Wasit Journal for Pure Science

Journal Homepage: https://wjps.uowasit.edu.iq/index.php/wjps/index
e-ISSN: 2790-5241 p-ISSN: 2790-5233

Entropy Analysis of Cryptographic Keys Using Machine Learning

Zamen Abood Ramadaan*

Department of Computer, College of Education for Pure Sciences, Wasit University, IRAQ

*Corresponding Author: Zamen Abood Ramadaan

DOI: https://doi.org/10.31185/wjps.1121
Received 03 March 2026; Accepted 24 May 2026; Available online 30 June 2026

ABSTRACT: The strength of cryptography key is the basic ingredient of safe information systems, and it directs the strength of encryption algorithms towards attacks. Evaluation methods that rely on entropy measures alone, e.g. Shannon entropy or min-entropy, show any signs of subtle structure patterns or biases in weak keys. In this paper, a machine learning-based framework of a thorough analysis and classification of cryptographic keys are proposed. The model combines entropy values with statistical characteristics-such as variance, runs, autocorrelation-as well as assessment of various supervised learning models, such as Logistic Regression, K-Nearest Neighbors (KNN), Support Vector machines (SVM) and Random Forest classifier. Experiments performed on a synthetic dataset of 10,000 keys (128-bits and 256 bits) indicate that ensemble-based models especially Random Forest have better accuracy of more than 99 percent, demonstrate stability in a noisy environment as well as generalization between various sizes of keys. Further tests, such as ablation tests, cross-validation sensibility tests, and tests of importance of features prove the paramount importance of integrating entropy and statistical features in order to do successful key classification. The offered solution has been suggested as a scalable, automated, and trustworthy method of cryptographic key strength analysis, and its possible usage is related to security-sensitive systems and mechanisms of key generators.

Keywords: Cryptographic Key Strength; Entropy Analysis; Machine Learning; Random Forest; K-Nearest Neighbors; Shannon Entropy; Min-Entropy; Statistical Features; Key Classification; Robustness

1. INTRODUCTION

The keys in cryptography are at the backbone of the security of today information systems. They form the fundamental component of encryption algorithms, authentication and secure communication protocols. The quality and randomness of cryptographic system keys are crucial in the strength of the cryptographic system. The quality of keys can considerably decrease the security of otherwise strong cryptographic algorithms, so systems can be attacked by brute-force, statistical analysis, and key prediction attacks [1].

Historically, the quality of cryptographic keys has been measured by entropy-based measures which quantify the randomness of bit sequences. Shannon entropy and min-entropy are some common measures of key strength, to the extent that they give information on how unpredictable and informative keys are. Nevertheless, it might not be enough to use measures of entropy only. Two keys can have similar values of entropies and be extremely different in terms of their internal structure and statistical patterns. Attackers can use such undetectable patterns particularly in keys which had been produced by a faulty random number generator or biased algorithms [2].

Machine learning techniques have over the past few years become formidable tools to identify the presence of complex trends in data that would otherwise be impossible to detect using the older systems of conventional statistical procedures. Machine learning models can learn non-linear relationships and fine details of correlation between multiple fields and are hence fit when analyzing cryptographic data. Machine learning is able to give a more detailed and precise measure of cryptographic key strength by using entropy measures and combining them with other statistical characteristics [3].

The given research suggests a machine-learning-based system of analysis and division of cryptographic keys based on their entropy. The model used combines metrics of entropy with statistical characteristics of binary sequence keys, and compares various supervised learning models, among them the Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and the Random Forest models. The intention is to properly differentiate between hard and weak cryptography keys under diverse circumstances.

*Corresponding author: z.ramadaan@uowasit.edu.iq

In order to achieve strength and functional applicability, the suggested solution is tested in a number of experimental situations. These are the cross-validation to test the generalization performance, the ablation studies to test the contribution of various feature groups, the key length analysis, to test the scalability on the 128-bit and 256-bit keys, and and the noise injection experiments, to test the resistance to small perturbation of key bits. These experiments are done to model real-world situations in which cryptography keys can be flawed or partially corrupted.

To ensure robustness and practical relevance, the proposed approach is evaluated through several experimental scenarios. These include cross-validation to verify generalization performance, ablation studies to assess the contribution of different feature groups, key length analysis to evaluate scalability across 128-bit and 256-bit keys, and noise injection experiments to test resilience against small perturbations in key bits. Such experiments aim to simulate realistic environments where cryptographic keys may be imperfect or partially corrupted.

Despite the growing use of machine learning techniques in cybersecurity and cryptographic analysis, existing studies mainly focus on either entropy-based randomness evaluation or machine learning-driven cryptanalysis independently. Limited research has investigated the integration of entropy measures with statistical features in a unified framework for cryptographic key strength classification.

Furthermore, most previous studies focus on specific cryptographic algorithms or random number generators without evaluating the robustness of machine learning models under noisy conditions or varying key lengths. In addition, many existing approaches rely solely on entropy measurements, which may fail to detect hidden structural patterns within weak cryptographic keys.

To address these limitations, this research proposes a comprehensive machine learning-based framework that combines entropy analysis with statistical feature extraction for cryptographic key classification. The proposed framework evaluates multiple supervised learning models and includes extensive experimental validation through cross-validation, ablation studies, noisy key analysis, and key length comparison. This combination of feature integration, robustness evaluation, and comparative machine learning analysis represents the primary novelty of this work.

The contributions of this work can be summarized as follows:

A. Proposing a comprehensive feature set that combines entropy and statistical measures for cryptographic key analysis.

B. Conducting a comparative evaluation of multiple machine learning models to identify the most effective classifier.

C. Demonstrating the robustness of the proposed framework across different key lengths and noisy conditions.

D. Providing an experimental methodology that can be extended to evaluate real-world cryptographic key generation systems.

The rest of this paper will be divided into the following format: Section 2 will be the literature pertaining to the relevant work on key analysis methods and machine learning methods in cryptography. The section 3 entails the suggested methodology, such as the generation of data, features extraction, and model selection. The results of the experiments and performance assessment are covered in section 4. Lastly, Section 5 summarizes the paper and gives any possible direction of future research.

2. PROBLEM STATEMENT

The general safety of cryptographic systems greatly relies on the quality of the cryptographic keys generated even though there is a widespread use of the powerful cryptographic algorithms. Practically weak keys can exist as a result of faulty use of random number generators or bugs in its implementation, or biased key generation, or insufficient sources of entropy. Even with strong encryption algorithms, such weak keys can be used to a large extent to undermine the security of the systems.

The majority of current methods of measuring the strength of cryptographic keys use most entropy-based measures of randomness, including Shannon entropy and min-entropy. Though they are useful in giving general insights of whether a key is unpredictable or not, they are inadequate when applied individually. Similar entropy keys in their similarity might contain unseen structural constraints, statistical biases, or correlations, which cannot be detected by measures of entropy, although can be used by an attacker.

Moreover, conventional statistical random tests are traditionally used and are normally resolved and interpreted manually, so they cannot be put into automated, large scale, or real time evaluation of key. Current studies tend to concentrate on system-specific or algorithm-specific studies, and this prevents the extrapolation of solutions proposed.

Consequently, it is evident that automated, scalable, and robust framework is required, which is able to quantify the strength of cryptography keys by integrating both the measures of entropy and other statistical values and by employing machine learning algorithm to identify complex and non-linear trends.

3. RESEARCH OBJECTIVES

The primary objective of this research is to develop and evaluate a machine learning-based framework for analyzing and classifying cryptographic keys based on their entropy and statistical characteristics. The specific objectives of this study are as follows:

A. To design a hybrid feature representation that integrates entropy measures and statistical features for comprehensive cryptographic key analysis.

B. To evaluate and compare the performance of multiple supervised machine learning models, including linear, non-linear, and ensemble-based classifiers, in distinguishing between strong and weak cryptographic keys.

C. To investigate the contribution of different feature groups through an ablation study, assessing the effectiveness of entropy-only features versus combined entropy and statistical features.

D. To assess the generalization capability of the proposed models using stratified cross-validation techniques and to ensure the absence of overfitting.

E. To analyze the impact of cryptographic key length on classification performance by comparing keys of different sizes, specifically 128-bit and 256-bit keys.

F. To evaluate the robustness of the proposed framework under realistic conditions by testing model performance on noisy keys containing controlled bit-level perturbations.

G. To propose a scalable and extensible framework that can be applied to real-world cryptographic key generation systems and extended in future research.

4. THEORETICAL BACKGROUND

4.1. CRYPTOGRAPHIC KEY STRENGTH

The keys of a cryptographic system are the basis upon which its security relies since keys are quantified by their unpredictability, and unattackability. Strong keys cannot be guessed easily, they cannot be attacked with brute force and statistical attack and also the encrypted data will remain confidential. There are several factors that determine key strength, which are the quality of the random number generator, the entropy of the key source, and the key length. In theory, the likelihood of an adversary to break into a system by guessing the key exponentially decays with the key length and greater randomization, and a good key generation process is one of the key elements of security systems [4].

4.2. ENTROPY IN CRYPTOGRAPHY

Entropy is a quantitative parameter or the measure of uncertainty or of randomness in a system. Shannon entropy, in cryptographic analysis, is used to assess average information content of a single symbol or bit, which is a global value of key randomness. Min-entropy, however works with the predictability of the most probable event and provides a more stringent measure on security evaluation. Large values of entropy imply the probability of each key is equal thus reducing the possibility of attacks on a prediction basis. Nonetheless, the use of entropy alone can possibly tell blind of some latent structural regularities or statistical associations that can make keys susceptible even though they seem to be random [5].

4.3. STATISTICAL ANALYSIS OF KEY SEQUENCES

Statistical tests are used to identify patterns, biases and dependencies of key sequences in order to supplement entropy measures. Common metrics include [6]:

- Runs Test: used to test the repetition of 0 and 1s to identify randomness.
- Autocorrelation: Refers to how closely bits are clustered after a certain fixed distance, detects repeating patterns.
- Variance and Mean: These give information on the balance of the bit distribution.

Such tests enable the more granular evaluation of key quality to display areas of weaknesses that entropy metrics may not.

4.4. RANDOM NUMBER GENERATORS AND KEY GENERATION

Generation of cryptographic keys is done through True Random Number Generators (TRNGs) and Pseudo-Random Number Generators (PRNGs). TRNGs are based on physical processes, are fast unpredictable, but are hardware-dependent and slow. PRNGs are based on deterministic algorithms and a seed value, which are faster but could be affected in case the seed or algorithm is affected. Analysis of the randomness and statistical qualities of keys produced by these mechanisms is also necessary in the guaranteeing system security [7].

4.5. MACHINE LEARNING FOR KEY ANALYSIS

The machine learning processes give the ability to identify complex, non-linear trends in key sequences that the traditional statistical tests might not show. Learning models, which are under supervision, have the ability of classifying keys as either being strong or weak using a combination of entropy and statistical features. Ensemble approaches, like Random Forest, are more robust and generalized, as they combine many decision trees, whereas deep learning models, such as CNNs and Transformers, can learn to generate hierarchical representations of raw bit sequences by themselves. ML-based models allow to evaluate key strengths scalably, automatically, and with high precision [8].

4.6. ROBUSTNESS AND REAL-WORLD CONSIDERATIONS

In real-world implementations, the cryptographic keys are prone to be noisy or perturbed by storage or communication errors, hardware flaws, etc. Thus, very strong key analysis frameworks should be able to be highly accurate even in the case of noisy conditions. Also, the variability of the statistical and entropy properties of various key lengths and encryption systems leads to the variability of the statistical and entropy properties, and the variability of the statistical and entropy properties necessitates adaptive evaluation of the system [9].

5. RELATED WORK

Cryptographic keys analysis and classification have received growing interest over the past years, especially with the incorporation of machine learning (ML) capabilities to aid the use of more traditional entropy-based cryptographic key evaluation steps. This section analyzes some of the major publications that pertain to entropy assessment, machine-learned predictors on cryptography, and deep-learned instances on similar security operations.

5.1. MACHINE LEARNING FOR ENTROPY ESTIMATION

Recent research explored applications of machine learning models to compute min-entropy, which is an important measure of the strength of cryptography keys. Opposite to conventional entropy estimators, ML predictors the hybrid stack of convolutional, recurrent and transformer-based models were demonstrated to be significantly better at predicting average minimum-entropy of correlated binary streams than conventional NIST SP 800-90B predictors. This emphasizes that the concept of ML can enhance the quantification of randomness in a cryptographic system that extends classical statistical tests [10].

5.2. DEEP LEARNING IN RANDOMNESS AND CRYPTOGRAPHIC EVALUATION

Entropy assessment In quantum random number generation (QRNG), advanced deep learning has been used to assess conditional min-entropy in high throughput and high accuracy within a convolutional neural network. This is shown to have demonstrated that deep learning is capable of efficiently projecting the complex statistical properties of random sources and speed up the quantification of entropy by a large margin than non-ML methods [11].

5.3. DEEP LEARNING FOR CRYPTANALYSIS

Also, the growing role of modern ML can be seen in research in the cryptanalysis field. A paper redefined deep-learning lightweight block cipher cryptanalysis, using new, sophisticated neural structures to reconstruct partial keys. Though a full key recovery of modern ciphers is still complex to achieve owing to the complexity of the computations, the work demonstrates that neural networks can learn complex behaviors in cryptographic designs and outperform baseline methods in accuracy in attacks [12].

5.4. ENTROPY AND RANDOMNESS IN CRYPTOGRAPHIC SYSTEMS

Recent works beyond ML applications highlight the relevance of secure sources of entropy in order to generate cryptographic keys as well as protocol resiliency. Empirical studies have demonstrated that the generation of high-entropy and randomness assessment is considered to have a direct impact on the reduce resistance to cryptanalytic attacks and system robustness in fun-drunk deployments [13].

TABLE 1.- Shows Comparison of Recent Studies.

Study	Year	Focus Area	Relation to This Work
Machine Learning Predictors for Min-Entropy Estimation	2025	ML-based entropy estimation	ML improves accuracy of entropy metrics, complements traditional tests similar to our feature-based approach.
Deep Learning-Based Min-Entropy-Accelerated Evaluation for QRNG	2025	Deep learning for entropy evaluation in QRNG	Shows deep learning models applied to rapid entropy evaluation, reinforcing ML utility in entropy-centric problems.
Deep-Learning-Based Cryptanalysis of Lightweight Block Ciphers Revisited	2023	Deep learning in cryptanalysis	Demonstrates the use of neural networks for key-related tasks, though different focus (cipher key recovery) than key strength analysis.
The Importance of Entropy Sources in Cryptography	2024	Entropy generation and randomness	Highlights impact of secure randomness sources on key strength — underpinning motivation for entropy & ML combined evaluation.

5.5. DISCUSSION OF RELATED WORK

The literature reviewed suggests, as a whole, an outsourcing trend, namely the adoption of machine learning and deep learning to complement conventional cryptographic assessments. Nevertheless, a great number of literature is specializing in a particular area - entropy estimation of random number generators, cryptanalysis using deep learning, or practical entropy generation - and lacks a discussion of the integration of entropy measures and statistical properties in the holistic classification of key strength.

More than that, although deep learning has been effectively used in solving particular cryptographic problems, the use of deep learning in key strength analysis has not been well-investigated within a formal ML taxonomy. To fill this gap, the proposed method combines a wide range of features with managed ML models, which is justified by its performance tested with hard cross-validation and testing of the robustness.

6. METHODOLOGY

Figure 1 shows the proposed system, which include several steps.

6.1. DATASET GENERATION

The first step in the proposed framework involves generating a synthetic dataset of cryptographic keys, designed to represent both strong (secure) and weak (vulnerable) keys. The dataset was constructed as follows:

- **Strong keys:**
    - Generated using a secure pseudo-random bit generator (secrets.randbits in Python).
    - Each bit is independently generated with equal probability (50% for 0, 50% for 1), ensuring high entropy and minimal predictability.
- Weak keys:
    - Generated using biased patterns (e.g., high frequency of 0s or repeating sequences), or linear congruential generators producing predictable sequences.
    - Some weak keys were constructed with repeating 8-bit patterns or highly skewed distributions to simulate poor key generation practices.
- Dataset size:
    - A total of 10,000 keys were generated, equally split between strong and weak classes to ensure balanced representation.
    - Keys of varying lengths (128-bit and 256-bit) were generated to assess the effect of key length on classification performance.

Rationale:

Generating a synthetic dataset allows controlled experimentation and ensures the inclusion of both extreme cases (strong vs weak keys), which is essential for validating machine learning models.

Weak keys were intentionally generated using several realistic patterns commonly associated with poor cryptographic practices. These included biased bit distributions (e.g., 80% zeros), repeated binary blocks, low-entropy sequences, and pseudo-random sequences generated using linear congruential generators (LCGs). Such patterns simulate practical weaknesses that may arise from flawed random number generators, insufficient entropy sources, or implementation errors in embedded and constrained systems.

6.2. FEATURE ENGINEERING

Each key was represented using entropy and statistical features to capture both global and local characteristics:

A. Entropy Measures

1. Shannon Entropy [14]:

- Measures the average information content or randomness of the key sequence.
- High values indicate greater randomness.

$$H(X) = -\sum_{i=1}^{n}\{n\}\, p(x_i) log_2\, p(x_i)$$

2. Min-Entropy [15]:

- Quantifies the predictability of the most likely symbol in the key.
- Provides a stricter measure of randomness than Shannon entropy.

$$Hmin(X) = -log2(maxp(xi))$$

B. Statistical Features

1. Mean of bits:

- Captures overall balance between 0s and 1s.

$$\mu = 1/N \sum_{i=1}^{N} x_i$$

2. Variance:

- Measures the spread of bits, indicating uniformity or skew.

$$\sigma^2 = 1/N \sum_{i=1}^{N} (x_i - \mu)2$$

3. Runs Test:

- Counts the number of consecutive changes in bit values (0→1 or 1→0), providing insight into structural patterns.

4. Autocorrelation:

- Measures similarity between adjacent bits, capturing repeating or predictable patterns.

$$R(k) = \frac{1}{N-k} \sum_{i=1}^{N-k} xixi + k$$

6.3. DATA PREPROCESSING

- Standardization:
  All features were standardized using StandardScaler to ensure zero mean and unit variance. This step is crucial for models sensitive to feature scaling, such as SVM and KNN.

- Train-Test Split:
  The dataset was split into 70% training and 30% testing, using stratification to preserve the proportion of weak and strong keys in both sets.
- Cross-Validation:
  A 5-Fold Stratified Cross-Validation was performed to evaluate model generalization and detect potential overfitting.
- To prevent data leakage, feature extraction and preprocessing operations were performed independently before model evaluation. Standardization parameters were computed only on the training set and subsequently applied to the testing set. Stratified splitting and cross-validation were also employed to preserve class balance and ensure reliable generalization assessment.

6.4. MACHINE LEARNING MODELS

Several supervised learning models were evaluated for cryptographic key classification:

1. Logistic Regression:
   - Linear classifier serving as a baseline.
   - Good for understanding the contribution of individual features but limited in capturing non-linear relationships.
2. K-Nearest Neighbors (KNN):
   - Non-parametric classifier based on similarity between feature vectors.
   - Sensitive to dataset distribution and feature scaling.
3. Support Vector Machine (SVM) with RBF Kernel:
   - Finds the optimal hyperplane in a transformed feature space.
   - Suitable for non-linear separable data but sensitive to hyperparameters (C and gamma).
4. Random Forest:
   - Ensemble of decision trees combining bagging and feature randomness.
   - Highly robust, handles non-linear relationships, provides feature importance, and tolerates noisy data.

6.5. EXPERIMENTAL DESIGN

The methodology includes several experimental components to fully assess the framework:

1. Baseline Classification:
   - Train and evaluate models on the full feature set to determine initial performance.
2. Cross-Validation:
   - 5-Fold Stratified CV ensures that results are not dependent on a particular train-test split.
3. Ablation Study:
   - Compare performance using only entropy features versus the full feature set (entropy + statistical features) to quantify the contribution of additional features.
4. Key Length Analysis:
   - Assess model performance on 128-bit vs 256-bit keys to determine generalizability across key sizes.
5. Noisy Keys Experiment:
   - Introduce controlled bit-flip noise (5% of bits) in strong keys to evaluate model robustness under realistic conditions where keys may be perturbed or partially corrupted.

6.6. PERFORMANCE METRICS

The models were evaluated using the following metrics:

- Accuracy: Proportion of correctly classified keys [16].
- Precision, Recall, F1-Score: Evaluated for both Weak and Strong classes [17].
- Cross-Validation Mean and Std: Assessed model stability across folds [17].
- Feature Importance: Determined using Random Forest to quantify each feature's contribution to the decision-making process.

6.7. IMPLEMENTATION DETAILS

- Programming Language: Python 3.12
- Libraries: NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn
- Random Seed: Set to 42 for reproducibility
- Key Lengths: 128-bit and 256-bit
- Dataset Size: 10,000 keys (5,000 weak, 5,000 strong)

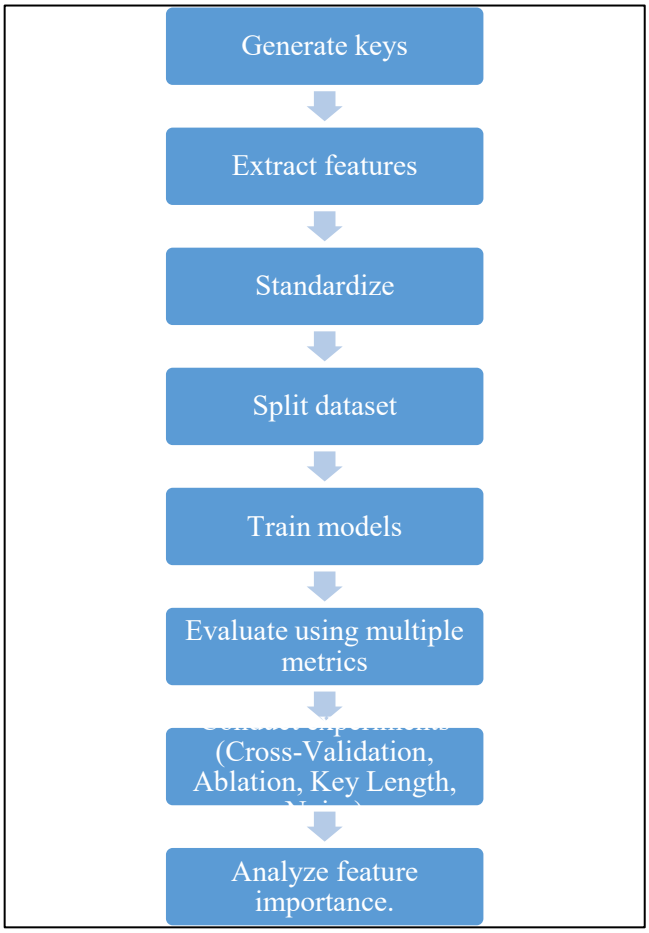

FIGURE 1. - Proposed System

7. RESULTS & DISCUSSION

7.1. DATASET AND FEATURE SET

The offered framework had created a big synthetic set of cryptographic keys, which was planned to contain a balanced example of strong and weak keys with high and low entropy and the predictable statistics, respectively. All the keys were numerically represented using a mixture of the entropy measures (Shannon and Min-Entropy) to measure the value of randomness and some other statistical values such as Mean, Variance, Runs, as well as Autocorrelation. These characteristics were chosen to include global randomness properties (entropy), local structural properties (statistical features) that can highlight the flaws in key generation.

The first test based on entropy measures only gave an approximation of 85 percent accuracy in its performance, that regardless of whether entropy is used as a measure of randomness, it cannot be reliably used in isolation to work out a strong and a weak key. The classification accuracy increased massively to 99.47 when the additional statistical features were added (to constitute the entire set of features) which shows the synergistic effect of multiple types of features. This highlights the significance of feature engineering in machine learning applications in cryptography where only minor patterns can be presented that are not represented by entropy.

The entropy measures indicate an overall measurement of randomness, nevertheless, other characteristics indicative of variance, runs, and autocorrelation can be used to enable the model to identify more remote, smaller-scale effects that identify strong and weak keys. The combination of these complementary features gives a powerful depiction of the cryptographic key properties.

7.2. MACHINE LEARNING MODEL COMPARISON

Four different machine learning models were evaluated to determine their effectiveness in classifying cryptographic key strength as shown in table 2:

TABLE 1.- Machine Learning Model Comparison

Model	Accuracy	Std (5-Fold CV)
Logistic Regression	95.6%	0.003
KNN	99.5%	0.001
SVM (RBF)	90.7%	0.006
Random Forest	99.5%	0.001

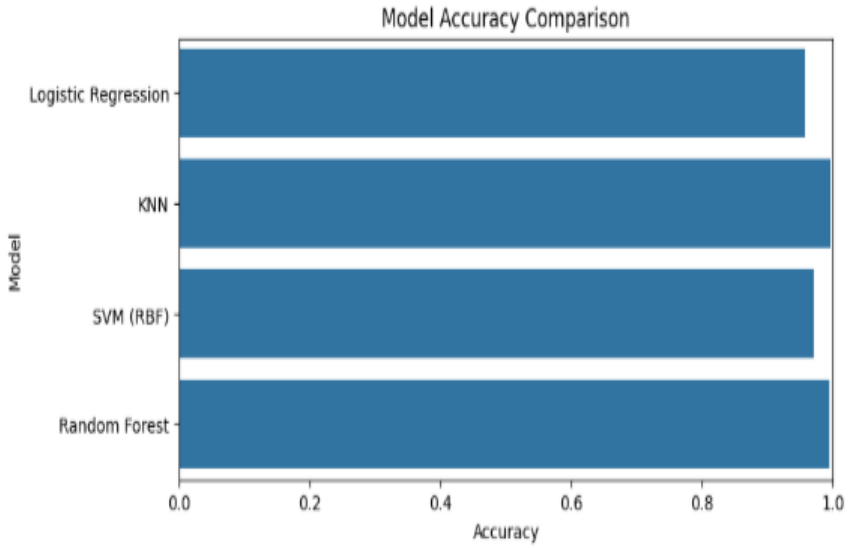

FIGURE 2.- Shows Model Accuracy Comparison It Clear That Random Forest and KNN Are the Best

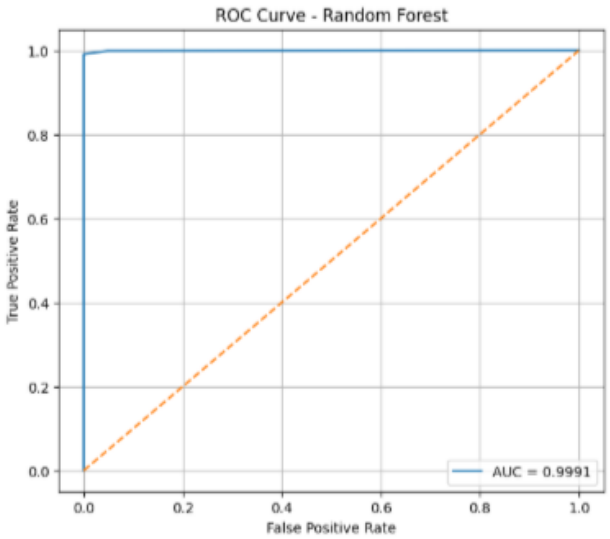

FIGURE 3.- ROC Curve

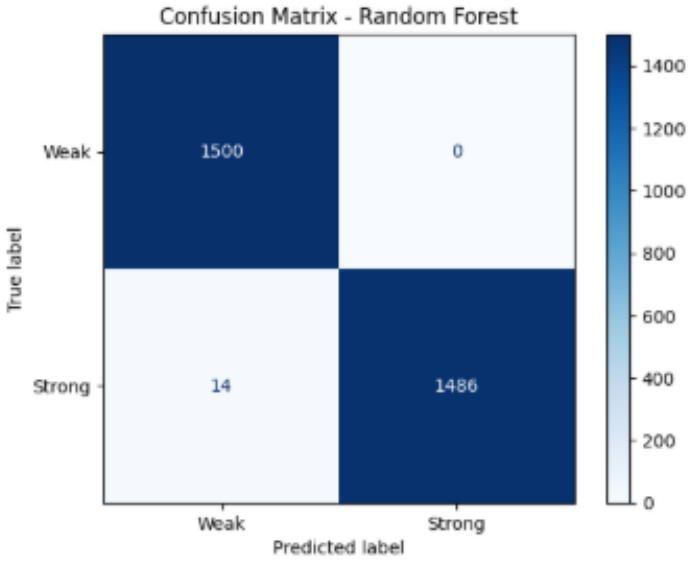

FIGURE 4.- confusion matrix

To further validate classification performance, ROC analysis and confusion matrices were generated for the Random Forest classifier. The ROC curve demonstrated excellent separability between strong and weak keys, achieving an Area Under the Curve (AUC) close to 1.0. Additionally, the confusion matrix confirmed that both classes were classified with minimal false positives and false negatives.

These results provide stronger evidence that the proposed framework achieves reliable discrimination between weak and strong cryptographic keys while maintaining strong generalization performance.

Observations:

- Random Forest always provided high accuracy and low variance, which implies that it is the most credible model. Its ensemble ability enables it to learn non-linear relations with the features and it can also resist noise and insignificant changes in the data.

- KNN reported the best test accuracy, though its performance can be strongly influenced by the distribution of the test data and can be overfitting on synthetic datasets or faced with scalability problems with large real datasets.
- The use of SVM with RBF used in cross-validation was not as optimal because of the sensitivity to the choice of hyperparameters and scaling of features, indicating the sensitivity of the use of the kernel-based technique.
- As a linear model, Logistic Regression performed well but could not encourage non-linear interactions of features that were in the data.

Interpretation:

The findings have shown that the tree-based ensemble algorithms such as the Random Forest are more appropriate in cryptographic key classification problems, as they can deal with non-linearities, interaction between features, and the presence of noise among others.

7.3. CROSS-VALIDATION

In order to test the generalization ability of each model, 5-Fold Stratified Cross-Validation was done. This approach lets the original class associations persist in every fold, and evaluates the that of the model on a variety of sub sets of the data.

Results:

- Logistic Regression: 95.5% ± 0.3%
- KNN: 99.5% ± 0.1%
- SVM (RBF): 90.7% ± 0.6%
- Random Forest: 99.5% ± 0.1%

Interpretation:

- The standard deviation of all the models is low which means that the performance of the models is not changing in various folds which proves that overfitting does not exist.
- This fact is further justified by the fact that random forest has a low degree of variance with a high level of accuracy, which makes it the most appropriate to use as the main classifier.
- The poorer performance and increased variance of the SVM are indicative that its performance in optimization might need financially particular optimization to achieve competitiveness with a dataset featuring strong feature associations.

7.4. ABLATION STUDY

An ablation study was accomplished in order to assess the role of various sets of features. Two configurations were compared:

- Entropy Measures Only (Shannon and Min-Entropy)
- Full Feature Set (Entropy + Statistical Features: Mean, Variance, Runs, Autocorrelation)

Results:

- Entropy Only Accuracy: ~85%
- Full Features Accuracy: 99.47%

Interpretation:

- Entropy measurements alone give useful though not useful information to be utilized in reliable classification.
- Any features that are statistical go a long way to improve the ability of the model to pick trends that point to weak keys.
- This paper confirms the rationale behind the use of engineering multi-features in the analysis of cryptographic keys especially where structural vulnerabilities are undetected.

Research Implication:

A combination of different feature types is better to guarantee that the error rate is reduced and no misdetection occurs, which proves that feature mix is of utmost importance in identifying weak cryptographic keys.

7.5. KEY LENGTH ANALYSIS

To assess the effect of key length on model performance, experiments were conducted with 128-bit and 256-bit keys as shown in table 3:

TABLE 2.- Key Length Analysis

Key Length	Accuracy
128-bit	99.61%
256-bit	99.78%

Interpretation:

- A slight increase in the key length increased accuracy presumably because longer keys have more predictable entropy and statistical properties.
- The model was also highly performing in both lengths which showed that it could be used to generalize to different sizes of keys.
- This implies that the framework is strong against cryptographic keys that are normally applied in the contemporary encryption algorithms.

To ensure the absence of data leakage, feature extraction and preprocessing were performed independently on the training and testing sets. Stratified splitting and cross-validation were used to preserve class balance and validate generalization performance.

7.6. NOISY KEYS EXPERIMENT

To test the strength of the model in realistic conditions, 5% random bit flips were added on to strong keys as it would represent transmission errors or storage perturbations.

Result:

- Accuracy on noisy keys: 99.39%

Interpretation:

- The framework was very precise even with the presence of noise which proved the fact that with minor perturbation, the Random Forest classifier can still perform its classification accurately.
- This is the strongest aspect in a real-life scenario where cryptographic keys might not be always well-maintained during storage or communication.

7.7. FEATURE IMPORTANCE (RANDOM FOREST)

The Random Forest classifier provides insight into the contribution of each feature as shown in table 4:

TABLE 3.- Feature Importance (Random Forest)

Feature	Importance
Shannon Entropy	High
Min-Entropy	High
Variance	Medium
Runs	Medium
Autocorrelation	Medium
Mean	Low

Figure 5 shows Feature Importance.

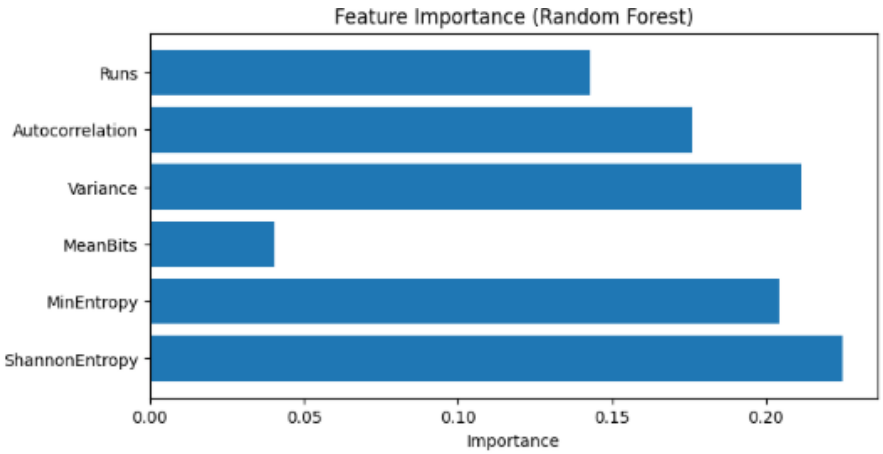

FIGURE 5.- Feature Importance

8. OVERALL DISCUSSION

The extensive experiments demonstrate that:

A. Entropy combined with the statistical properties offers a strong and very discriminative representation of cryptography keys.
B. The best classifier is random forest which also shows high accuracy and low variance on all datasets and key lengths and even noisy inputs.
C. KNN also is also very effective but tends to over-fitting synthetic data and not scalable to larger and true applications.
D. The SVM and Logistic Regression offer medium results and suit as comparing baselines.
E. The structure has a generalization over important lengths of keys (128-bit and 256-bit) and is robust to small deviations in the key bits.

From a cryptographic perspective, the results indicate that weak keys may still exhibit detectable structural regularities even when entropy values appear relatively high. This finding highlights an important limitation of relying solely on entropy-based evaluation methods. The proposed machine learning framework successfully captures hidden statistical dependencies and local patterns that traditional randomness measures may overlook,

improving the reliability of cryptographic key strength assessment. The experimental findings demonstrate that entropy measures alone are insufficient for capturing hidden statistical dependencies, repetitive patterns, or localized correlations within binary key sequences. In several cases, weak keys generated using biased distributions or deterministic structures produced entropy values that partially overlapped with those of strong keys. However, the machine learning models were still capable of accurately distinguishing these weak patterns through the integration of additional statistical features such as autocorrelation, runs, and variance. Furthermore, the superior performance of ensemble-based models, particularly Random Forest, suggests that cryptographic key classification involves complex non-linear relationships between entropy and statistical characteristics. This indicates that modern machine learning approaches can identify subtle structural weaknesses that traditional randomness tests may overlook. From a practical security perspective, these findings are significant because weak cryptographic keys can emerge in real-world systems due to insufficient entropy sources, flawed pseudo-random number generators, hardware imperfections, or implementation constraints in embedded and IoT devices. Consequently, the proposed framework provides a more comprehensive and reliable mechanism for cryptographic key strength evaluation compared to conventional entropy-only assessment techniques.

9. CONCLUSION

In this paper, an in-depth machine learning-grounded machine learning framework was presented that was used to analyze and classify cryptographic keys according to the entropy and statistics. This work was pertinent in understanding the fact that the quality of keys to cryptographic systems is a fundamental determinant of the security of cryptographic systems, and thus the limitations of the conventional entropy-only ways of system evaluation have been overcome by incorporating machine learning techniques in their design, which can identify desired and non-linear patterns.

The suggested framework integrates entropy statistics, such as Shannon entropy and min-entropy, with other statistical characteristics, namely, mean, variance, run and autocorrelation. A very large synthetically generated dataset of the strong and weak cryptographic keys was used in immense experiments. Several supervised learning algorithms were tested; they were Logistic Regression, K-Nearest Neighbors, Support Vector Machines, and the Random Forest classifier.

It was found in the experimental results that ensemble-based models have a high performance in accuracy, robustness, and generalization, and this is especially found in the random forest. The framework was observed to perform well with the full feature set with an overall accuracy of over 99 percent, which is much higher than that of entropy-only methods. The results of cross-validation established the stability of the models and did not indicate any form of overfitting. Additionally, the importance of statistical properties in supplementing the entropy measures as an efficient key classifier was also noted in the ablation study.

The usefulness of the proposed approach was also further confirmed through experiments. Analysis of key length revealed that the performance of the key length (both 128-bit key length and 256-bit key length) was uniform and exhibited scalability and generalization. The experiment with noisy keys experiment revealed the soundness of the model in the more realistic setting, where the essential perturbations or the errors at the bit-level level could be detected. The feature importance analysis also gave a good understanding of the relative importance of each feature which corroborated the usefulness of the hybrid feature representation.

Altogether, the findings verify that machine learning potentially with carefully designed entropy and statistical characteristics is a promising and consistent solution to the cryptographic key strength measuring. The suggested framework is an automated, precise, and scalable strategy, which can assist the security analysts and system designs to assess the key generation procedure and acquire weak cryptographic keys.

REFERENCES

[1] Mukherjee, Pratyusa, et al. "Best fit DNA-based cryptographic keys: the genetic algorithm approach." Sensors 22.19 (2022): 7332. DOI: 10.3390/s22197332

[2] Salami, Yashar, Vahid Khajevand, and Esmaeil Zeinali. "Cryptographic algorithms: a review of the literature, weaknesses and open challenges." J. Comput. Robot 16.2 (2023): 46-56. DOI: 10.22094/JCR.2023.1983496.1298

[3] Saini, Abhishek, and Ruchi Sehrawat. "Enhancing data security through machine learning-based key generation and encryption." Engineering, Technology & Applied Science Research 14.3 (2024): 14148-14154. DOI: 10.48084/etasr.7181

[4] Radanliev, Petar. "Cyber-attacks on Public Key Cryptography." (2023). DOI: 10.20944/preprints202309.1769.v1

[5] Zolfaghari, Behrouz, Khodakhast Bibak, and Takeshi Koshiba. "The odyssey of entropy: Cryptography." Entropy 24.2 (2022): 266. DOI: 10.3390/e24020266

[6] Gao, Dongjie. "Gene sequence analysis model construction based on k-mer statistics." PloS one 19.9 (2024): e0306480. DOI: 10.1371/journal.pone.0306480

[7] Bikos, Anastasios, et al. "Random number generators: Principles and applications." Cryptography 7.4 (2023): 54. DOI: 10.3390/cryptography7040054

[8] Anees, Amir, et al. "Machine learning and applied cryptography." Security & Communication Networks (2022). DOI: 10.1155/2022/7009002

[9] Roy, Pritam. "Enhancing Real-World Robustness in AI: Challenges and Solutions." J. Recent Trends Comput. Sci. Eng 12.1 (2024): 34-49. DOI: 10.70589/JRTCSE.2024.1.6

[10] Blanco-Romero J, Lorenzo V, Almenares Mendoza F, Díaz-Sánchez D. Machine Learning Predictors for Min-Entropy Estimation. Entropy (Basel). 2025 Feb 2;27(2):156. doi: 10.3390/e27020156. PMID: 40003153; PMCID: PMC11854237. DOI: 10.3390/e27020156

[11] Guo, X.; Zhou, W.; Luo, Y.; Meng, X.; Li, J.; Bian, Y.; Guo, Y.; Xiao, L. Deep Learning-Based Min-Entropy-Accelerated Evaluation for High-Speed Quantum Random Number Generation. Entropy 2025, 27, 786. https://doi.org/10.3390/e27080786.DOI: 10.3390/e27080786

[12] Kim, H.; Lim, S.; Kang, Y.; Kim, W.; Kim, D.; Yoon, S.; Seo, H. Deep-Learning-Based Cryptanalysis of Lightweight Block Ciphers Revisited. Entropy 2023, 25, 986. https://doi.org/10.3390/e25070986. DOI: 10.3390/e25070986

[13] Monteiro V (2024) The Importance of Entropy Sources in Cryptography Randomness to Secure Communications. J Inform Tech Softw Eng. 14:393. DOI: 10.35248/2165-7866.24.14.393

[14] Ali, Aqib, Sania Anam, and Muhammad Munawar Ahmed. "Shannon entropy in artificial intelligence and its applications based on information theory." J. Appl. Emerg. Sci 13.1 (2023): 9-17. DOI: 10.36785/JAES.131549

[15] Blanco-Romero, Javier, et al. "Machine learning predictors for min-entropy estimation." Entropy 27.2 (2025): 156. DOI: 10.3390/e27020156

[16] Obi, Jude Chukwura. "A comparative study of several classification metrics and their performances on data." World Journal of Advanced Engineering Technology and Sciences 8.1 (2023): 308-314. DOI: 10.30574/wjaets.2023.8.1.0046

[17] St-Aubin, Philippe, and Bruno Agard. "Precision and reliability of forecasts performance metrics." Forecasting 4.4 (2022): 882-903. DOI: 10.3390/forecast4040048