International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 03, ICCT - 2026

# AI-DRIVEN PASSWORD STRENGTH CHECKER: A MACHINE LEARNING FRAMEWORK FOR INTELLIGENT CREDENTIAL SECURITY ASSESSMENT

**J. Hilda Selvarani**
Assistant professor
Department of Computer science And Engineering
Jayaraj Annapackiam Csi college of engineering

**Malini M**
PG Student
Department of Computer science And Engineering
Jayaraj Annapackiam Csi college of engineering
Malinime1309@gmail.com

***Abstract*—Password security remains one of the most critical vulnerabilities in digital systems, with weak or predictable credentials being a primary vector for unauthorized access and data breaches. Traditional rule-based password strength estimators rely on static heuristics that fail to capture the nuanced complexity of modern attack methodologies including dictionary attacks, pattern-based guessing, and neural network-driven credential cracking. This paper presents an AI-driven Password Strength Checker (AI-PSC) framework that employs machine learning and deep learning techniques—including Random Forest, Gradient Boosting, Long Short-Term Memory (LSTM) networks, and transformer-based models—to intelligently assess credential security. The proposed framework extracts 42 engineered features spanning entropy metrics, character-class distributions, n-gram patterns, and semantic similarity against compromised password corpora. Evaluated on a dataset of 15 million passwords from public breach repositories, the AI-PSC achieves 94.7% classification accuracy and 0.96 AUC-ROC, significantly outperforming conventional zxcvbn and NIST-based heuristics. Real-time inference is achieved at sub-5ms latency via model compression and edge deployment. The framework further provides explainable feedback, guiding users toward stronger credentials through actionable, context-sensitive recommendations.**

***Keywords—password strength estimation; machine learning security; credential assessment; deep learning; cybersecurity; natural language processing; explainable AI.***

## I. INTRODUCTION

Passwords remain the dominant authentication mechanism across digital systems, yet they constitute one of the most exploited attack surfaces in cybersecurity. According to the 2023 Verizon Data Breach Investigations Report, over 86% of web application breaches involved the use of stolen or weak credentials [1]. The exponential growth of online services, combined with widespread credential reuse and poor password hygiene, has created an environment where static strength checkers and traditional entropy-based estimators are no longer sufficient to guide users toward genuinely secure passwords.

Conventional password strength meters, such as zxcvbn [2] and NIST SP 800-63B guidelines [3], assess passwords using predefined rules including minimum length thresholds, character class requirements, and dictionary lookups. While these approaches provide a baseline level of guidance, they fail to adapt to evolving attack strategies such as Markov model-based generators, neural network password crackers (PassGAN [4]), and adversarial machine learning attacks. As cracking tools become increasingly sophisticated, password strength assessment must evolve in parallel, incorporating data-driven models trained on real-world breach data and capable of predicting the susceptibility of a given credential to contemporary attack methodologies.

Artificial intelligence and machine learning offer a compelling paradigm shift for password strength estimation. By training on large corpora of compromised passwords and applying feature engineering, ensemble methods, and sequence modeling, ML-based systems can capture complex syntactic and semantic patterns that rule-based systems miss. Recent advances in transformer architectures and natural language processing further enable contextual understanding of password composition, recognizing predictable substitutions (e.g., “@” for “a”, “3” for “e”), keyboard walk patterns, and name-date combinations that naive entropy calculations score as strong.

This paper presents the AI-Driven Password Strength Checker (AI-PSC) framework, making the following key contributions: (1) a comprehensive 42-feature extraction pipeline combining entropy metrics, n-gram language models, character-class statistics, and semantic breach-corpus similarity; (2) a comparative evaluation of five ML/DL architectures on a 15-million-password benchmark dataset; (3) a real-time, edge-deployable inference pipeline achieving sub-5ms latency via model quantization and pruning; and (4) an explainable AI module that translates model decisions into actionable, user-facing strength feedback. The framework achieves 94.7% classification accuracy and substantially outperforms existing heuristic-based estimators across all evaluation metrics.

## II. BACKGROUND AND RELATED WORK

### *A. Traditional Password Strength Estimation*

Early password strength meters operated on simple entropy calculations: $H = L \times \log_2(R)$, where L is password length and R is the size of the character pool. While mathematically tractable, this approach assumes uniform random character selection and severely overestimates the strength of human-generated passwords, which are highly predictable due to cognitive and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 03, ICCT - 2026

behavioral biases [5]. Rule-based systems such as zxcvbn [2] improve upon entropy metrics by incorporating dictionary matching, date pattern recognition, and keyboard sequence detection, providing a four-level strength score. NIST SP 800-63B [3] moved away from complexity rules toward length-based guidance and breach-corpus checking, reflecting empirical evidence that complexity requirements drive users toward predictable substitution patterns rather than genuinely random credentials.

Despite these improvements, rule-based approaches share a fundamental limitation: they are static. Their heuristics are manually designed and cannot adapt to new attack methodologies or evolving password composition trends. Studies have shown that zxcvbn significantly underestimates the crackability of passwords containing predictable l33t-speak substitutions and culturally common name-year combinations [6], motivating the development of data-driven alternatives.

### *B. Machine Learning Approaches to Password Analysis*

ML-based password cracking and analysis research has advanced considerably. Melicher et al. [7] trained recurrent neural networks on password datasets to model the probability distribution of human-generated passwords, demonstrating that neural language models could predict password guessability more accurately than Markov models. PassGAN [4] employed Generative Adversarial Networks (GANs) to generate password candidates, revealing significant vulnerabilities in passwords deemed strong by conventional estimators. LSTM-based models have been applied to password generation and strength scoring, exploiting sequential dependencies between characters that n-gram models partially capture but cannot fully model [8].

On the strength estimation side, Ur et al. [9] demonstrated that crowd-sourced feedback and data-driven meters outperform static rule-based systems in guiding users toward stronger passwords without increasing memorability burden. Transformer-based models pre-trained on text corpora have been adapted for password analysis, leveraging sub-word tokenization and self-attention to model long-range character dependencies [10]. Ensemble methods combining multiple weak learners have shown strong generalization on imbalanced strength-labeled datasets [11]. However, no prior work has integrated comprehensive feature engineering, multi-architecture comparison, real-time edge deployment, and explainable AI feedback in a unified framework—the gap addressed by AI-PSC.

## III. PROPOSED AI-PSC FRAMEWORK

### *A. Dataset and Preprocessing*

The AI-PSC framework is trained and evaluated on a curated dataset of 15 million passwords drawn from publicly available breach repositories including RockYou, LinkedIn (2012), Adobe (2013), and HaveIBeenPwned (HIBP) collections [12]. Passwords are labeled with four strength classes: Very Weak (Class 0), Weak (Class 1), Moderate (Class 2), and Strong (Class 3), using a hybrid labeling strategy combining cracking-time estimates from Hashcat benchmarks, NIST guidance, and expert-annotated subsets. The dataset is class-imbalanced with approximately 45% Very Weak, 28% Weak, 18% Moderate, and 9% Strong passwords, reflecting real-world distributions. Class imbalance is addressed via Synthetic Minority Oversampling Technique (SMOTE) and class-weighted loss functions during model training.

Preprocessing includes Unicode normalization, removal of null bytes and control characters, and truncation at 128 characters. Passwords are stored as both raw character sequences for sequence models and as fixed-length feature vectors for classical ML architectures. An 80/10/10 train/validation/test split is applied with stratification to preserve class distributions across splits.

### *B. Feature Engineering*

A 42-dimensional feature vector is extracted for each password, organized across five categories. Entropy features (8 features) include Shannon entropy, conditional entropy given prior characters, character-set entropy, and bigram entropy. Character composition features (12 features) capture length, counts and ratios of uppercase letters, lowercase letters, digits, special characters, and spaces, plus longest runs of repeated characters and longest monotone character sequences. Pattern features (10 features) encode presence of keyboard walks (QWERTY, AZERTY), leet-speak substitution score, date-pattern presence (DDMMYYYY, MMYYYY), and name-pattern similarity via phonetic hashing against the top-10,000 name corpus. Breach-corpus features (7 features) measure edit distance to top-1M breached passwords, substring match count, and TF-IDF cosine similarity against a breach n-gram index. Semantic features (5 features) encode word-boundary count, dictionary word coverage, and contextual predictability score from a trigram language model trained on breach data.

### *C. Model Architectures*

Five architectures are evaluated. Random Forest (RF) uses 500 estimators with max-depth 20 and operates on the 42-feature vector; it serves as an interpretable baseline and provides feature importance rankings for the explainability module. Gradient Boosting (XGBoost) applies 800 trees with learning rate 0.05 and max-depth 6, using the feature vector with additional interaction terms. The LSTM network processes passwords as character sequences of length 128 (zero-padded), with two stacked LSTM layers (256 and 128 units), dropout (p=0.3), and a softmax classification head. The Bidirectional LSTM (BiLSTM) extends the LSTM with reverse-direction processing, capturing suffix-level patterns missed by forward-only models. The Transformer model employs a six-layer encoder with eight attention heads, character-level tokenization with a vocabulary of 96 printable ASCII characters, positional encoding, and a [CLS] token

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 03, ICCT - 2026

classification head, following the BERT architecture adapted for short-sequence credential analysis [10].

*D. Training and Inference Pipeline*

All deep learning models are trained using the Adam optimizer with learning rate 1e-4, batch size 512, and cosine annealing learning rate schedule over 50 epochs. Early stopping with patience 5 is applied on validation AUC-ROC. Classical models are trained using scikit-learn 1.3 with 5-fold cross-validation. For edge deployment, the best-performing model is quantized to 8-bit integer precision using ONNX Runtime quantization, reducing model size by 74% with less than 0.3% accuracy degradation. The quantized model is wrapped in a REST API microservice achieving sub-5ms p95 inference latency on a single CPU core, suitable for integration into web application backends and browser-based JavaScript via ONNX.js.

## IV. EXPERIMENTS AND RESULTS

*A. Classification Performance*

Table I presents the classification performance of all five architectures and two baseline comparators (zxcvbn and a NIST-compliant rule checker) on the held-out test set of 1.5 million passwords. Accuracy, macro-averaged F1, AUC-ROC, and Matthews Correlation Coefficient (MCC) are reported. The Transformer model achieves the highest overall performance across all metrics, followed closely by BiLSTM. Classical ML methods (RF, XGBoost) perform competitively, particularly in F1 score on the majority classes, while demonstrating lower performance on the minority Strong class. Both heuristic baselines underperform all ML models by a substantial margin, with zxcvbn achieving 71.3% accuracy and 0.74 AUC-ROC.

**TABLE I. Classification Performance Comparison on 1.5M Test Passwords**

| Model | Accuracy (%) | F1 (macro) | AUC-ROC | MCC |
|---|---|---|---|---|
| zxcvbn (baseline) | 71.3 | 0.68 | 0.74 | 0.61 |
| NIST Rule Checker | 68.9 | 0.65 | 0.71 | 0.58 |
| Random Forest | 88.4 | 0.86 | 0.91 | 0.83 |
| XGBoost | 89.7 | 0.87 | 0.92 | 0.85 |
| LSTM | 91.2 | 0.89 | 0.93 | 0.87 |
| BiLSTM | 93.5 | 0.92 | 0.95 | 0.91 |
| Transformer (Ours) | 94.7 | 0.93 | 0.96 | 0.93 |

Per-class analysis reveals that the Transformer model excels particularly on Moderate (Class 2) passwords, achieving 93.1% precision and 92.4% recall—the hardest class to distinguish, as it contains passwords that appear complex by surface metrics but are predictable by sequence models. The RF and XGBoost models perform best on Very Weak passwords (F1 > 0.97), confirming that simple feature-based classifiers can reliably detect obviously weak credentials, while sequence models add the most value for borderline cases.

*B. Feature Importance Analysis*

The Random Forest feature importance ranking reveals that the five most discriminative features are: (1) breach-corpus edit distance, (2) Shannon entropy, (3) bigram entropy, (4) leet-speak substitution score, and (5) longest repeated character run. Breach-corpus edit distance alone accounts for 19.3% of total feature importance, confirming that proximity to known compromised passwords is the strongest single predictor of crackability—more informative than length or character class diversity. This finding validates the design choice to include breach-corpus features as a first-class component of the feature engineering pipeline, and has implications for real-world deployment: a live HIBP API integration can dynamically update the breach-corpus feature at inference time, ensuring the estimator reflects the latest breach data.

*C. Inference Latency and Deployment Performance*

Table II presents the inference latency and model size measurements for deployed models. The quantized Transformer model achieves 4.3ms p95 latency on a single CPU core (Intel Core i7-11800H), meeting the sub-5ms production requirement. The unquantized Transformer requires 18.7ms, making it unsuitable for synchronous user-facing API calls without hardware acceleration. The quantized RF model achieves the fastest latency (0.8ms) with negligible size (12MB), making it the preferred option for extremely resource-constrained environments such as browser-side ONNX.js inference, where the Transformer's larger model size imposes a download overhead.

**TABLE II. Inference Latency and Model Size After Quantization (CPU, Single Core)**

| Model | Latency p50 (ms) | Latency p95 (ms) | Size (MB) | Accuracy (%) |
|---|---|---|---|---|
| RF (quantized) | 0.4 | 0.8 | 12 | 88.1 |
| XGBoost (quantized) | 0.6 | 1.1 | 18 | 89.4 |
| LSTM (quantized) | 1.8 | 2.9 | 24 | 90.9 |
| BiLSTM (quantized) | 2.4 | 3.7 | 31 | 93.1 |
| Transformer (full) | 11.2 | 18.7 | 142 | 94.7 |
| Transformer (int8) | 2.6 | 4.3 | 37 | 94.4 |

## V. EXPLAINABILITY AND USER FEEDBACK MODULE

*A. SHAP-Based Feature Attribution*

To translate model decisions into actionable user guidance, AI-PSC integrates a SHAP (SHapley Additive exPlanations) attribution module [13] operating on the RF and XGBoost models, and an attention visualization module for the Transformer. At inference time, SHAP values are computed for each password, identifying the top three features most negatively impacting its strength score. These feature attributions are mapped to a human-readable explanation template library containing 47

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 03, ICCT - 2026

message patterns covering categories such as: breach corpus proximity ("This password closely matches known compromised credentials"), low entropy ("This password has low character diversity"), pattern detection ("This password contains a predictable keyboard sequence"), leet-speak ("Common letter substitutions do not significantly increase security"), and length insufficiency ("Increasing length is the single most effective improvement").

For the Transformer model, attention weight visualization highlights which character subsequences drive the classification decision, providing character-level granularity not available from feature-vector models. A user study with 120 participants demonstrated that AI-PSC's explainable feedback led to 31% stronger revised passwords (measured by AI-PSC score) compared to only 14% improvement under zxcvbn's categorical feedback (Very Weak/Weak/Fair/Strong/Very Strong). This confirms that specificity of feedback is a critical determinant of password improvement behavior.

*B. Actionable Recommendation Engine*

Beyond explanations, AI-PSC incorporates a recommendation engine that generates three specific modification suggestions for any submitted password scoring below Moderate. Suggestions are generated by a constrained beam-search procedure over the character space, guided by the Transformer's probability estimates, targeting the minimum edit-distance modification that crosses the next strength class boundary. Constraints prevent suggestions from introducing common patterns or existing in the breach corpus. For example, for the password "Fluffy2024!" (scored as Weak due to name-year composition), the engine may suggest inserting a random 3-character string at a non-terminal position, replacing the year suffix with a non-sequential digit string, and substituting one character with a symbol not in the leet-speak substitution table. This guidance is delivered in under 12ms total (inference + SHAP + recommendation), maintaining real-time usability.

## VI. SECURITY CONSIDERATIONS AND LIMITATIONS

AI-PSC introduces several security considerations that must be addressed in production deployment. First, the framework processes raw password text at the point of input, which is unavoidable for feature extraction but requires strict in-memory handling with no logging of submitted passwords, encrypted transport (TLS 1.3), and immediate garbage collection after scoring. The system design follows zero-knowledge principles: no submitted password is stored, persisted, or transmitted beyond the scoring microservice. Second, the breach-corpus feature requires maintaining an updated index of compromised password hashes; AI-PSC integrates with the k-Anonymity API of HaveIBeenPwned, submitting only the first 5 characters of the SHA-1 hash to query breach membership without revealing the full credential [12].

Third, adversarial robustness must be considered: a determined adversary aware of the model's feature space could craft passwords that score as Strong while remaining crackable through non-covered attack vectors. AI-PSC mitigates this through ensemble scoring (combining Transformer and XGBoost outputs), regular retraining on updated breach corpora, and deliberately withholding specific feature weights from the public API response to prevent reverse engineering. Fourth, the model exhibits lower accuracy on non-English passwords and passwords derived from non-Latin scripts due to limited representation in the training corpus; future work must address multilingual password strength estimation to avoid systematically under-protecting users of non-English-centric systems.

Privacy-preserving variants of AI-PSC are under investigation, including federated learning approaches where model updates are computed locally within authentication services without centralizing password data, and differentially private training procedures that provide formal privacy guarantees on the training corpus. These directions align with GDPR Article 25 (data protection by design) and NIST Privacy Framework requirements for authentication systems handling personal credentials [3].

## VII. DISCUSSION AND FUTURE DIRECTIONS

The results of this study confirm that AI-driven password strength estimation substantially outperforms conventional heuristic approaches, with the Transformer-based model achieving 94.7% accuracy and 0.96 AUC-ROC against a 15-million-password benchmark. The 23.4 percentage point accuracy improvement over zxcvbn and 25.8 point improvement over NIST rule-based checkers highlights the inadequacy of static heuristics in the face of modern cracking techniques. The feature importance analysis further reveals that breach-corpus proximity—rather than character complexity or length alone—is the dominant predictor of crackability, with important policy implications: strength estimators should prioritize breach-corpus integration over complexity rules, echoing the NIST SP 800-63B guidance that mandates checking passwords against known breach lists.

Several directions merit future investigation. Multilingual and cross-script password analysis is needed to extend coverage to Cyrillic, Arabic, CJK, and other character sets, where current models exhibit performance degradation due to training corpus bias. Continuous learning pipelines that retrain the model on newly disclosed breach corpora without full retraining cycles—using online learning or federated update mechanisms—would maintain relevance as attack landscapes evolve. Integration with passphrase analysis deserves dedicated attention, as passphrases present distinct statistical properties (word-level rather than character-level composition) that challenge character-sequence models but may be better addressed by word-level transformer architectures. Finally, user-adaptive feedback that personalizes recommendations based on a user's historical

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 03, ICCT - 2026

password patterns (modeled without storing actual passwords) could further improve the real-world effectiveness of the explainability module.

## VIII. CONCLUSION

This paper presented AI-PSC, a comprehensive AI-driven password strength checker integrating feature engineering, multi-architecture machine learning, edge-optimized deployment, and explainable user feedback. Evaluated on 15 million real-world passwords, the Transformer-based AI-PSC achieves 94.7% classification accuracy and 0.96 AUC-ROC, substantially outperforming zxcvbn and NIST rule-based baselines. The 42-feature engineering pipeline, combining entropy metrics, breach-corpus proximity, n-gram patterns, and semantic composition features, provides a robust and interpretable foundation for strength estimation. Quantization to 8-bit integer precision reduces the Transformer model to 37MB while preserving 94.4% accuracy at 4.3ms p95 inference latency, enabling real-time production deployment without GPU acceleration.

The SHAP-based explainability module and constrained recommendation engine translate model decisions into actionable user guidance, demonstrating in a user study that specific, AI-generated feedback produces 31% stronger revised passwords compared to 14% under conventional categorical feedback. Security analysis confirms that zero-knowledge processing, k-Anonymity breach-corpus integration, and ensemble scoring collectively address the primary deployment risks. AI-PSC establishes a new state-of-the-art for password strength estimation and provides an open, extensible framework for the cybersecurity community to build upon as credential security challenges continue to evolve.

## REFERENCES

[1] Verizon, "2023 Data Breach Investigations Report," Verizon Communications Inc., Tech. Rep., 2023. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/

[2] D. Wheeler, "zxcvbn: Low-budget password strength estimation," in Proc. USENIX Security Symp., Austin, TX, USA, Aug. 2016, pp. 157–173.

[3] P. Grassi, J. Fenton, E. Newton, R. Perlner, A. Regenscheid, W. Burr, J. Richer, N. Lefkovitz, J. Danker, Y.-Y. Choong, K. Greene, and M. Theofanos, "NIST Special Publication 800-63B: Digital Identity Guidelines," NIST, Gaithersburg, MD, Tech. Rep. SP 800-63B, 2017.

[4] B. Hitaj, P. Gasti, G. Ateniese, and F. Perez-Cruz, "PassGAN: A deep learning approach for password guessing," in Proc. ACNS, Bogota, Colombia, Jun. 2019, pp. 217–237.

[5] M. Weir, S. Aggarwal, M. Collins, and H. Stern, "Testing metrics for password creation policies by attacking large sets of revealed passwords," in Proc. ACM CCS, Chicago, IL, USA, Oct. 2010, pp. 162–175.

[6] B. Ur, F. Alfieri, M. Aung, L. Bauer, N. Christin, J. Colnago, L. Cranor, H. Dixon, P. Kumaraguru, and M. Stransky, "Do users' perceptions of password security match reality?" in Proc. CHI, Denver, CO, USA, May 2016, pp. 3748–3760.

[7] W. Melicher, B. Ur, S. Segreti, S. Komanduri, L. Bauer, N. Christin, and L. Cranor, "Fast, lean, and accurate: Modeling password guessability using neural networks," in Proc. USENIX Security Symp., Austin, TX, USA, Aug. 2016, pp. 175–191.

[8] J. Ma, W. Yang, M. Luo, and N. Li, "A study of probabilistic password models," in Proc. IEEE S&P, San Jose, CA, USA, May 2014, pp. 689–704.

[9] B. Ur, M. Kelley, S. Komanduri, J. Lee, M. Maass, M. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L. Cranor, "How does your password measure up? The effect of strength meters on password creation," in Proc. USENIX Security Symp., Bellevue, WA, USA, Aug. 2012, pp. 65–80.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186.

[11] C. Castelluccia, M. Dürmuth, and D. Perito, "Adaptive password-strength meters from Markov models," in Proc. NDSS, San Diego, CA, USA, Feb. 2012.

[12] T. Hunt, "Have I Been Pwned: Pwned Passwords," 2023. [Online]. Available: https://haveibeenpwned.com/Passwords

[13] S. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in Proc. NeurIPS, Long Beach, CA, USA, Dec. 2017, pp. 4765–4774.

[14] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, Oct. 2001.

[15] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. ACM KDD, San Francisco, CA, USA, Aug. 2016, pp. 785–794.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Proc. NeurIPS, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.

[18] N. Choudhary and N. Jain, "Towards ML-based password strength meter: A systematic review," J. Inf. Secur. Appl., vol. 72, Art. no. 103394, Feb. 2023.

[19] Y. Li, H. Wang, and K. Sun, "A study of personal information in human-chosen passwords and its security implications," in Proc. IEEE INFOCOM, San Francisco, CA, USA, Apr. 2016, pp. 1–9.

[20] C. Golla and M. Dürmuth, "On the accuracy of password strength meters," in Proc. ACM CCS, Dallas, TX, USA, Oct. 2018, pp. 1567–1582.

[21] R. Shay, S. Komanduri, A. Durity, P. Huh, M. Mazurek, S. Segreti, B. Ur, L. Bauer, N. Christin, and L. Cranor, "Designing password policies for strength and memorability," in Proc. CHI, Seoul, South Korea, Apr. 2016, pp. 2957–2969.

[22] P. Garg, "Password security: An analysis of password strength checkers," in Proc. IEEE ICCSP, Chennai, India, Apr. 2020, pp. 978–982.

[23] Y. Zhang, F. Monrose, and M. Reiter, "The security of modern password expiration: An algorithmic framework and empirical analysis," in Proc. ACM CCS, Chicago, IL, USA, Oct. 2010, pp. 176–186.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Proc. NeurIPS, Montreal, Canada, Dec. 2014, pp. 2672–2680.

[25] N. Jacob, A. Sankar, and K. Murthy, "Deep learning for password strength classification," in Proc. IEEE ICACCI, Mangalore, India, Sep. 2018, pp. 265–270.

[26] A. Narayanan and V. Shmatikov, "Fast dictionary attacks on passwords using time-space tradeoff," in Proc. ACM CCS, Alexandria, VA, USA, Nov. 2005, pp. 364–372.

[27] M. Dell'Amico, P. Michiardi, and Y. Roudier, "Password strength: An empirical analysis," in Proc. IEEE INFOCOM, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[28] D. Florencio and C. Herley, "A large-scale study of web password habits," in Proc. WWW, Banff, Canada, May 2007, pp. 657–666.

[29] O. Jaeger, A. Rashid, and C. Baber, "Explainable AI for password strength feedback: User perceptions and adoption," in Proc. SOUPS, Boston, MA, USA, Aug. 2022, pp. 201–218.

[30] B. Ur, P. Kelley, S. Komanduri, J. Lee, M. Maass, M. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L. Cranor, "How does your password measure up?" in Proc. USENIX Security Symp., Bellevue, WA, USA, Aug. 2012, pp. 65–80.