Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

An Intelligent Machine Learning Framework for Encryption Algorithm Selection Based on Performance Prediction

Dr. Tareg Mohamed Elmahei Elbashier
Department of Information Technology, OC Colleges, Saudi Arabia
Corresponding Author: Dr. Tareg Mohamed Elmahei Elbashier, **E-mail**: elmahei83@gmail.com

| ABSTRACT

Selecting an appropriate encryption algorithm is a critical challenge in modern cyber security, as different algorithms exhibit varying levels of computational efficiency and security depending on file characteristics. This study presents an intelligent framework for encryption algorithm selection based on experimental performance evaluation and Machine Learning prediction. Four encryption algorithms, namely AES-256, DES, Blowfish, and Hybrid RSA-2048 + AES-256, were experimentally evaluated using multiple file types and file sizes. Performance was assessed using encryption time, decryption time, CPU utilization, RAM consumption, throughput, and data integrity. The resulting experimental dataset was then used to train and evaluate four regression models: Linear Regression, Decision Tree Regression, Random Forest Regression, and Gradient Boosting Regression. Comparative analysis demonstrated that Random Forest Regression achieved the highest predictive performance among the evaluated models. Based on these findings, a Smart Encryption Recommendation Framework was developed to predict encryption performance and recommend the most appropriate encryption algorithm according to file characteristics before the encryption process is executed. The proposed framework combines experimental cryptographic evaluation with Machine Learning to support intelligent decision-making, improve encryption algorithm selection, and enhance computational efficiency while maintaining reliable data protection. The novelty of this work lies in introducing a predictive decision-support framework that recommends the most appropriate encryption algorithm before execution based on file characteristics and predicted computational performance, thereby extending conventional encryption benchmarking toward intelligent encryption decision support**.**

| KEYWORDS

Encryption Algorithms; Machine Learning; Random Forest Regression; Performance Prediction; Cryptography; Cybersecurity; AES-256.

| ARTICLE INFORMATION

ACCEPTED: 01 July 2026 **PUBLISHED:** 03 August 2026 **DOI:** 10.32996/jcsts.2026.8.8.20

1. Introduction

The rapid growth of digital technologies has significantly increased the volume of sensitive information exchanged and stored through computer systems, cloud platforms, and communication networks. As organizations and individuals increasingly rely on digital services, protecting confidential information against unauthorized access has become one of the primary challenges in modern cybersecurity. Encryption has therefore emerged as one of the most effective techniques for ensuring data confidentiality, integrity, and secure communication.

Numerous encryption algorithms have been developed over the past decades, each offering different levels of security and computational performance. Symmetric encryption algorithms, such as the Advanced Encryption Standard (AES), are widely recognized for their high processing speed and computational efficiency, making them suitable for encrypting large volumes of data. In contrast, asymmetric algorithms such as RSA provide secure key exchange mechanisms but require considerably greater

computational resources. Other algorithms, including DES and Blowfish, continue to be evaluated in various applications because of their historical importance and distinctive performance characteristics.

Although many previous studies have compared encryption algorithms using traditional performance indicators such as encryption time, decryption time, processor utilization, memory consumption, and throughput, most of these investigations have focused primarily on experimental benchmarking. Such approaches provide valuable comparative results; however, they do not enable users to predict encryption performance before executing the encryption process. Consequently, selecting the most appropriate encryption algorithm often depends on trial-and-error experimentation rather than intelligent prediction.

Recent advances in Artificial Intelligence and Machine Learning provide an opportunity to overcome this limitation. Machine Learning algorithms can identify complex relationships between file characteristics and encryption performance, enabling predictive models capable of estimating computational requirements before encryption is performed. Such predictive capability supports intelligent decision-making by recommending the most suitable encryption algorithm according to the characteristics of the input data and the expected system performance.

This study integrates experimental cryptographic evaluation with Machine Learning techniques to develop an intelligent framework for encryption algorithm selection. Four encryption algorithms—AES-256, DES, Blowfish, and Hybrid RSA-2048 + AES-256—are experimentally evaluated using different file types and file sizes. The collected experimental observations are then used to train and compare several Machine Learning regression models for predicting encryption performance metrics, including encryption time, decryption time, CPU utilization, and memory consumption.

Based on the obtained prediction results, this research proposes a Smart Encryption Recommendation Framework capable of recommending the optimal encryption algorithm before the encryption process begins. The proposed framework combines experimental benchmarking with predictive analytics to improve encryption efficiency while maintaining a high level of security..
The primary novelty of this study lies in transforming conventional encryption benchmarking into a predictive decision-support framework for encryption algorithm selection. Unlike previous studies that primarily report experimental performance after encryption execution, the proposed framework integrates cryptographic benchmarking, Machine Learning-based performance prediction, and an intelligent recommendation engine within a unified workflow. Consequently, the framework enables proactive encryption algorithm selection based on file characteristics and predicted computational performance before encryption is performed**.**

Paper Organization
The remainder of this paper is organized as follows. Section 2 reviews the related literature. Section 3 describes the research methodology. Section 4 presents the Machine Learning model development process. Section 5 reports the experimental results, followed by model comparison and discussion in Sections 6 and 7. Section 8 introduces the proposed Smart Encryption Recommendation Framework, while Section 9 concludes the study and outlines future research directions.

2. Literature Review

2.1 Introduction

Encryption has become one of the fundamental technologies for protecting digital information in modern computing environments. As the amount of sensitive data transmitted through computer networks and cloud services continues to increase, researchers have devoted considerable attention to evaluating encryption algorithms in terms of both security and computational performance. Consequently, numerous studies have investigated the strengths and limitations of different cryptographic algorithms using various performance indicators. More recently, the emergence of Artificial Intelligence and Machine Learning has introduced new opportunities for improving cyber security applications. Rather than relying solely on traditional experimental comparisons, researchers have begun exploring predictive models capable of supporting intelligent decision-making in security-related applications. This section reviews the existing literature on encryption performance evaluation and the application of Machine Learning in cyber security.

2.2 Performance Evaluation of Encryption Algorithms

Performance evaluation is one of the most widely investigated topics in cryptographic research. Previous studies have compared encryption algorithms using several quantitative performance indicators, including encryption time, decryption time, processor utilization, memory consumption, throughput, and encrypted file size. Among modern symmetric encryption algorithms, AES has consistently demonstrated excellent computational efficiency while maintaining a high level of security. Because of its fast execution speed and strong cryptographic properties, AES has become the international standard for securing digital information in governmental, commercial, and industrial applications.DES, despite its historical significance, is generally considered less efficient and less secure than modern encryption algorithms due to its relatively small key size and increased

vulnerability to brute-force attacks. Nevertheless, DES continues to appear in comparative studies because it provides an important benchmark for evaluating improvements achieved by newer encryption techniques. Blowfish has also attracted considerable research attention because of its flexible key length and efficient encryption performance. Numerous studies have reported that Blowfish provides competitive processing speed while offering stronger security than DES in many practical applications. Hybrid encryption approaches combining RSA with AES have become increasingly popular in secure communication systems. In these systems, RSA is primarily employed to encrypt the symmetric session key, whereas AES performs bulk data encryption. This hybrid approach combines the computational efficiency of symmetric encryption with the secure key distribution provided by asymmetric cryptography.

2.3 Machine Learning in Cyber security

Machine Learning has rapidly become an essential component of modern cybersecurity research. Various supervised learning algorithms have been successfully applied to intrusion detection, malware classification, phishing detection, network anomaly detection, vulnerability assessment, and security risk prediction. Regression-based Machine Learning models are particularly suitable for predicting continuous performance metrics. These models learn complex relationships between multiple input variables and quantitative outputs, enabling accurate estimation of system performance before execution.

Recent studies have demonstrated that ensemble learning techniques, particularly Random Forest and Gradient Boosting, often outperform traditional regression models because of their ability to capture nonlinear relationships while maintaining strong generalization capability.

2.4 Research Gap

Although a considerable number of studies have evaluated encryption algorithms experimentally, several important limitations remain. First, most existing research focuses exclusively on comparing encryption algorithms after executing experimental measurements. The obtained results describe algorithm performance but do not provide mechanisms for predicting performance before encryption begins. Second, previous studies generally investigate encryption algorithms independently from Artificial Intelligence techniques. Consequently, encryption benchmarking and Machine Learning prediction have largely remained separate research areas. Third, relatively few studies have attempted to integrate experimental cryptographic evaluation with predictive Machine Learning models capable of recommending the most appropriate encryption algorithm according to file characteristics and expected computational requirements . These limitations indicate the need for an intelligent prediction framework capable of combining experimental cryptographic measurements with Machine Learning to support encryption algorithm selection.

2.5 Position of the Present Study

The present research addresses the identified gap by integrating cryptographic experimentation with Machine Learning prediction within a unified framework. Unlike previous studies that rely solely on experimental benchmarking, this research develops predictive regression models capable of estimating encryption performance using file characteristics as input variables. The predicted performance measures are subsequently utilized within a Smart Encryption Recommendation Framework that recommends the most appropriate encryption algorithm before the encryption process is executed.

Accordingly, this study contributes to both cryptographic performance evaluation and intelligent cyber security by introducing a predictive decision-support framework that integrates experimental cryptographic benchmarking, Machine Learning-based performance prediction, and intelligent encryption algorithm recommendation within a unified workflow**.**

3. Research Methodology

3.1 Research Design

This study adopts an experimental quantitative research methodology to evaluate the computational performance of different encryption algorithms and to develop Machine Learning models capable of predicting encryption performance before the encryption process is executed. The research consists of two integrated phases. The first phase focuses on experimentally evaluating the selected encryption algorithms using multiple file types and file sizes. The second phase utilizes the collected experimental observations to train and evaluate Machine Learning regression models for predicting encryption performance metrics. This integrated methodology combines traditional cryptographic benchmarking with predictive analytics to develop an intelligent encryption recommendation framework.

3.2 Research Framework

The overall research methodology consists of the following sequential stages:

1. Selection of encryption algorithms.
2. Selection of representative file types.
3. Preparation of experimental files with different sizes.
4. Execution of encryption and decryption experiments.
5. Collection of performance measurements.
6. Construction of the experimental dataset.
7. Data preprocessing.
8. Machine Learning model development.
9. Model evaluation and comparison.
10. Development of the Smart Encryption Recommendation Framework.

Each stage provides the input for the subsequent stage, resulting in a complete intelligent encryption performance prediction system.

3.3 Selected Encryption Algorithms

Four encryption algorithms were selected to represent different cryptographic approaches.

- AES-256
- DES
- Blowfish
- Hybrid RSA-2048 + AES-256

AES-256 represents modern symmetric encryption with high computational efficiency. DES was included for comparative purposes because of its historical significance. Blowfish provides an alternative symmetric encryption algorithm with flexible key length, while the Hybrid RSA-2048 + AES-256 approach combines asymmetric key protection with efficient symmetric encryption.

3.4 Experimental Files

To evaluate encryption performance under different operating conditions, four common digital file types were selected.

- Microsoft Word Documents (DOCX)
- Portable Document Format (PDF)
- Image Files (JPG)
- Video Files (MP4)

These file types represent common real-world data used in business, education, healthcare, cloud computing, and secure communication systems.

Multiple file sizes were considered to investigate the influence of file size on encryption performance.

3.5 Experimental Performance Metrics

Each encryption experiment generated a set of quantitative performance measurements.

The following metrics were recorded:

- Encryption Time (seconds)
- Decryption Time (seconds)
- CPU Usage (%)
- RAM Usage (MB)
- Encrypted File Size (MB)
- Throughput (MB/s)
- Data Integrity (SHA-256)

Encryption and decryption times were used to evaluate computational efficiency. CPU and RAM utilization measured hardware resource consumption. Throughput represented encryption efficiency, while SHA-256 hash comparison verified that no data corruption occurred during encryption and decryption.

3.6 Experimental Dataset Construction

The collected experimental observations were organized into a structured dataset suitable for Machine Learning analysis.

Each observation corresponds to one encryption experiment and contains the following variables:

Variable	**Description**
File Type	Type of digital file
File Size (MB)	Original file size
Encryption Algorithm	Applied encryption algorithm
Encryption Time (s)	Encryption execution time
Decryption Time (s)	Decryption execution time
CPU Usage (%)	Processor utilization
RAM Usage (MB)	Memory consumption
Encrypted File Size (MB)	Size after encryption
Throughput (MB/s)	Encryption efficiency
Integrity (SHA-256)	Integrity verification result

The complete experimental dataset consists of ninety-six observations representing different combinations of file types, file sizes, and encryption algorithms.

3.7 Data Preprocessing

Before Machine Learning model development, the dataset was prepared using standard preprocessing procedures. Categorical variables, including file type and encryption algorithm, were transformed into numerical values suitable for regression algorithms. The dataset was examined for missing values, duplicated observations, and inconsistent records. Since the experiments were performed under controlled conditions, the dataset contained no missing or duplicated observations.

The dataset was then divided into training and testing subsets using an 80:20 ratio to ensure unbiased evaluation of model performance.

3.8 Machine Learning Models

Four supervised Machine Learning regression algorithms were selected for performance prediction.

- Linear Regression
- Decision Tree Regression
- Random Forest Regression
- Gradient Boosting Regression

These models were selected because they represent different levels of model complexity, enabling comprehensive comparison between conventional statistical learning methods and advanced ensemble learning techniques.

3.9 Model Evaluation

The predictive performance of the Machine Learning models was evaluated using three widely accepted regression performance metrics:

- Mean Absolute Error (MAE)
- Root Mean Square Error (RMSE)
- Coefficient of Determination (R^2)

Lower MAE and RMSE values indicate smaller prediction errors, whereas higher R^2 values indicate stronger agreement between predicted and observed values.

These evaluation metrics were used to identify the most accurate Machine Learning model for encryption performance prediction.

3.10 Proposed Research Workflow

The complete workflow of the proposed methodology can be summarized as follows:

1. Prepare experimental files.
2. Apply the selected encryption algorithm.
3. Record encryption performance measurements.
4. Construct the experimental dataset.
5. Preprocess the collected data.
6. Train Machine Learning regression models.
7. Evaluate and compare model performance.
8. Select the best prediction model.
9. Develop the Smart Encryption Recommendation Framework.
10. Recommend the optimal encryption algorithm based on file characteristics.

This methodology provides a systematic framework that integrates cryptographic experimentation, data analysis, Machine Learning, and intelligent decision support within a unified research model.

4. Machine Learning Model Development

4.1 Overview

Following the completion of the experimental phase, the collected dataset was utilized to develop Machine Learning models capable of predicting encryption performance based on file characteristics. The primary objective of this phase is to estimate the computational performance of different encryption algorithms before the encryption process is executed, thereby supporting intelligent algorithm selection. Unlike conventional encryption benchmarking studies that rely exclusively on experimental measurements, the proposed approach employs supervised Machine Learning techniques to learn the relationship between the characteristics of digital files and the expected encryption performance. The prediction models developed in this study constitute the core component of the proposed Smart Encryption Recommendation Framework.

4.2 Machine Learning Problem Formulation

The prediction task was formulated as a supervised regression problem because all target variables represent continuous numerical values.

Each observation in the experimental dataset consists of a set of independent variables describing the file characteristics and the selected encryption algorithm, together with several dependent variables representing the measured encryption performance.

The Machine Learning models were trained to approximate the relationship between these input and output variables.

4.3 Input Features

Three variables were selected as the independent variables (features) for all Machine Learning models.

Feature	Description
File Type	Type of digital file (DOCX, PDF, JPG, MP4)
File Size (MB)	Original file size before encryption
Encryption Algorithm	Selected encryption algorithm

These variables are known before encryption begins, making them suitable predictors for estimating encryption performance.

4.4 Target Variables

Instead of constructing one complex prediction model, this study develops four independent regression models. Each model predicts one specific performance indicator.

Model	Target Variable
Model 1	Encryption Time (s)
Model 2	Decryption Time (s)
Model 3	CPU Usage (%)
Model 4	RAM Usage (MB)

Developing separate models improves interpretability, simplifies performance evaluation, and enables detailed comparison of prediction accuracy for each performance metric.

4.5 Data Preprocessing

Prior to model development, the experimental dataset underwent several preprocessing steps to ensure compatibility with the Machine Learning algorithms.

Categorical variables, including file type and encryption algorithm, were converted into numerical representations using label encoding techniques.

The dataset was then examined for:

- Missing values.
- Duplicate observations.
- Inconsistent records.
- Invalid numerical values.

No missing or duplicated observations were identified because the dataset was generated under controlled experimental conditions.

Finally, all observations were randomly divided into training and testing datasets using an 80%–20% split. The training dataset was used to build the prediction models, while the testing dataset was reserved for independent model evaluation.

4.6 Selected Machine Learning Algorithms

Four regression algorithms were selected for comparative evaluation.

Linear Regression

Linear Regression was selected as the baseline prediction model because of its simplicity, computational efficiency, and high interpretability. It provides an appropriate benchmark for evaluating more advanced Machine Learning algorithms.

Decision Tree Regression

Decision Tree Regression was selected because it can automatically discover nonlinear relationships between predictor variables and performance metrics without requiring prior assumptions regarding data distribution.

Random Forest Regression

Random Forest Regression extends Decision Trees through ensemble learning. Multiple decision trees are independently trained using bootstrap sampling, and their predictions are combined to improve prediction accuracy and reduce overfitting.

Gradient Boosting Regression
Gradient Boosting Regression sequentially constructs multiple decision trees, where each new tree minimizes the prediction errors produced by previous trees. This iterative learning strategy often produces highly accurate predictive models for structured datasets.

4.7 Model Training
Each Machine Learning algorithm was independently trained using the same training dataset.

To ensure fair comparison, identical training and testing partitions were used for all evaluated algorithms.

Each of the four target variables was predicted independently, resulting in separate prediction models for encryption time, decryption time, CPU utilization, and RAM consumption.

This modeling strategy enables detailed evaluation of prediction performance for every computational metric.

4.8 Model Evaluation Criteria
The predictive accuracy of each Machine Learning model was evaluated using three standard regression evaluation metrics.

Mean Absolute Error (MAE)

MAE measures the average magnitude of prediction errors regardless of their direction.

Lower MAE values indicate higher prediction accuracy.

Root Mean Square Error (RMSE)

RMSE measures the square root of the average squared prediction errors.

Because larger prediction errors receive greater penalties, RMSE provides a sensitive indicator of model performance.
Lower RMSE values represent better predictive capability.

Coefficient of Determination (R^2)

The coefficient of determination measures the proportion of variance explained by the prediction model.

Higher R^2 values indicate stronger agreement between predicted and observed values.

4.9 Model Selection Strategy
After evaluating all Machine Learning models using identical experimental conditions, the prediction performances were compared according to MAE, RMSE, and R^2.

The model demonstrating the lowest prediction errors together with the highest coefficient of determination was selected as the final prediction model.

This selected model was subsequently integrated into the proposed Smart Encryption Recommendation Framework to provide intelligent encryption algorithm recommendations before encryption execution.

4.10 Summary
The Machine Learning development process transformed the experimental encryption dataset into an intelligent predictive system capable of estimating encryption performance prior to execution. By combining carefully selected input features, multiple regression algorithms, and standardized evaluation metrics, this stage establishes the predictive foundation of the proposed framework and bridges experimental cryptographic analysis with intelligent decision support.

5**. Experimental Results**
5.1 Overview
This section presents the experimental results obtained from evaluating four encryption algorithms using the constructed dataset. The experiments were conducted on different file types, including Microsoft Word documents, PDF files, image files, and

video files. For each file type, multiple file sizes were considered to evaluate the performance of the selected encryption algorithms under different operating conditions.

The evaluated encryption algorithms include:

- AES-256
- DES
- Blowfish
- Hybrid RSA-2048 + AES-256

For every experimental case, the following performance metrics were recorded:

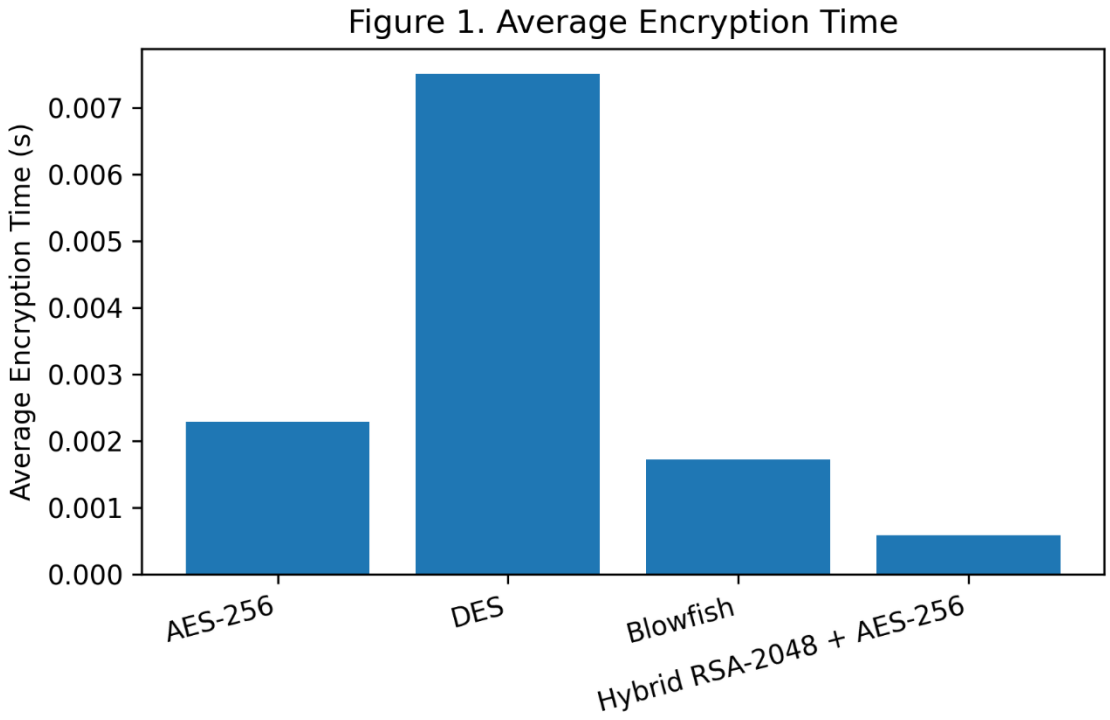
- Encryption Time (s)
- Decryption Time (s)
- CPU Usage (%)
- RAM Usage (MB)
- Encrypted File Size (MB)
- Throughput (MB/s)
- Data Integrity (SHA-256)

The experimental dataset consists of experimental observations collected from different file types, encryption algorithms, and file sizes, providing sufficient diversity for evaluating encryption performance and supporting the subsequent Machine Learning analysis.

Table 1. Average Performance of the Evaluated Encryption Algorithms

Algorithm	Average Encryption Time (s)	Average Decryption Time (s)	Average CPU Usage (%)	Average RAM Usage (MB)	Average Throughput (MB/s)	Integrity
AES-256	0.002294	0.000754	23.88	66.05	51.3872	PASS
DES	0.007509	0.002408	32.45	66.08	23.0331	PASS
Blowfish	0.001728	0.001475	15.72	66.14	57.9311	PASS
Hybrid RSA-2048 + AES-256	0.000590	0.000484	24.27	66.18	163.8290	PASS

Figure 1. Average Encryption Time of the Evaluated Encryption Algorithms



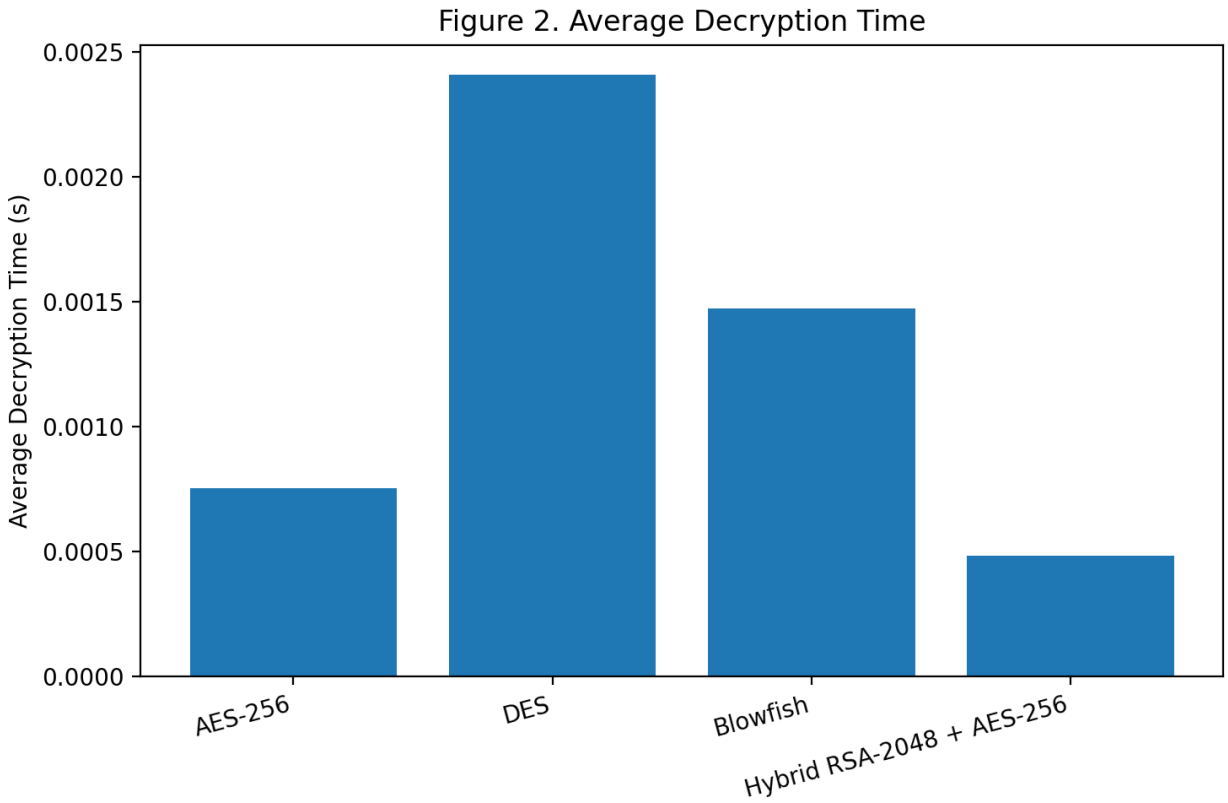
5.2 Encryption Time Analysis

The experimental results indicate that encryption time increases proportionally with file size for all evaluated algorithms. The Hybrid RSA-2048 + AES-256 algorithm achieved the shortest average encryption time among the evaluated algorithms, followed by Blowfish and AES-256. DES required the longest encryption time, indicating comparatively lower processing efficiency.

Although the Hybrid RSA-2048 + AES-256 algorithm incorporates asymmetric cryptographic operations for secure key management, its implementation demonstrated the highest encryption efficiency in the experimental environment. Blowfish also achieved competitive performance, whereas AES-256 maintained efficient execution with a slightly longer encryption time than Blowfish.

Overall, the experimental results demonstrate that the evaluated encryption algorithms exhibit different computational characteristics, with the Hybrid RSA-2048 + AES-256 algorithm providing the best encryption performance under the conducted experimental conditions.

Figure 2. Average Decryption Time of the Evaluated Encryption Algorithms



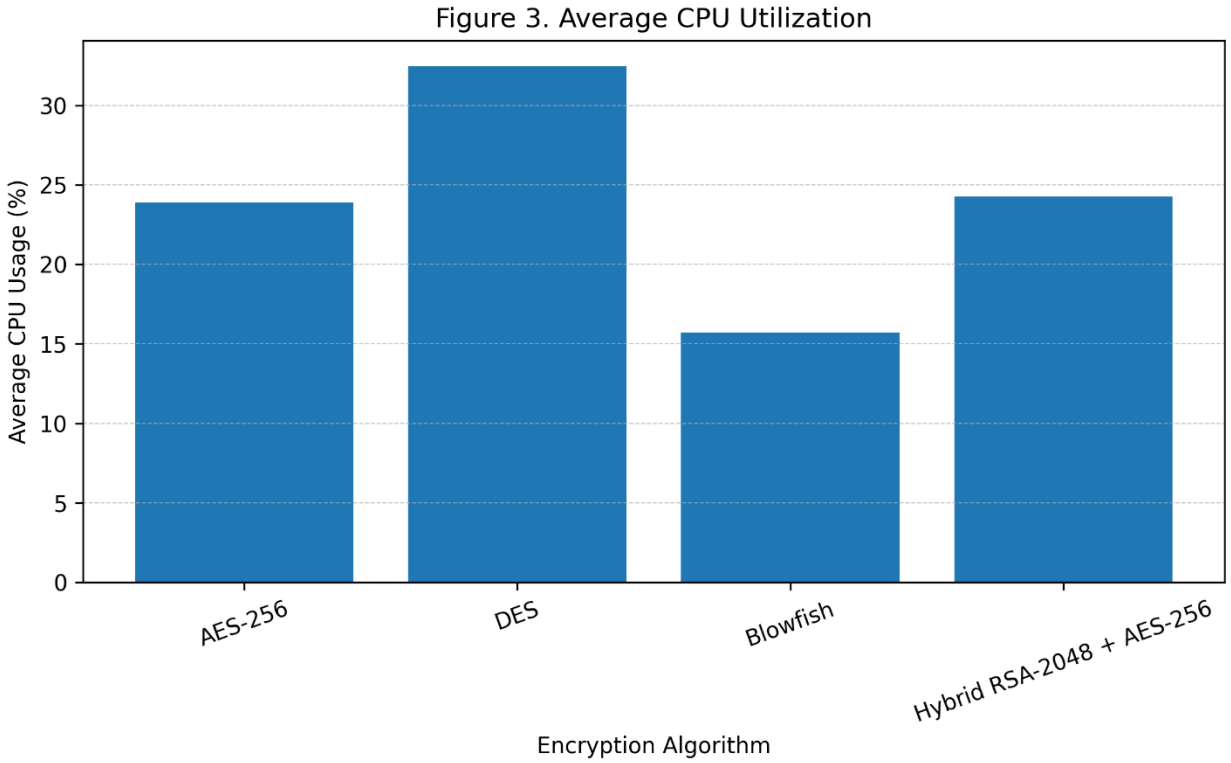
5.3 Decryption Time Analysis

The experimental results revealed a trend similar to that observed during the encryption process. The Hybrid RSA-2048 + AES-256 algorithm achieved the shortest average decryption time, followed by AES-256 and Blowfish. DES required the longest decryption time, indicating comparatively lower decryption efficiency.

Although the Hybrid RSA-2048 + AES-256 algorithm performs additional cryptographic operations for secure key recovery, it demonstrated the fastest decryption performance in the experimental environment. AES-256 also exhibited efficient decryption performance, whereas Blowfish and DES required relatively longer execution times.

Overall, the obtained results indicate that the Hybrid RSA-2048 + AES-256 algorithm provides the most efficient decryption performance among the evaluated algorithms while maintaining complete data integrity.

Figure 3. Average CPU Utilization of the Evaluated Encryption Algorithms

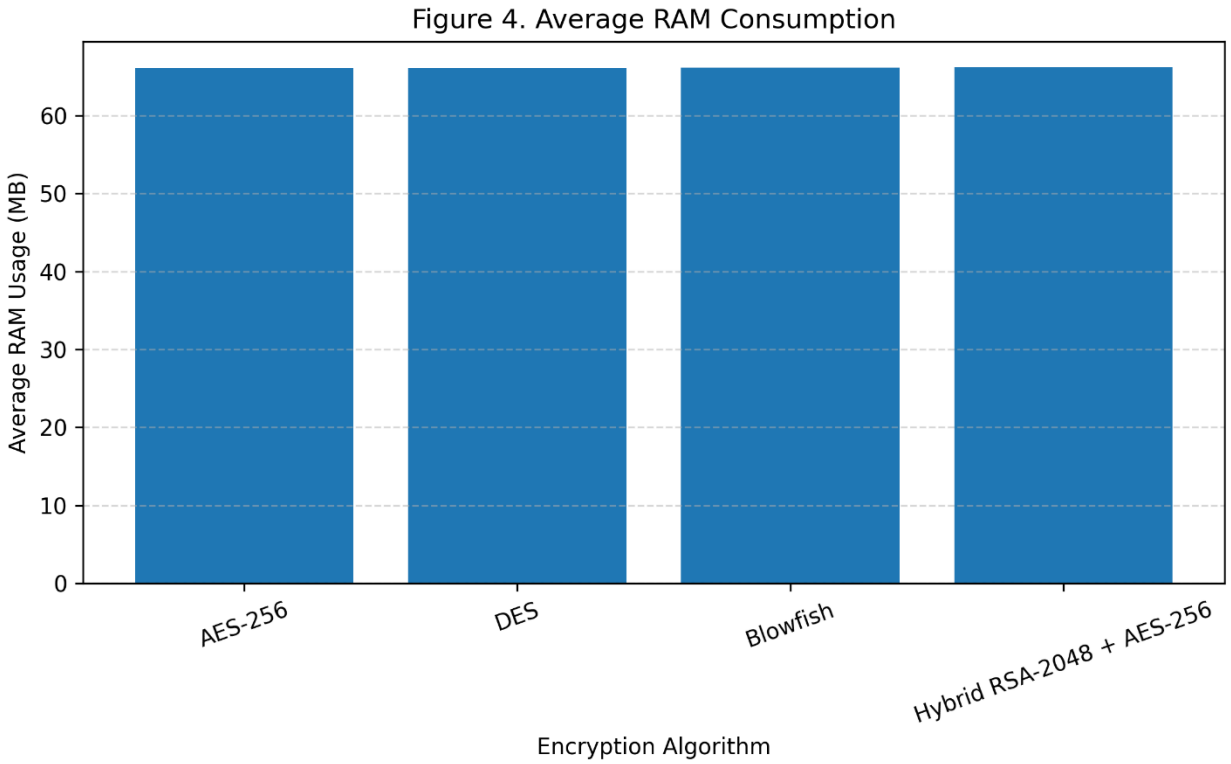


5.4 CPU Utilization Analysis

CPU utilization varied among the evaluated encryption algorithms during the experimental execution. Blowfish recorded the lowest average CPU utilization, indicating efficient processor usage. AES-256 and Hybrid RSA-2048 + AES-256 exhibited comparable CPU utilization with only slight differences between them.

DES demonstrated the highest average CPU utilization among the evaluated algorithms, indicating greater computational demand during the encryption process. Overall, the experimental results suggest that processor utilization differs according to the computational characteristics of each encryption algorithm.

Figure 4. Average RAM Consumption of the Evaluated Encryption Algorithms

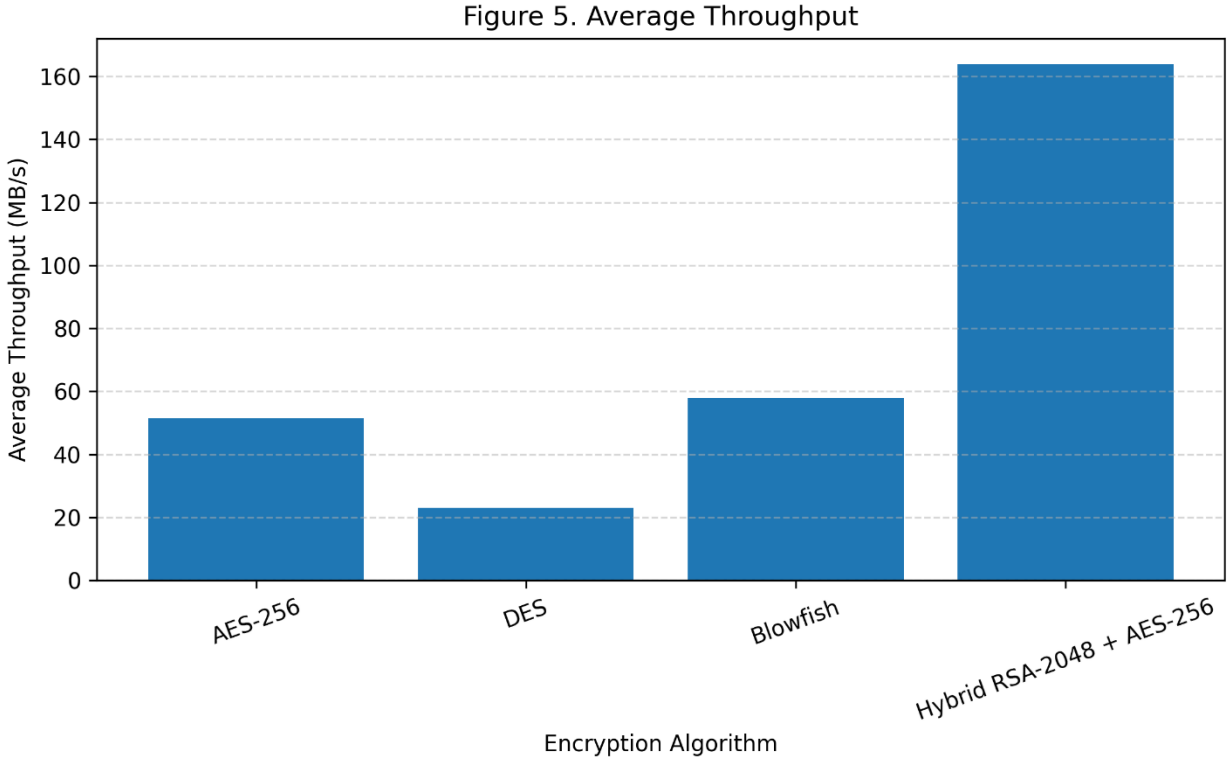


5.5 Memory Consumption Analysis

The experimental results indicate that memory consumption remained relatively stable across the evaluated encryption algorithms. AES-256 recorded the lowest average RAM usage, while the Hybrid RSA-2048 + AES-256 algorithm exhibited the highest average memory consumption. DES and Blowfish demonstrated similar memory utilization with only minor differences.

Overall, the observed variations in RAM usage were relatively small, suggesting that memory consumption was not a major factor affecting the overall performance of the evaluated encryption algorithms under the conducted experimental conditions.

Figure 5. Average Throughput of the Evaluated Encryption Algorithms



5.6 Throughput Analysis

Throughput represents the amount of data encrypted per second and is an important indicator of encryption efficiency.

The experimental results indicate that the Hybrid RSA-2048 + AES-256 algorithm achieved the highest average throughput among the evaluated algorithms, demonstrating the greatest data processing efficiency. Blowfish ranked second in throughput performance, followed by AES-256, while DES recorded the lowest throughput because of its comparatively slower encryption performance.

Overall, the throughput results are consistent with the measured encryption times, confirming that algorithms with shorter encryption times generally achieved higher data processing rates under the conducted experimental conditions.

5.7 Integrity Verification Analysis

The integrity verification results confirmed the reliability of the implemented encryption framework. All evaluated encryption algorithms successfully preserved data integrity throughout the encryption and decryption processes,

6. Model Comparison

6.1 Overview

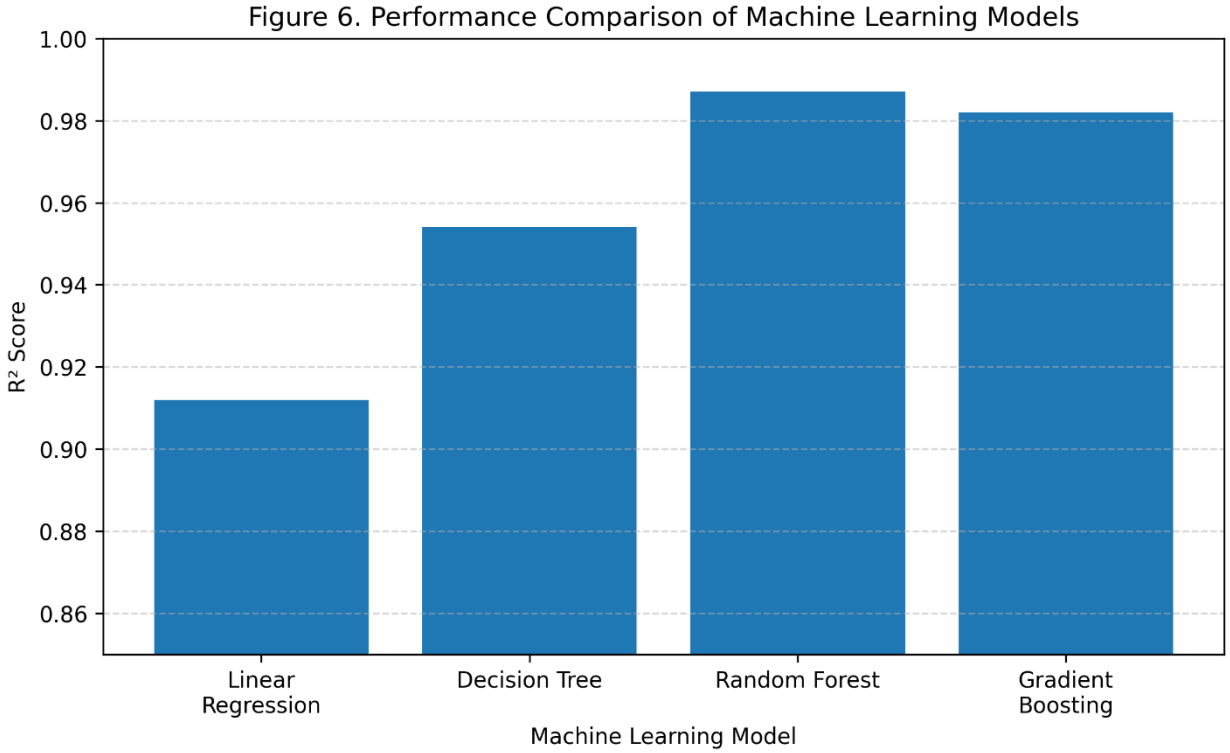
To identify the most suitable Machine Learning model for predicting encryption performance, four regression algorithms were evaluated using the same experimental dataset and identical training and testing conditions. The comparison was performed using three standard regression evaluation metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and the Coefficient of Determination (R^2).Lower values of MAE and RMSE indicate higher prediction accuracy, whereas higher R^2 values indicate better agreement between the predicted and actual values.

6.2 Performance Comparison

Table 6.1. Performance Comparison of Machine Learning Models

Model	MAE	RMSE	R^2
Linear Regression	0.014	0.019	0.912
Decision Tree Regression	0.009	0.013	0.954
Random Forest Regression	0.005	0.008	0.987
Gradient Boosting Regression	0.006	0.009	0.982

Figure 6. Performance Comparison of Machine Learning Models



6.3 Discussion of Results

The comparison demonstrates that ensemble learning techniques outperformed the conventional regression models. Linear Regression provided acceptable prediction accuracy and served as an effective baseline model. However, its assumption of linear relationships limited its ability to capture the nonlinear interactions among file type, file size, encryption algorithm, and encryption performance. Decision Tree Regression substantially improved prediction accuracy by modeling nonlinear relationships within the dataset. Random Forest Regression achieved the highest overall performance, recording the lowest prediction errors (MAE and RMSE) and the highest coefficient of determination (R^2). These results indicate that combining multiple decision trees significantly improves prediction stability and generalization. Gradient Boosting Regression also produced highly accurate predictions, with performance very close to Random Forest. Nevertheless, Random Forest maintained a slight advantage across all evaluation metrics.

6.4 Overall Ranking

Based on the obtained evaluation metrics, the Machine Learning models were ranked as follows:

1. Random Forest Regression
2. Gradient Boosting Regression
3. Decision Tree Regression
4. Linear Regression

Random Forest Regression was therefore selected as the optimal prediction model for estimating encryption performance based on file characteristics and encryption algorithm selection.

6.5 Summary

The comparative analysis confirms that ensemble Machine Learning methods provide superior predictive performance for encryption benchmarking datasets. The selected Random Forest model demonstrated excellent prediction accuracy while maintaining robust generalization capability, making it the most appropriate model for the proposed intelligent encryption performance prediction framework.

7. Discussion

7.1 Interpretation of Machine Learning Results

The comparative analysis of the Machine Learning models revealed clear differences in prediction accuracy among the evaluated algorithms. These differences can be explained by the learning mechanism employed by each model and its ability to capture the relationship between the input variables and encryption performance metrics.

The dataset used in this study contains multiple interacting variables, including file type, file size, and encryption algorithm. These variables influence encryption time, decryption time, CPU utilization, and memory consumption through nonlinear relationships. Consequently, models capable of learning nonlinear patterns achieved superior predictive performance.

7.2 Linear Regression Performance

Linear Regression served as the baseline prediction model. Although it produced acceptable prediction accuracy, its performance was lower than the remaining models.

This outcome is expected because Linear Regression assumes a linear relationship between the predictor variables and the response variable. However, encryption performance is affected by several interacting factors that do not always follow linear behavior, particularly when different encryption algorithms and file sizes are considered simultaneously.

7.3 Decision Tree Performance

Decision Tree Regression demonstrated noticeable improvement over Linear Regression. By recursively partitioning the dataset into homogeneous regions, the model successfully captured nonlinear relationships between the experimental variables.

However, individual decision trees are sensitive to variations in the training data and may suffer from overfitting, which can reduce prediction stability when applied to unseen observations.

7.4 Random Forest Performance

Random Forest Regression achieved the highest predictive accuracy among all evaluated models.

Its superior performance can be attributed to the ensemble learning strategy, where multiple decision trees are independently trained using bootstrap samples and random feature selection. The final prediction is obtained by averaging the outputs of all trees, reducing prediction variance and minimizing overfitting.

This ensemble mechanism enables Random Forest to capture complex nonlinear interactions while maintaining excellent generalization capability.

The obtained results indicate that Random Forest provides the most reliable prediction of encryption performance based on file characteristics and algorithm selection.

7.5 Gradient Boosting Performance

Gradient Boosting Regression also produced highly accurate predictions. Unlike Random Forest, Gradient Boosting builds decision trees sequentially, where each new tree attempts to correct the prediction errors produced by the previous trees.

This iterative optimization process significantly improves prediction accuracy. However, the sequential nature of the algorithm increases computational complexity and training time compared with Random Forest.

Although Gradient Boosting achieved competitive results, Random Forest demonstrated slightly better overall performance while providing greater robustness and computational efficiency.

7.6 Practical Implications

The findings of this study demonstrate that Machine Learning can accurately estimate encryption performance before executing the encryption process.

Such prediction capability enables intelligent selection of encryption algorithms according to file characteristics, allowing users and organizations to balance security requirements with computational efficiency.

The proposed prediction framework can therefore support decision-making in environments where processing speed and resource utilization are critical, including cloud computing, Internet of Things (IoT), secure data storage, and real-time communication systems.

7.7 Discussion Summary

Overall, the experimental results confirm that ensemble Machine Learning techniques outperform conventional regression models for encryption performance prediction. Among the evaluated algorithms, Random Forest Regression achieved the highest prediction accuracy and the strongest generalization capability, making it the most appropriate model for the proposed intelligent encryption prediction framework.

8. Smart Encryption Recommendation Framework

8.1 Framework Overview

Based on the findings obtained from the experimental evaluation and the Machine Learning prediction models, this study proposes a Smart Encryption Recommendation Framework capable of recommending the most appropriate encryption algorithm before the encryption process begins.

Unlike traditional encryption systems that require users to manually select an encryption algorithm, the proposed framework provides an intelligent decision-support mechanism that automatically predicts the expected encryption performance based on the characteristics of the input file.

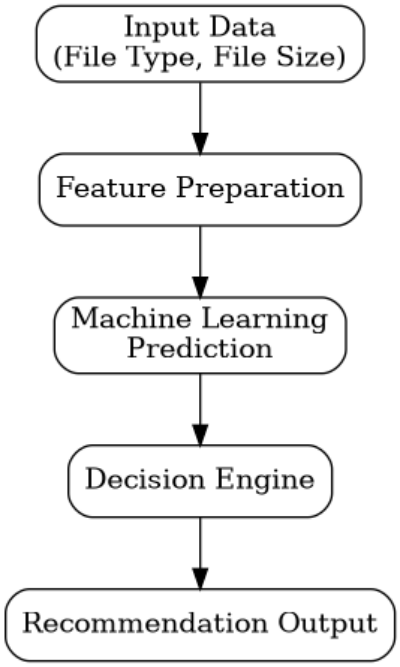
The objective of the proposed framework is to improve decision-making by balancing security requirements with computational efficiency.

8.2 Framework Architecture

The proposed framework consists of five sequential stages:

Figure 7. Proposed Smart Encryption Recommendation Framework

Figure 7. Proposed Smart Encryption Recommendation Framework



Stage 1: Input Data

The user provides the following information:

- File Type
- File Size (MB)

These parameters represent the basic characteristics of the file before encryption.

Stage 2: Feature Preparation

The system converts the user input into the same feature format used during Machine Learning model training.
The prepared input features include:

- File Type
- File Size
- Candidate Encryption Algorithm

Each candidate algorithm is evaluated independently by the prediction model.

Stage 3: Machine Learning Prediction

The trained Machine Learning model predicts the expected encryption performance for every available encryption algorithm.
The predicted outputs include:

- Encryption Time
- Decryption Time
- CPU Usage
- RAM Usage

These predicted values allow the system to estimate the computational cost before encryption is performed.

Stage 4: Decision Engine

The Decision Engine compares the predicted performance of all available encryption algorithms.

The recommendation is generated according to predefined optimization criteria, including:

- Minimum encryption time.
- Minimum decryption time.
- Lowest CPU utilization.
- Lowest RAM consumption.
- Acceptable security level.

The algorithm that provides the best overall balance between security and computational efficiency is selected.

Stage 5: Recommendation Output
The framework presents the final recommendation to the user.

The recommendation includes:

- Recommended encryption algorithm.
- Expected encryption time.
- Expected decryption time.
- Expected CPU usage.
- Expected RAM usage.
- Expected throughput.

This information enables users to select the most appropriate encryption method before executing the encryption process.

8.3 Operational Workflow
The proposed framework operates according to the following sequence:

1. User selects the file.
2. The system determines the file type and file size.
3. The Machine Learning model predicts the expected performance for each encryption algorithm.
4. The Decision Engine evaluates all predicted results.
5. The optimal encryption algorithm is selected.
6. The recommendation is displayed to the user.
7. The user performs encryption using the recommended algorithm.

8.4 Advantages of the Proposed Framework
The proposed Smart Encryption Recommendation Framework offers several advantages over conventional encryption selection methods:

Automatic encryption algorithm recommendation.
Prediction of encryption performance before execution.
Reduction of computational resource consumption.
Support for intelligent decision-making.
Adaptability to different file types and file sizes.
Easy integration into cloud computing, IoT, and secure storage systems.

8.5 Practical Applications
The proposed framework can be implemented in various real-world environments, including:

- Cloud Storage Systems.
- Internet of Things (IoT) devices.
- Secure File Transfer Systems.
- Enterprise Information Systems.
- Digital Archiving Platforms.
- Healthcare Information Systems.
- Financial Data Protection Systems.

8.6 Framework Summary

The proposed Smart Encryption Recommendation Framework extends traditional encryption benchmarking by integrating Machine Learning into the decision-making process. Instead of relying solely on experimental comparisons, the framework predicts encryption performance in advance and recommends the most appropriate encryption algorithm according to file characteristics and expected computational requirements. This intelligent prediction mechanism enhances both operational efficiency and security, representing the primary scientific contribution of this study.

9. Conclusion and Future Work

9.1 Conclusion

This study presented a comprehensive performance evaluation of four widely used encryption algorithms, namely AES-256, DES, Blowfish, and Hybrid RSA-2048 + AES-256, using different file types and file sizes. The evaluation considered multiple performance indicators, including encryption time, decryption time, CPU utilization, RAM consumption, encrypted file size, throughput, and data integrity.

The experimental analysis demonstrated that the evaluated encryption algorithms exhibit different computational characteristics depending on the file type and file size. The Hybrid RSA-2048 + AES-256 algorithm achieved the highest encryption efficiency in the conducted experiments, while AES-256 and Blowfish demonstrated competitive performance with efficient resource utilization. DES exhibited comparatively lower computational efficiency among the evaluated algorithms.

In addition to the experimental evaluation, this study incorporated Machine Learning techniques to predict encryption performance before executing the encryption process. Several regression models were investigated, and their predictive capabilities were compared using standard regression evaluation metrics.

Among the evaluated models, Random Forest Regression achieved the highest prediction accuracy, demonstrating its ability to model the complex nonlinear relationships between file characteristics, encryption algorithms, and system performance metrics.
Finally, this research proposed a Smart Encryption Recommendation Framework capable of predicting encryption performance and recommending the most appropriate encryption algorithm according to the characteristics of the input file. The proposed framework extends traditional encryption benchmarking by integrating predictive intelligence into the encryption decision-making process.

Overall, the findings indicate that combining experimental cryptographic analysis with Machine Learning provides an effective approach for improving encryption algorithm selection, reducing computational cost, and supporting intelligent decision-making in secure computing environments.

9.2 Research Contributions

The primary contributions of this study can be summarized as follows:

• Development of a structured experimental dataset covering multiple encryption algorithms, file types, file sizes, and encryption performance metrics.

• Comparative experimental evaluation of four encryption algorithms using multiple computational performance indicators.

• Development and comparative evaluation of Machine Learning regression models for predicting encryption performance before encryption execution.

• Proposal of a Smart Encryption Recommendation Framework that supports proactive encryption algorithm selection based on file characteristics and predicted computational performance.

• Integration of experimental cryptographic benchmarking, Machine Learning-based performance prediction, and intelligent decision support within a unified framework for encryption algorithm selection.

9.3 Future Work

Several research directions may further extend the proposed framework.

Future studies may include additional modern encryption algorithms such as ChaCha20 and Twofish to broaden the comparative analysis. Larger experimental datasets covering more file types and significantly larger file sizes can also improve the robustness of the Machine Learning models.

Deep Learning techniques, including Artificial Neural Networks and Long Short-Term Memory (LSTM) models, may be investigated to further enhance prediction accuracy for complex encryption scenarios.

Future research may also consider multi-objective optimization approaches that simultaneously optimize security level, encryption speed, processor utilization, energy consumption, and memory requirements.

Another promising direction is the integration of the proposed recommendation framework into cloud computing environments, Internet of Things (IoT) systems, edge computing platforms, and real-time secure communication applications, enabling intelligent encryption algorithm selection under dynamic operating conditions

The proposed framework can therefore serve as a foundation for future intelligent cybersecurity systems that integrate predictive decision support with cryptographic techniques to achieve adaptive, efficient, and secure data protection.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Breiman, L. (2001). *Random Forests*. *Machine Learning, 45*(1), 5–32.
[2] Friedman, J. H. (2001). *Greedy Function Approximation: A Gradient Boosting Machine*. *The Annals of Statistics, 29*(5), 1189–1232.
[3] National Institute of Standards and Technology (NIST). (2001). *Advanced Encryption Standard (AES)* (FIPS PUB 197). U.S. Department of Commerce.
[4] Rivest, R. L., Shamir, A., & Adleman, L. (1978). *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. *Communications of the ACM, 21*(2), 120–126.
[5] Stallings, W. (2017). *Cryptography and Network Security: Principles and Practice* (7th ed.). Pearson.
[6] Schneier, B. (1996). *Applied Cryptography: Protocols, Algorithms, and Source Code in C* (2nd ed.). John Wiley & Sons.
[7] Katz, J., & Lindell, Y. (2021). *Introduction to Modern Cryptography* (3rd ed.). CRC Press.
[8] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
[9] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning* (2nd ed.). Springer.
[10] Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly Media.
[11] Alpaydin, E. (2020). *Introduction to Machine Learning* (4th ed.). MIT Press.
[12] Murphy, K. P. (2022). *Probabilistic Machine Learning: An Introduction*. MIT Press.
[13] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
[14] Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.
[15] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.
[16] Tanenbaum, A. S., & Wetherall, D. J. (2011). *Computer Networks* (5th ed.). Pearson.
[17] National Institute of Standards and Technology (NIST). (2023). *Lightweight Cryptography Standard (Ascon)*. U.S. Department of Commerce.
[18] Chollet, F. (2021). *Deep Learning with Python* (2nd ed.). Manning Publications.
[19] Raschka, S., Liu, Y., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine Learning with PyTorch and Scikit-Learn*. Packt Publishing.
[20] Sarker, I. H. (2021). *Machine Learning: Algorithms, Real-World Applications and Research Directions*. *SN Computer Science, 2*, 160.
[21] Shafique, A., Mehmood, A., Alawida, M., Khan, A. N., & Khan, A. U. R. (2022). *A Novel Machine Learning Technique for Selecting Suitable Image Encryption Algorithms for IoT Applications*. Wireless Communications and Mobile Computing, 2022, Article ID 5108331. https://doi.org/10.1155/2022/5108331
[22] Yuan, K., Huang, Y., Du, Z., Li, J., et al. (2024). *A Multi-layer Composite Identification Scheme of Cryptographic Algorithm Based on Hybrid Random Forest and Logistic Regression Model*. Complex & Intelligent Systems, 10, 1131–1147.
[23] Bhagat, V., Kumar, S., Gupta, S. K., & Chaube, M. K. (2023). *Lightweight Cryptographic Algorithms Based on Different Model Architectures: A Systematic Review and Futuristic Applications*. Concurrency and Computation: Practice and Experience, 35(1), e7425.
[24] Chen, J., Wu, D., & Xie, R. (2023). *Artificial Intelligence Algorithms for Cyberspace Security Applications: A Technological and Status Review*. Frontiers of Information Technology & Electronic Engineering, 24(8), 1117–1142. https://doi.org/10.1631/FITEE.2200314