23/presentation/wang-ding-silver-bullet>

[7] Blase Ur, Fumiko Noma, Jonathan Bees, Sean M. Segreti, Richard Shay, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor, “I Added ‘!’ at the End to Make It Secure: Observing Password Creation in the Lab,” *Proceedings of the Eleventh Symposium on Usable Privacy and Security (SOUPS)*, 2015. <https://www.usenix.org/conference/soups2015/proceedings/presentation/ur>

[8] Blase Ur, Patrick Gage Kelley, Saranga Komanduri, Joel Lee, Michael Maass, Michelle L. Mazurek, Timothy Passaro, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor, “How Does Your Password Measure Up? The Effect of Strength Meters on Password Creation,” *Proceedings of the 21st USENIX Security Symposium*, 2012, pp. 65–80. <https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/ur>

[9] Blase Ur, “Helping Users Make Better Passwords Through Data-Driven Methods,” *SOUPS 2017 Workshop Program*, USENIX Association, 2017. <https://www.usenix.org/conference/soups2017/workshop-program/way2017>

Verification note: this is Neha’s own self-built research paper, produced from the Prompting_Guide.pdf issued earlier in this project, checked against the programme’s citation-verification protocol before delivery. All nine references were individually confirmed against publisher records (NIST, USENIX, dblp) — this is the cleanest self-built draft verified in this programme to date. Only one correction was needed: reference [3] was missing a co-author (Lorrie Faith Cranor) from the real paper’s ten-author list — added back in the correct author order. No fabricated citations, no invented results, no leftover AI commentary, and every claim about the proposed system is correctly hedged as proposed rather than demonstrated, exactly as the programme’s protocol requires. This document supersedes the earlier mentor-built Research_Paper.md, which is superseded and kept only for reference.