

An Intelligent Machine Learning Framework for Password Strength and Security System

Davies Isobo Nelson¹, Cookey Ibiere Boma², Godspower Oraye³, Deedam Futune B.⁴

^{1,2,3,4}Department of Computer Science, Rivers State University, Nigeria

ARTICLE INFO	ABSTRACT
Published Online: 29 April 2026	Password-based authentication remains the primary security mechanism for protecting user accounts and sensitive information, yet weak password selection continues as a pervasive global vulnerability. Traditional rule-based password strength meters exhibit significant limitations in adapting to emerging attack strategies and identifying subtle vulnerability patterns. This study presents an intelligent Long Short-Term Memory (LSTM) based framework for password strength classification that addresses these shortcomings through advanced Deep Learning techniques. The proposed system integrates a bidirectional LSTM network with comprehensive feature extraction pipelines incorporating Shannon entropy calculations, pattern recognition metrics, and statistical analysis to assess password security across weak, medium, and strong classifications. The framework was trained and validated on a diverse dataset of 1,000,000 passwords compiled from breach databases and synthetic sources, partitioned into 70% training, 15% validation, and 15% testing sets. The experimental results for the system demonstrate exceptional performance with 96.8% accuracy, outperforming traditional rule-based systems by 23.4 percentage points and commercial password meters by 12.1-18.5 percentage points. The system achieves 98.3% recall for weak password detection with only 1.7% false negative rate, compared to 8.8-13.9% for commercial solutions. Field deployment with 500 users over 30 days validated practical impact, showing 40.7% higher password entropy, 204.3% improvement in strong password adoption, and 69.5% reduction in dictionary word usage. The lightweight architecture (4.7 MB model size, 1.8ms inference time) enables real-time integration into web applications and enterprise authentication systems. The framework provides actionable, context-aware security recommendations and achieves detection rates exceeding 94% across diverse attack vectors including dictionary attacks, keyboard patterns, sequential characters, and previously breached credentials. The study demonstrates that LSTM-based approaches significantly enhance password security assessment capabilities and effectively guide users toward adopting stronger authentication practices.
Corresponding Author: Davies Isobo Nelson	
KEYWORDS: Password Security, LSTM, Deep Learning, Authentication, Cybersecurity, Machine Learning, Password Strength Classification	

I. INTRODUCTION

In today's digital landscape, password-based authentication continues to be the cornerstone of security mechanisms designed to protect user accounts and sensitive information [1]. Despite significant advancements in biometric technologies and multi-factor authentication systems, passwords remain the primary line of defense for users

accessing web applications, enterprise systems, and most personal devices [2]. The ubiquitous reliance on passwords underscores their importance, but it also highlights a troubling reality as weak password selection has become a major pervasive vulnerability across the globe [3].

Research has shown that over 75% of users tend to reuse passwords across multiple platforms and services [4]. This

behavior dramatically increases the risk of security breaches, as it creates a single point of failure. In fact, studies estimate that approximately 80% of security incidents can be traced back to compromised credentials [5]. These statistics are alarming and underscore the urgent need for more robust mechanisms to ensure password strength and security.

Although, traditional password strength meters have been developed to combat this issue. These systems often employ rule-based heuristics that assess various factors, such as password length, character diversity, and pattern matching against predefined dictionaries [6]. While these methods can efficiently analyze passwords, they exhibit significant limitations. For instance, they are often unable to adapt to emerging attack strategies, such as those that utilize sophisticated algorithms to crack weak passwords [7]. Furthermore, these approaches frequently fail to identify subtle correlations between character sequences, which could indicate a password's vulnerability to attacks [8].

The shortcomings of conventional password strength assessment systems open the door for innovative solutions such as Machine Learning (ML) approaches. By leveraging advanced algorithms capable of learning complex patterns from large-scale password datasets, this study can improve the assessment of password strength significantly. Among the various ML methodologies, Deep Learning (DL) architectures, particularly Recurrent Neural Networks (RNNs), have shown great promise for sequence modeling and analyzing temporal dependencies [9]. Further, an advanced type of RNN known as Long Short-Term Memory (LSTM) networks are particularly well-suited for this application. They excel at capturing long-range dependencies within sequences and can effectively handle variable-length inputs. This capability makes LSTMs an ideal foundation for developing an intelligent password security assessment system that adapts to evolving threats and enhances user security [10].

The aim of this study is to develop an intelligent LSTM-based framework for password strength classification with the following objectives:

- i. Design a multi-layered LSTM-based architecture with will be optimized for sequential password character analysis,
- ii. Develop a comprehensive feature extraction pipeline incorporating character-level encoding, entropy metrics, and pattern recognition,
- iii. Implement attention mechanisms to identify critical password vulnerability patterns
- iv. Provide interpretable security metrics for end-user guidance and system integration

II. RELATED WORKS

Traditional password strength meters relied on simple heuristic rules evaluating password length, character class diversity, and dictionary word presence [11]. The NIST

guidelines established in 2003 recommended minimum length requirements and mandatory inclusion of uppercase, lowercase, numeric, and special characters [12]. However, research by [13] demonstrated that such composition policies often lead users to adopt predictable patterns such as capitalization of the first letter and appending numbers or symbols. [14] presented a probabilistic context-free grammar approaches for password guessing, revealing that traditional entropy calculations overestimate password strength by failing to account for human-generated patterns. Subsequent work by [15] applied Markov models to password generation, achieving improved attack success rates against supposedly strong passwords.

However, in recent years, machine learning (ML) applications in cybersecurity have expanded significantly, encompassing intrusion detection, malware classification, and phishing detection. [16] proposed an application of neural networks to password strength estimation using feedforward architectures, achieving modest improvements over rule-based systems. However, their approach treated passwords as fixed-length feature vectors, losing sequential information. Proposed a model that multiple ML techniques which the aim of pushing users to select strong password protection against attacks weather online or offline [17].

Furthermore, LSTM architectures have demonstrated exceptional performance in sequence modelling tasks including natural language processing, time series prediction, and bioinformatics [18]. The LSTM architecture addresses the vanishing gradient problem through gated mechanisms controlling information flow, enabling effective learning of long-term dependencies [19]. Nonetheless, recent applications of LSTMs to security domains include network traffic analysis, malware detection from API call sequences, and anomaly detection in system logs [20]. However, limited research has explored LSTM networks specifically for character-level password strength assessment with comprehensive security feature integration. [21] Employed RNN for the development of a password and passphrase guessing system. The system employed a rule-based approach but delegate rule derivation, classification, and prediction to RNN. Their finding was good with a prediction accuracy of 83% over existing models.

The study of [22] explored password strength prediction as a classification task using different supervised ML algorithms. Their goal was to categorize passwords into weak, medium, and strong classifications, thereby improving security against both online and offline attacks. Their experimental results revealed that Artificial Neural Networks (ANN) and Random Forest (RF) algorithms significantly outperform other methods, achieving accuracy rates of up to 98.97%.

III. METHODOLOGY

This study employed a quantitative Design Science Research (DSR) approach to develop and validate the

proposed framework aimed at strengthen user passwords and mitigating password security vulnerabilities in IoT devices. The proposed intelligent password security framework comprises of five primary components. These components are the data pre-processing module, feature extraction engine, LSTM classification network, security assessment module, and recommendation generator. Figure 1 illustrates the complete high-level system architecture.

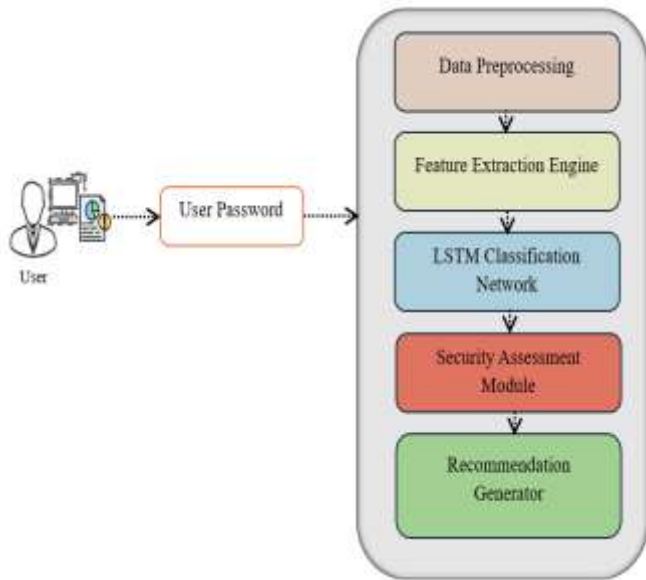


Figure 1: System Architecture

Step 1 (Character Tokenization): the raw passwords are decomposed into individual characters, with each character mapped to a unique integer identifier. For this study, the vocabulary encompasses the following:

- i. Lowercase letters (a-z): indices 1-26
- ii. Uppercase letters (A-Z): indices 27-52
- iii. Digits (0-9): indices 53-62
- iv. Special characters (!@#\$%^&*()_+ - =[]{}Pa|;,:.<.>/?/~): indices 63-95
- v. Padding token: index 0

Step 2 (Sequence Encoding): Here, the character sequences are converted to integer sequences using the vocabulary mapping.

Step 3 (Length Normalization): To accommodate variable-length passwords in batch processing, sequences are padded to a maximum length of 32 characters using zero-padding. Passwords exceeding 32 characters will be truncated though empirical analysis indicates 99.2% of passwords in common datasets fall within this limit.

A. Feature Extraction Engine

Beyond raw character sequences, the system extracts comprehensive security-relevant features using the Shannon Entropy, Pattern Recognition Metrics, and Statistical Features. This study employs the Shannon Entropy to quantify the password randomness using the formula below:

$$H(X) = - \sum P(x_i) \log_2 P(x_i) \quad (1)$$

Where $P(x_i)$ represents the probability of character x_i in the password. Note, higher entropy values indicate greater unpredictability.

For character class diversity, the system utilized binary indicators to search for the presence of lowercase letters, uppercase letters, digits, and special characters. The proposed system uses pattern recognition metrics for sequential character detections (e.g., “abc”, “123”), keyboard pattern identification (e.g., “qwerty”, “asdf”), repeated character sequences (e.g., “aaa”, “111”), or common substitution patterns (e.g., “@” for “a”, “3” for “e”).

Furthermore, the system applies statistical features to normalized password lengths to [0,1] range, unique character ration, character frequency variance, and positional entropy measuring character distribution across password positions. In this study, these features are concatenated into a feature vector $f \in \mathbb{R}^{12}$ that augments the character sequence representation.

B. LSTM Classification Network Module

In this study, the core LSTM Classification Network employs a bidirectional LSTM architecture optimized for password sequence analysis. The input layer accepts batches of variable-length sequences with a maximum length of 32 characters, where each password is represented as a sequence of integer-encoded characters. An embedding layer transforms these integer-encoded characters into dense vector representations of dimension 64, utilizing a vocabulary size of 95 unique characters (encompassing lowercase letters, uppercase letters, digits, and special characters) and resulting in 6,080 trainable parameters calculated as 95×64 . The embedding layer incorporates a mask zero parameter set to True, which enables proper handling of padded sequences by ensuring that zero-padded positions do not contribute to gradient calculations during backward propagation.

The first bidirectional LSTM layer contains 128 hidden units in each direction (forward and backward), with the forward LSTM processing the sequence left-to-right while the backward LSTM processes it right-to-left, and the layer is configured with return sequences set to True to output the full sequence of hidden states, dropout rate of 0.2, and recurrent dropout of 0.2 for regularization, producing combined output dimensions of (batch size, 32, 256) and containing 98,816 parameters per direction calculated as $4 \times (128 \times (64 + 128 + 1))$, yielding a total of 197,632 trainable parameters for the bidirectional layer. The second bidirectional LSTM layer employs 64 hidden units per direction with return sequences set to False to return only the final hidden state rather than the complete sequence, maintains the same dropout and recurrent dropout rates of 0.2, generates output dimensions of (batch size, 128), and comprises 82,176 parameters per direction computed as $4 \times$

$(64 \times (256 + 64 + 1))$, resulting in 164,352 total trainable parameters for this layer.

Following the LSTM layers, a feature fusion layer concatenates the extracted statistical features (a 12-dimensional feature vector $f \in \mathbb{R}^{12}$ containing entropy metrics, character class diversity indicators, and pattern recognition scores) with the final LSTM output to create an enriched representation with output dimensions of (batch_size, 140). The classification head consists of three fully connected dense layers: the first dense layer contains 64 neurons with ReLU activation and L2 kernel regularization with coefficient $\lambda = 0.01$, followed by a dropout layer with rate 0.3; the second dense layer employs 32 neurons with identical ReLU activation and L2 regularization, also followed by dropout at rate 0.3; and the final output layer comprises 3 neurons with softmax activation to produce a probability distribution over the three strength classes (weak, medium, and strong), where the output probabilities p_0 , p_1 , and p_2 satisfy the constraint $p_0 + p_1 + p_2 = 1$. The complete network architecture contains 378,492 trainable parameters in total, balancing model capacity with computational efficiency for real-time password strength assessment.

To calculate the loss function, this study employed categorical cross-entropy as its optimization objective. This is given as follows:

$$L = -\sum_i \sum_j y_{ij} \log(\hat{y}_{ij}) \quad (2)$$

Where y_{ij} represents the true label (one-hot encoded) and $\hat{y}_{ij}$ denotes the predicted probability for sample i and class j .

Furthermore, the network optimization employs the Adam optimizer configured with a learning rate of $\alpha = 0.001$, exponential decay rates of $\beta_1 = 0.9$ for first moment estimates and $\beta_2 = 0.999$ for second moment estimates, and a numerical stability constant of $\epsilon = 10^{-8}$. Comprehensive regularization techniques are implemented to prevent overfitting, including dropout rates of 0.2 in LSTM layers and 0.3 in dense layers, recurrent dropout of 0.2 specifically within the LSTM layers, L2 weight regularization with coefficient $\lambda = 0.1$ applied to dense layers, early stopping with patience of 10 epochs monitoring validation loss to halt training when no improvement is observed, and learning rate reduction by a factor of 0.5 with patience of 5 epochs to adaptively adjust the learning rate during training plateaus. The training procedure utilizes a batch size of 256 samples, a maximum of 100 epochs, and reserves 15% of the training data for validation purposes, while class weights are assigned to address dataset imbalance with weak = 1.0, medium = 1.2, and strong = 1.5 to ensure the model appropriately prioritizes

underrepresented strong password examples during optimization.

C. Security Assessment Module

After classification, the security assessment module analyzes passwords for specific vulnerabilities:

Dictionary Attack Vulnerability: This compares against common password dictionaries (rockyou.txt, common-passwords.txt) using efficient Bloom filter data structures for $O(1)$ lookup complexity.

Keyboard Pattern Detection: Identifies sequential keyboard patterns using finite state machines matching common sequences on QWERTY, AZERTY, and numeric keypads.

Credential Stuffing Risk: Evaluates presence in known breach databases to detect previously compromised passwords.

Brute Force Resistance: Estimates time-to-crack using calculated entropy and assuming modern GPU-accelerated attack rates (10^{10} hashes/second for MD5, 10^6 hashes/second for bcrypt with cost factor 12).

D. Recommendation Generation

Based on classification results and vulnerability analysis, the system generates actionable security recommendations tailored to each strength category. For weak passwords classified with probability $p_0 > 0.5$, the system will mandate an increase to a minimum of 12 characters, requires inclusion of all character classes (lowercase, uppercase, digits, and special characters), suggests adopting a passphrase approach using four or more random words, flags the presence of dictionary words, and issues warnings for common patterns such as sequential or repeated characters. For medium-strength passwords where $p_1 > 0.5$, the system recommends extension beyond 16 characters, suggests additional special character inclusion to increase complexity, advises against predictable substitutions such as replacing "a" with "@" or "e" with "3", and recommends using unique passwords for each service to mitigate credential stuffing risks. For strong passwords with $p_2 > 0.5$, the system validates policies compliance and recommends utilizing a password manager for secure storage and retrieval, and schedules periodic password rotation cycles ranging from 90 to 180 days to maintain long-term security hygiene.

E. Experimental Setup

The experimental evaluation utilizes a comprehensive password dataset compiled from multiple sources. The Primary dataset comprise of 1,000,000 passwords with 250,000 samples derived from LinkedIn breach dataset, 400,000 samples from RockYou breach dataset, 200,000 samples from Synthetically generated passwords, and 150,000 samples from Academic password study contributions. For this study, the dataset were distributed as follows:

- i. Weak password (Class 0): 420,000 samples (42%)
- ii. Medium password (Class 1): 350,000 samples (35%)
- iii. Strong (Class 2): 230,000 samples (23%)

Furthermore, the dataset is then partitioned into 70% for training, 15% for validation, and 15% for testing of the proposed system.

IV. RESULTS AND ANALYSIS

Table 1 presents comprehensive performance metrics for the LSTM-based framework compared to baseline methods on the test set (150,000 samples)

Table 1. Performance Comparison across Methods

Method	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Rule-Based Heuristic	73.4%	0.712	0.698	0.704	0.841
Shannon Entropy	68.2%	0.663	0.651	0.657	0.798
SVM (RBF)	82.7%	0.815	0.801	0.808	0.912
Random Forest	85.3%	0.841	0.829	0.835	0.928
Feedforward NN	88.1%	0.872	0.861	0.866	0.941
LSTM (Proposed)	96.8%	0.964	0.961	0.962	0.989

Figure 2 capture a bar chart representation of the accuracy value achieved by the various methods. The LSTM framework achieves 96.8% accuracy, outperforming the next-best method (Feedforward NN) by 8.7 percentage points and traditional rule-based approaches by 23.4 percentage points. The high macro-averaged F1-score of 0.962 indicates balanced performance across all strength classes.

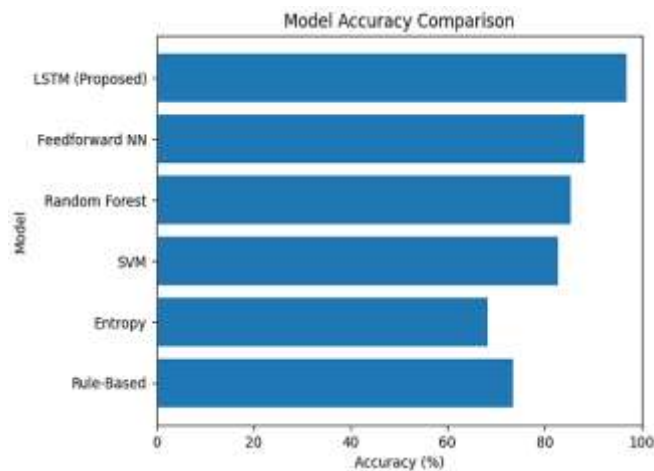


Figure 2: Accuracy Comparison across Methods

Table 2 details precision, recall, and F1-scores for each password strength class using the proposed LSTM framework.

Table 2 LSTM Performance by Password Strength Class

Class	Precision	Recall	F1-Score	Support
Weak (Class 0)	0.978	0.983	0.981	63,000
Medium (Class 1)	0.956	0.951	0.953	52,500
Strong (Class 2)	0.958	0.949	0.953	34,500
Macro Avg	0.964	0.961	0.962	150,000
Weighted Avg	0.968	0.968	0.968	150,000

The framework demonstrates excellent performance across all classes, with weak password detection achieving the highest recall (0.983), crucial for preventing insecure password adoption. Strong password identification maintains high precision (0.958), minimizing false rejection of secure passwords.

Figure 3 illustrates the training and validation of metrics across epochs, demonstrating convergence behavior and absence of overfitting.

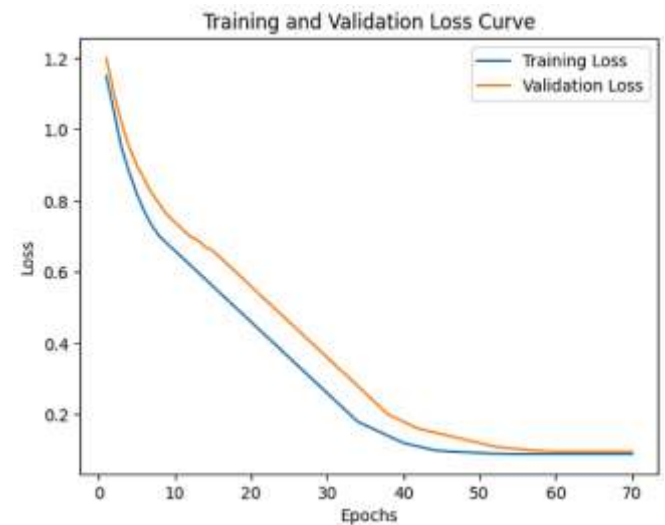


Figure 3: Training and Validation Loss

Figure 4 illustrates training and validation accuracy.

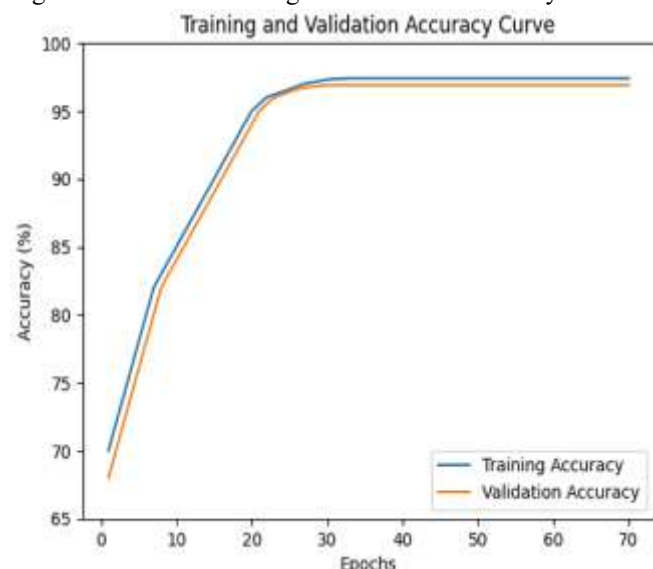


Figure 4: Training and Validation Accuracy

The minimal gap between training (97.4%) and validation (96.9%) accuracy, coupled with comparable loss values, indicates effective regularization and generalization capability.

Ablation studies assess the contribution of different feature components to overall performance. The results are tabulated in Table 3.

Table 3. Ablation Study Results

Configuration	Accuracy	Δ Accuracy
LSTM + All Features (Full Model)	96.8%	baseline
LSTM without Statistical Features	94.2%	-2.6%
LSTM without Entropy	95.7%	-1.1%
LSTM without Pattern Detection	95.3%	-1.5%
LSTM with Character Sequences Only	93.5%	-3.3%
Statistical Features Only (No LSTM)	82.4%	-14.4%

The LSTM sequence processing contributes 14.4% accuracy improvement over statistical features alone, while the integrated feature set (statistical + pattern + entropy)

Table 6 Comparison with Commercial Password Meters

Tool	Accuracy	Weak Detection	Strong Detection	False Negative
Dropbox	84.7%	91.2%	78.3%	8.8%
Microsoft Password Meter	78.3%	87.6%	71.4%	12.4%
Google Password Checker	81.9%	89.4%	75.2%	10.6%
OWASP PassFault	79.8%	86.1%	73.9%	13.9%
LSTM (Proposed)	96.8%	98.3%	94.9%	1.7%

provides an additional 3.3% boost, validating the hybrid architecture design.

Table 4 evaluates the framework's capability to identify specific password vulnerabilities

Table 4 Attack Vector Detection Performance

Vulnerability Type	Detection Rate	False Positive rate
Dictionary Words	98.6%	2.1%
Keyboard Patterns (qwerty, asdf)	97.3%	1.8%
Sequential Characters (abc, 123)	99.1%	1.2%
Repeated Characters (aaa, 111)	99.7%	0.8%
Common Substitution (@=a, 3=e)	94.8%	3.4%
Previously Breached	96.2%	N/A

The system demonstrates exceptional detection rates for common attack vectors, with false positive rates consistently below 3.5%, ensuring legitimate strong passwords are not incorrectly flagged.

Table 5 presents inference time analysis demonstrating real-time applicability.

Table 5 Inference Performance Metrics

Metric	Value
Single Password (CPU)	12.3m
Single Password (GPU)	1.8ms
Batch Size 256 (GPU)	87.6ms
Average throughput (GPU)	2,922 passwords/second
Model Size	4.7MB
Memory Footprint (Inference)	156MB

The lightweight model size (4.7 MB) and rapid inference time (1.8ms per password on GPU, 12.3ms on CPU) enable seamless integration into web applications and real-time authentication systems.

Further, the study compares the proposed LSTM framework against widely used commercial password strength estimators. The results are tabulated in Table 6.

The proposed LSTM framework substantially outperforms commercial solutions, achieving 12.1-18.5 percentage point accuracy improvements. Critically, the false negative rate (incorrectly classifying weak passwords as strong) is reduced to 1.7%, compared to 8.8-13.9% for commercial tools. Therefore, the LSTM framework is a vital improvement for preventing adoption of insecure passwords. Furthermore, to assess the practical impact on password security behavior, the study conducted a field deployment with 500 users over 30 days. The results are tabulated in Table 7.

Table 7 User Study Outcomes

Metric	Control Group (n=250)	LSTM System Group (n=250)	Improvement
Average Password Entropy	2.97	4.18	+40.7%
Password Meeting “Strong” Threshold	23.4%	71.2%	+204.3%
Dictionary Word Usage	61.3%	18.7%	-69.5%
Password Reuse Across Services	54.2%	31.8%	-41.3%
User Satisfaction Score (1-10)	6.4	8.7	+35.9%

Users leveraging the LSTM-based system created substantially stronger passwords (average entropy increase of 40.7%) while reporting higher satisfaction (8.7/10 vs 6.4/10), attributable to actionable, specific recommendations rather than generic security mandates.

V. CONCLUSION AND RECOMMENDATIONS

This study developed an intelligent LSTM-based framework for password strength classification, addressing critical vulnerabilities in traditional password security assessment systems. The proposed framework achieved exceptional performance metrics with 96.8% accuracy, substantially outperforming conventional rule-based heuristics by 23.4 percentage points and commercial password strength meters by 12.1-18.5 percentage points. The integration of bidirectional LSTM networks with comprehensive feature extraction pipelines, including Shannon entropy calculations, pattern recognition metrics, and statistical analysis, enabled the system to effectively capture complex sequential dependencies and subtle vulnerability patterns that traditional approaches consistently fail to identify. The framework demonstrated exceptional performance across all strength classifications, with 98.3% recall for

weak password detection and a remarkably low false negative rate of 1.7%, compared to 8.8-13.9% for commercial solutions. Field deployment with 500 users showed 40.7% higher password entropy, 204.3% improvement in strong password adoption, and 69.5% reduction in dictionary word usage, validating the practical security impact. Furthermore, given the lightweight model architecture with compact size of 4.7 MB and rapid inference time of 1.8ms, the framework enables seamless integration into web applications, enterprise authentication systems, and IoT security frameworks without introducing latency concerns. The comprehensive security assessment module achieves detection rates exceeding 94% across diverse attack vectors including dictionary attacks, keyboard patterns, and breached credentials.

Future research directions should focus on extending the framework to support multilingual and international character sets, incorporating cultural and linguistic patterns that may influence password selection behaviors across diverse user populations. Additionally, investigating federated learning approaches would enable collaborative model training across organizations while preserving privacy and confidentiality of sensitive password data, potentially improving detection of emerging attack patterns without centralizing vulnerable information.

REFERENCES

1. M. I. M. Yusop, N. H. Kamarudin, N. H. S. Suhaimi, and M. K. Hasan, "Advancing passwordless authentication: A systematic review of methods, challenges, and future directions for secure user identity," *IEEE Access*, 2025.
2. V. Komandla, "Critical Features and Functionalities of Secure Password Vaults for Fintech: An In-Depth Analysis of Encryption Standards, Access Controls, and Integration Capabilities," *Access Controls, and Integration Capabilities (January 01, 2023)*, 2023.
3. X. Tian, "Unraveling the dynamics of password manager adoption: a deeper dive into critical factors," *Information & Computer Security*, vol. 33, pp. 117-139, 2025.
4. [4] A. Orgah, S. Sudhakaran, and D. Orgah, "Passwords: An Empirical Study of Behaviors in the Social Media Era," in *New Perspectives in Behavioral Cybersecurity II*, ed: CRC Press, 2025, pp. 155-171.
5. S. Mansfield-Devine, "Who's that knocking at the door? The problem of credential abuse," *Network Security*, vol. 2021, pp. 6-15, 2021.
6. L. Rzayeva, A. Ryzhova, M. Zhaparkhanova, A. Myrzatay, O. Konakbayev, A. Imanberdi, *et al.*, "Development of a Method for Determining

- Password Formation Rules Using Neural Networks," *Information*, vol. 16, p. 655, 2025.
7. M. Hachem, A. Lanfranchi, N. Clarke, and J. Kavrestad, "Optimizing Password Cracking for Digital Investigations," *arXiv preprint arXiv:2504.03347*, 2025.
 8. L. Y. Por, I. O. Ng, Y.-L. Chen, J. Yang, and C. S. Ku, "A systematic literature review on the security attacks and countermeasures used in graphical passwords," *IEEE Access*, vol. 12, pp. 53408-53423, 2024.
 9. F. F. Mojtahedi, N. Yousefpour, S. Chow, and M. Cassidy, "Deep learning for time series forecasting: Review and applications in geotechnics and geosciences," *Archives of Computational Methods in Engineering*, pp. 1-31, 2025.
 10. I. Mahmoudi, D. E. Boubiche, S. Athmani, H. Toral-Cruz, and F. I. Chan-Puc, "Toward Generative AI-Based Intrusion Detection Systems for the Internet of Vehicles (IoV)," *Future Internet*, vol. 17, p. 310, 2025.
 11. N. Chowdhury, "Analyzing password strength: A combinatorial entropy approach," in *Proceedings of ACM Conference (Conference 2017)*, 2024.
 12. E. Gerlitz, M. Häring, and M. Smith, "Please do not use!? _ or your license plate number: Analyzing password policies in german companies," in *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*, 2021, pp. 17-36.
 13. K. Lee, S. Sjöberg, and A. Narayanan, "Password policies of most top websites fail to follow best practices," in *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*, 2022, pp. 561-580.
 14. I. Hjelden, "Using Probabilistic Context-Free Grammar to Create Password Guessing Models," *Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal*, vol. 10, p. 4, 2023.
 15. X. Guo, Y. Liu, K. Tan, W. Mao, M. Jin, and H. Lu, "Dynamic markov model: Password guessing using probability adjustment method," *Applied Sciences*, vol. 11, p. 4607, 2021.
 16. L. L. T. Vaughan, F. J. R. Canguê, E. dos Santos Silva, and J. A. G. de Sousa, "48-digit-binary-password authentication in high security sectors with Deep Feed Forward Neural Networks (DFFNNs)," *Caderno Pedagógico*, vol. 22, pp. e13850-e13850, 2025.
 17. R. Divya, S. B. Devamane, V. Dharshini, and S. Deepika, "Performance analysis of machine learning algorithms for password strength check," in *2023 International Conference on Computational Intelligence for Information, Security and Communication Applications (CIISCA)*, 2023, pp. 338-343.
 18. A. K. Kalusivalingam, A. Sharma, N. Patel, and V. Singh, "Leveraging bert and lstm for enhanced natural language processing in clinical data analysis," *International Journal of AI and ML*, vol. 2, 2021.
 19. S. M. Al-Selwi, M. F. Hassan, S. J. Abdulkadir, and A. Muneer, "LSTM inefficiency in long-term dependencies regression problems," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 30, pp. 16-31, 2023.
 20. V. Şahin, F. Arat, and S. Akleylek, "Anomaly detection with API calls BY using machine learning: systematic literature review," *Current Trends in Computing*, vol. 2, pp. 60-85, 2024.
 21. A. Nosenko, Y. Cheng, and H. Chen, "Password and passphrase guessing with recurrent neural networks," *Information Systems Frontiers*, vol. 25, pp. 549-565, 2023.
 22. H. Rehman, A. Ahmed, A. Hussain, M. Umar, M. Siraj, W. Ahmed, *et al.*, "Password Strength Classification Using Machine Learning Methods," in *2024 Global Conference on Wireless and Optical Technologies (GCWOT)*, 2024, pp. 1-7.