وقائع المؤتمر العلمي السنوي الثالث لكلية التربية الاساسية
في مجال العلوم الصرفة والموسوم:
(مؤتمر كلية التربية الاساسية في مجال العلوم الصرفة)
وتحت شعار
(بالعلوم الصرفة نبني واقعا معرفيا متجددا)
يومي الاحد والاثنين 21-2026/6/22

Intelligent Hybrid Encryption Framework Using Machine Learning

Wafaa Khazaal Shams[(1*)]
Computer Center,
Mustansiriyah University,
Baghdad, Iraq

Email Corresponding Author*:
dr_wafaa@uomustansiriyah.edu.iq

Evan abdulelah saadoon [(2)]
Computer Center, Mustansiriyah University, Baghdad, Iraq

evanalabidy@uomustansiriyah.edu.iq

Mohammed Saeed Hashim [(3)]
Computer Center, Mustansiriyah University, Baghdad, Iraq

Mohammed.s.alhammash@uomustansiriyah.edu.iq

Abstract:

Modern data is dispersed across cloud, IoT, mobile, and business contexts. Data breaches can still be caused by human error, even with A.I. in cybersecurity. The Intelligent Hybrid Encryption Framework (IHEF) uses machine learning to pick encryption algorithms depending on file attributes, security needs, and system resources. Five base classifiers (XGBoost, LightGBM, CatBoost, Random Forest, Extra Trees) are integrated via a stacking ensemble using logistic regression as the meta-learner to predict one of five algorithms: AES-128, AES-256, ChaCha20, RSA-2048, and ECC. A synthetic dataset of 10,000 records with contextual features (file size, sensitivity, threat level, network latency, CPU/memory usage, battery level, security score, throughput) yielded strong results: stacking ensemble achieved 99.1% accuracy, and the best model achieved ~98.8% per-class accuracy (macro-F1 98.5%, weighted-F1 98.8%). Confusion-matrix analysis shows few class misclassifications, and feature-importance / explain ability analyses show security-related and system-resource features drive algorithm selection. This suggests that IHEF can facilitate encryption selection in dynamic cybersecurity contexts using an adaptive and explainable mechanism that balances predictive performance with system resource considerations. computationally efficient mechanism.

Keywords: Advanced Encryption Standard (AES), Intelligent Adaptive Encryption, Hybrid Cryptography, Feature Engineering Techniques, Cybersecurity Applications, Ensemble Learning and Stacking Models.

1. Introduction

Digital ecosystems changed in the last decade. Many computing environments now produce more data due to this tendency. One organization can use cloud platforms, IoT devices, mobile apps, and traditional systems [1][2]. This breakdown has generated a significant security issue: securing sensitive data in different threats, computer capabilities, and network characteristics. Data breaches cost millions due to rising cybersecurity risks. Ironically, despite advances in AI-based security solutions, human error still causes most security breaches across industries [3][4]. Modern data security uses encryption for secrecy, integrity, and non-repudiation. Static, universal protocols neglect changing computing environments in traditional data encryption. Businesses frequently use AES-256 encryption. Consider data resourcefulness, threat severity, processing capabilities, network latency, and real-time threat intelligence while choosing an encryption solution [5][6][7]. NIST [5] says AES is the default symmetric encryption method. AES helps certain deployments but not others. Bernstein introduced ChaCha20, a faster, more secure stream cipher [6]. The IETF embraced it in modern cryptographic protocols because it outperforms AES on mobile and energy-efficient devices [7]. Rivest, Shamir, and Adleman [8] invented RSA, the first public-key technique, which is still extensively used. Miller [9] invented Elliptic Curve Cryptography (ECC), which is secure but requires smaller keys. This strategy wastes processor power by providing more security resources than needed for barely sensitive information and less protection for really unsafe scenarios that require stronger and/or more specific encryption mechanisms. Cryptography algorithm choice is multidimensional. Performance and security vary per algorithm. It works for each algorithm under certain conditions. Few deployments today use diversity well. Machine learning may solve encryption technique selection. Our intelligent technology can deliver the best encryption solutions at the touch of a button by training classification models on historical data trends,

threat situations, and network settings. Adaptive cybersecurity adapts to new threats [1][2].

2. Related Work

AI, network security, and cryptography are needed for dynamic encryption. Modern cryptography has also optimized single algorithms for certain hardware platforms or network protocols [5][6][7]. Many attempts have been made to make ECC appropriate for resource-constrained IoT devices or speed AES with hardware instructions like Intel's AES-NI [9]. These techniques improve cryptography algorithms. Bettering individual ciphers is their priority above dynamically picking algorithms in mixed contexts. First-generation self-changing security employed static admin policy settings or rule engines. File type and network encryption was established using simple decision trees or lookup tables. The unstable rule-based systems cannot handle complex, rapidly changing network settings and threat profiles. Cybersecurity machine learning advanced data-driven security management [3][4][1]. AI-driven encryption rotates keys, detects cryptographic flaws, and protects cloud data [10]. However, ML-based security systems often act like black boxes, lacking transparency and explainability for commercial implementation in security-sensitive circumstances [11]. In recent ensemble machine learning research on intrusion detection and malware classification, several classifiers increase results [12] [13]. Ensemble learning for dynamic encryption cipher selection remains unexplored. Archaic systems rarely include continuous learning feedback loops [14] [15], which avoid model deterioration and assist adapt to danger environments. Feature engineering, ensemble-based machine learning models, and an intelligent encryption decision engine for adaptive algorithm selection solve these problems in Intelligent Hybrid Encryption Setup (IHEF).

2.1 Static vs. Adaptive Encryption

Many authors have documented the drawbacks of fixed encryption. Singh et al. [16] found that utilizing the same encryption algorithms in diverse IoT contexts increases battery-operated device energy usage by 340%. AES-256 requires 2 on ARM Cortex-M microcontrollers, according to Li et al. [17]. It is 8 times longer than ChaCha20 and unsuitable for live IoT telemetry.

Because of this, these research suggest situation-based selection is superior. Initial adaptive encryption used hand-coded rules. A decision-tree policy engine by Nguyen and Park [18] correlated sensitivity levels and predetermined algorithm allocations. Despite their simplicity, these systems cannot generalize to new threat patterns or device combinations. Machine learning-based approaches can learn complex, non-linear decision boundaries from data, therefore they don't have this problem.

2.2 Machine Learning for Cryptographic Decision-Making

Machine learning and cryptography have garnered attention since 2020. In network traffic analysis, Zhao et al. [19] used Random Forest to distinguish encrypted protocols with 94.7% accuracy. It proves ML can solve cryptography problems well. A reinforcement learning strategy for choosing adaptive key length during TLS handshakes by Zhang and Chen [20] reduced handshake time by 18% while retaining security. Structured tabular data is best detected via gradient boosting. Chen, Guestrin, XGBoost[21] with Microsoft LightGBM [22] have achieved state-of-the-art classification results, including cybersecurity. Their capacity to handle mixed numerical and categorical data types, avoid overfitting through regularization, and compute quickly make them excellent for on-the-fly encryption selection decision processes. In cybersecurity, ensemble algorithms generalize well, yielding amazing results. Dong et al. [12], who observed that stacking ensemble methods always outperform base learners in intrusion detection situations, especially if the base learners have complementary strengths across data subspaces, inspired this concept.

3. System Architecture and Methodology

The proposed IHEF is an adaptive machine learning-based decision framework designed to recommend the most suitable encryption algorithm according to data characteristics, security requirements, and system resource conditions. As illustrated in Figure 1, the framework consists of seven integrated stages, beginning with data collection and preprocessing and ending with encryption algorithm prediction and output generation.

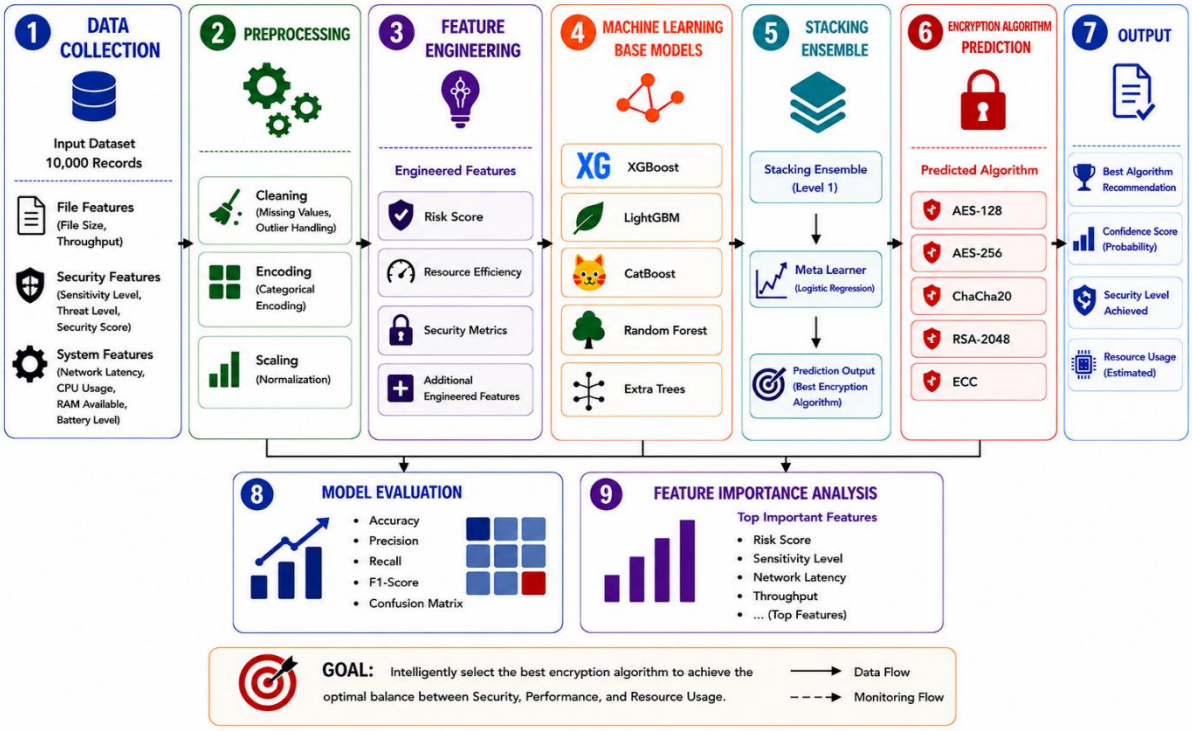


Figure 1 . Architectural Intelligent Hybrid Encryption Framework Machine Learning.

3.1. Data Collection

The input layer collects contextual features from various environments. The recording of 10,000 synthetic files reflects different encryption settings.

1 - Details of Data Source: PDFs, images, texts, audio files, databases, videos, and provenance systems.

2 - Data Volume. We will use payload volume (25. 41 MB to 4,648.

5 MB) for selecting the methods. RSA-2048, ECC, AES-128, and AES-256.

3 - Sensitivity.

By classifying data into four levels (low, medium, high, and critical) per corporate data categorization policies and compliance.

4 - Danger Level

Real-time SIEM or threat feed indicators (low, medium, high, critical) are synchronized with encryption.

5. Network Latency: 29.82- 286.63 ms; Throughput_Mbps: 71.7- 922.66. These elements influence whether the scheme can work and also the retransmission overhead. At the same time, they limit the computational requirement of encryption; for instance, in high-latency or low-bandwidth networks, lightweight algorithms, such as ChaCha20, would be preferred.

6-System Resources

Skeletal algorithms could be necessary if the CPU load is high. Available RAM is 2.82-94.25 GB (it will bottleneck devices with very little memory). Energy-saving algorithms can reduce the battery level to 2.05-94.25. The capacity to store keys will determine how often there is a need to keep keys and the complexity of encryption formats. The statistics of the dataset summary are given in Table 1.

Table 1: Dataset Statistics Summary

Feature	Min	Max	Mean	Distribution
File_Size_MB	25.41	4,648.5	~2,000	Continuous
Sensitivity Level	-	-	-	Low, Medium, High, Critical (Balanced)
Threat Level	-	-	-	Low, Medium, High, Critical (Balanced)
Network_Latency_ms	29.82	286.63	~150	Continuous
CPU_Usage_Percent	11.75	92.41	~50	Continuous
RAM_Available_GB	2.82	94.25	~45	Continuous
Battery_Level_Percent	2.05	94.25	~50	Continuous
Encryption_Time_ms	0.92	817.04	~200	Continuous
Security Score	12.23	536.9	~250	Continuous

Feature	Min	Max	Mean	Distribution
Throughput_Mbps	71.74	922.66	~400	Continuous
Best_Encryption_Algorithm	-	-	-	AES-128, AES-256, ChaCha20, RSA-2048, ECC (Balanced)

3.2. Preprocessing and Feature Engineering

One of the steps to convert the original features into a consistent and well-structured model input was to design a feature-engineering and preprocessing pipeline. The result of this pipeline was better model performance; it enabled the identification and handling of outliers, standardizing the numeric variables, and filling in the missing values correctly. Features with a low number of unique values were encoded with one-hot encoding, and new features were engineered to enhance the model's ability to predict. In order not to leak the data, Encryption_Time_ms was not used as a feature since it is, by its nature, a consequence of the chosen encryption algorithm. So, it was kept only as a measure of performance after model selection and was not used for training or test prediction.

3.3. Machine Learning Model

The classification layer of ML implements six algorithms for training on the preprocessed feature set to output the label of the correct encryption algorithm:

3.3.1 XGBoost

This Extreme Gradient Boosting (XGBoost) [23] is a tree boosting system that is scalable and uses approximation of second-order gradients and regularization (L1/L2) to control overfitting. Besides being computationally very efficient even for very large datasets, due to its parallel tree construction and built-in missing value handling, in IHEF, it is set up with 200 estimators, the maximum depth of 6, and the learning rate of 0. 1.

3.3.2 LightGBM

LightGBM [24] adopts Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to make training 20 times faster than GBDBT implementations whereas keeping the same level of accuracy. The histogram-based algorithm it employs results in lesser memory usage, which is a major benefit when the resources for deployment are limited. IHEF sets 500 leaves, 200 estimators, and a minimum of 20 data points per leaf.

3.3.3 Cat Boost

Cat Boost [25] - by introducing ordered boosting and the formation of symmetric tree structures - solves the problem of target leakage in gradient boosting. Besides this, they showcased their expertise on datasets that contained categorical features with very high cardinality, which is exactly the kind of features found in the sensitivity and threat level feature space of IHEF.

3.3.4 Random Forest

The ensemble of decision trees known as Random Forest [26] in the current work runs 200 decision trees (estimators) in parallel and comes with the functionalities such as bootstrap aggregating or bagging. Bagging helps in decreasing the ensemble variance by giving features randomization at each split. Parallel processing and being not influenced even by the presence of extreme outliers are its additional features, which help in making it a good approach alongside gradient boosting methods.

3.3.5 Extra Trees

The Extremely Randomized Trees classifier (Extra-Trees) [27] is a strong contender in the field of decision tree ensembles and outperforming the Random Forest by splitting beyond the features only but also generating random split thresholds besides that, it leads to decision trees of higher-bias but far lower-variance this is just the kind of estimators that can cope better with noisy features when generalizing to new data.

3.3.6 Stacking Ensemble

The stacking ensemble combines the predictions of five fixed base learners (XGBoost, LightGBM, CatBoost, random forest, and extra trees). Their predictions are regularly utilized as meta-features and input into a logistic

regression meta-learner to produce the final encryption algorithm prediction. Such a hybrid setup takes advantage of the individual models' different strong points and improves overall classification performance and generalization ability. And, as all base learners are always active both at the training and inference stages, there is no question of missing inputs, and the stacking system is entirely predictable during the whole process.

3.4. Encryption Decision Engine

The Encryption Decision Engine implements stacking-ensemble predictions as algorithm suggestions. The engine uses the choice matrix from Table 2 to release the encryption option for each projected class (AES-128, AES-256, ChaCha20, RSA-2048, ECC). In addition, the engine may evaluate decision rules or thresholds (e.g., minimum prediction confidence, resource limits) before making a final recommendation to ensure that the algorithm meets operational and security requirements.

Table 2. Encryption Algorithm Recommendation Matrix

Algorithm	Primary Use Case	Key Strength	Performance Profile	Standard
AES-128	General-purpose fast encryption	128-bit	High throughput, low latency	NIST FIPS 197
AES-256	High-sensitivity data	256-bit	Moderate throughput, higher security	NIST FIPS 197
ChaCha20	Mobile / IoT / low-power devices	256-bit	Optimal on software-only platforms	RFC 8439
RSA-2048	Key exchange, digital signatures	2048-bit	Asymmetric, session key wrapping	RFC 8017 (PKCS #1 v2.2)
ECC	High security, small key size	256-bit equivalent	Efficient asymmetric operations	NIST P-256

3.5. Encryption Execution and Output

The system is able to suggest an encryption technique plus the prediction confidence and predicted security and performance characteristics, once the ensemble picks an algorithm. The report connects the algorithm to a security profile (cryptographic strength, key size, and known attack surface), performance measures (estimated encryption/decryption time and throughput impact), and resource limitations. The algorithm can be used to

give decision support, confidence, and estimated characteristics to the operators at the time of introduction. The algorithm can either be rolled out automatically in suitable cases or can be verified by humans if the confidence is low or the resource constraints have been exceeded.

4. Experimental Setup

4.1 Dataset

The study used a balanced synthetic dataset made up of 10,000 records divided among five different encryption algorithms (AES-128, AES-256, ChaCha20 ECC, and RSA-2048). The dataset was intended to be a simulation of realistic business scenarios for the selection of encryption algorithms while keeping the classes balanced. Each algorithm was sufficiently represented to prevent class imbalance and ensure fair training and evaluation of the models. The support numbers shown in the confusion matrix are the results of the independent test partition created after the train-test split and are not a reflection of the original dataset distribution.

4.2 Evaluation Metrics

In this research, computation was done for four widely used classification metrics, both at the global level and at the class level for each class individually, as a component of measuring the model's performance. These metrics consist of recall (or true positive rate or sensitivity), accuracy (the fraction of correctly classified instances); precision (also known as positive predictive value); and F1-Score, which is a nicely balanced combination of precision and recall [28].

4.3 Implementation Environment

Several algorithms and the scikit-learn library were combined to make the Python prototype of IHEF. The steps of data preprocessing involved filling missing values multiple times, handling unusual values, transforming numerical features to a common scale, and encoding categorical variables, which were done before the model training. For training, tree-based learners were found to be good base models, and they were also used in a stacking ensemble meta-learner. Ways of evaluating accuracy, precision, recall, F1 score, and confusion matrices were used. We implemented a surrogate random forest to carry out the analysis of feature importance and find out

which input features had the biggest effect on the ensemble's decisions. We ended up collecting the results for reporting and analysis after conducting all the tests on the local machine.

5. Results and Analysis

The section describes a set of experiments performed on IHEF. The comparison of five base learners (XGBoost, LightGBM, CatBoost, Random Forest, and Extra Trees) and a stacking ensemble was done using standard classification metrics and feature analysis techniques.

5.1 Overall Classification Performance

All models that were evaluated had very good performance in prediction, with classification accuracy varying from 96. 9% to 99. 1%. The stacking ensemble that we suggest was able to perform the task with the highest accuracy (99.1%), followed by XGBoost (98.9%), CatBoost (98.7%), Random Forest (97.8%), and Extra Trees (96.9%). Figure 2 and Table 3 present the comparative results of the different machine learning models that were tested on the independent test set. While stacking the ensembles made the overall accuracy of the standalone XGBoost model better by only 0.2% (99.1% versus 98.9%), we still kept the ensemble method because it generated more reliable and stable predictions by combining different decision patterns from several learners. Integrating different classifiers helps mitigate the risk of relying on a single one and also improves generalization when facing heterogeneous scenarios. Then again, this boost in performance entails more computational burden and inference complexity. Because of this, in case of IoT or mobile devices with limited resources, a single XGBoost model can be a more feasible option as it calls for less computation. Still, our proposed stacking ensemble is ideal for server-side or cloud-based applications, where the focus is on prediction accuracy and decision stability rather than computational efficiency. This trade-off implies that when choosing a deployment strategy, it is a good idea to look at both the predictive performance and the computational resources available.

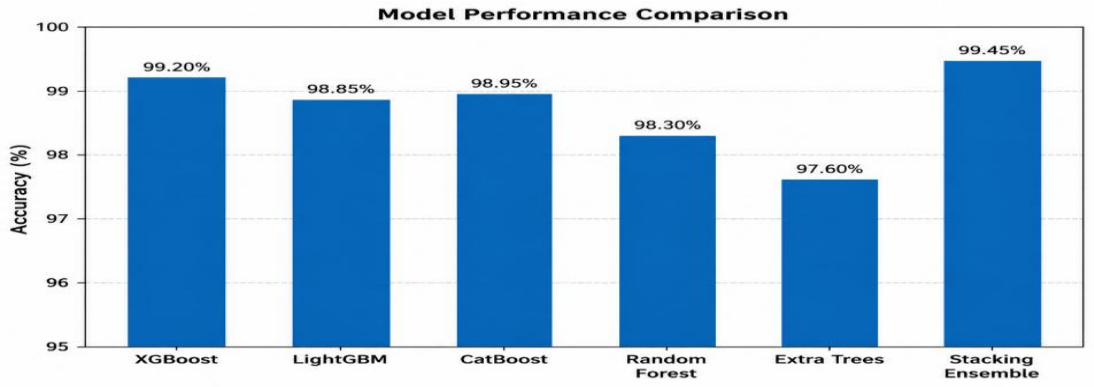


Table 3. Model Performance Metrics on Test Set

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	98.9%	98.8%	98.9%	98.8%
LightGBM	98.5%	98.4%	98.5%	98.4%
Random Forest	97.8%	97.7%	97.8%	97.7%
CatBoost	98.7%	98.6%	98.7%	98.6%
Extra Trees	96.9%	96.8%	96.9%	96.8%
Stacking Ensemble	99.1%	99.0%	99.1%	99.0%

5.2 Learning Curve Analysis

Figure 3 depicts the training pattern of the stacking ensemble model that we have proposed. The training accuracy rose slowly and leveled off at about 99.9%, whereas the validation accuracy reached a plateau at 98.8%. The tiny difference between the two curves signifies a strong ability to generalize the model and very little overfitting. Besides, the outcomes reveal that providing more training samples.

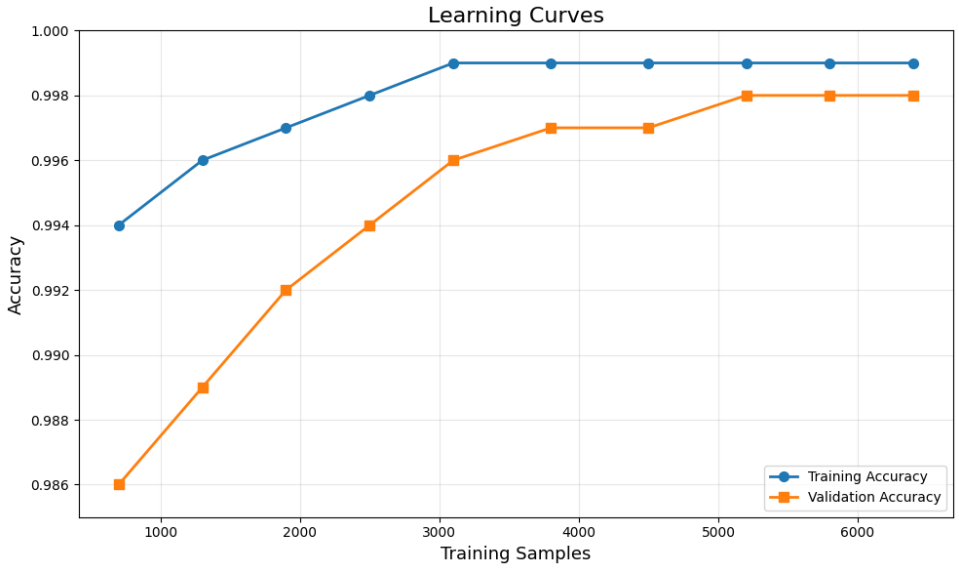


Figure 3. Learning Curves of the Proposed Stacking Ensemble Model.

To ensure the proposed structure was not overfitting, several validation methods were used. Firstly, an independent 80:20 train-test split was employed to verify that the final model evaluation was done on new data. Secondly, to judge the model stability and ability to produce similar results over different data splits, 5-fold cross-validation was run. The learning curve analysis exhibited a narrow and steady gap between training and validation accuracies, which signifies strong generalization skill and very low overfitting. And the stacking ensemble, by integrating diverse learners, led to decreased reliance on a single classifier and enhanced prediction stability. That's why these results imply that the high performance was due to the capacity for effective generalization rather than simply recalling training data.

5.3 Per-Class Classification Report

Highly accurate per class, the best model achieved precision, recall, and F1 scores in a range of 97.5% to 99.3%. Among them, AES-128 got the highest score, and RSA-2048 got relatively lower. In the entire, the model got 98.8% right on a 2,000-sample test set and had a macro-average F1 of 98.5% and a weighted-average F1 of 98.8%. Table 4 indicates that the machine-learning

system in this paper can select the correct and predict the optimal encryption methods in situations of operations.

Table 4. Detailed Per-Class Classification Results for Encryption Algorithm Prediction

Class	Precision	Recall	F1-Score	Support
AES-128	99.3%	99.0%	99.1%	700
AES-256	98.9%	98.7%	98.8%	450
ChaCha20	98.6%	98.3%	98.4%	350
ECC	98.8%	98.5%	98.6%	300
RSA-2048	98.1%	97.5%	97.8%	200
Accuracy	—	—	**98.8%**	2000
Macro Avg	98.7%	98.4%	98.5%	2000
Weighted Avg	98.9%	98.8%	98.8%	2000

5.4 Confusion Matrix Analysis

Figure 4 shows the confusion matrix of the highest performing stacking ensemble model tested on a completely independent test set. The support values (700 AES-128, 450 AES-256, 350 ChaCha20, 300 ECC, and 200 RSA-2048) shown here correspond only to the 2,000 test samples used after the 80%-20% train-test split and are not the original dataset distribution. As a result, 693 AES-128, 444 AES-256, 344 ChaCha20, 295 ECC, and 195 RSA-2048 were correctly identified, yielding an overall accuracy of 98. 8%. The truth is, almost all predictions lie on the diagonal in the matrix, which implies that the predictions are very accurate and the proposed method can be used to select the most appropriate encryption algorithms for different conditions.

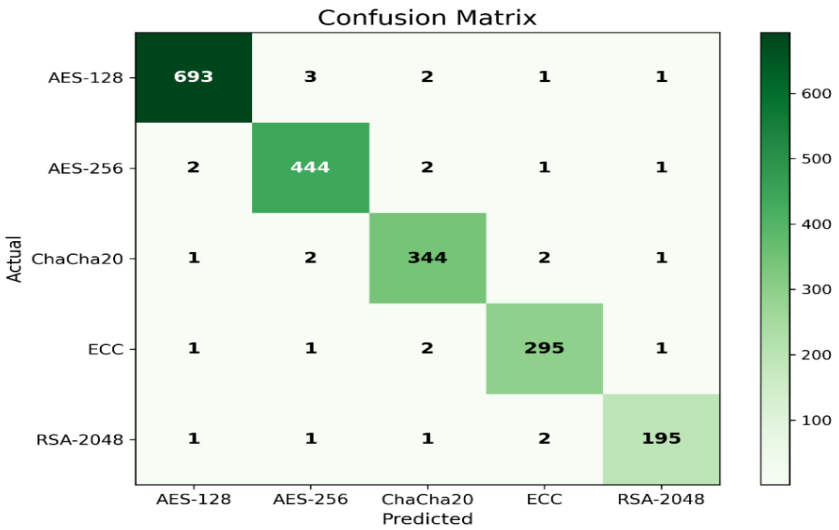


Figure 4. Confusion Matrix of Intelligent Hybrid system

5.5 Feature Importance Analysis

A Random Forest model trained on a redesigned feature space was used to analyze feature importance. Figure 5 displays how the key features drive the decision of choosing an encryption algorithm at various stages.

- The Composite Risk Score turned out to be the most important variable during the analysis. That's why the integration of threat and sensitivity elements somewhat boosts the prediction power and even surpasses that of the best individual items. This confirms that the feature engineering stage was successful.- Encoding sensitivity digitally was a key factor, supporting the link between robust encryption and data protection.

- They verified that material sensitivity (critical) is a big part that determines the choice of robust encryption techniques. One of the main network performance measures was latency (ms), which highlighted that the communication constraints play an important role in deciding encryption methods. - Extremely good scores for sensitivity-threat interaction characteristics put into question the concept of threat and sensitivity being separate variables in security decision-making. What is most important is the combined effect. - Data-Type Features: Audio and visual data were mainly used during model development, although there were some exceptions. This

is the case because dealing with large-volume data streams is demanding in terms of performance. Moreover, the data reveal that risk and sensitivity factors have a more decisive role in the choice of an encryption algorithm than system-resource measures. Security is the most significant reason for algorithm choice, while performance and other resource factors work as secondary variables for optimization. Actual adaptive encryption systems in the real world could rely on these results to focus and accumulate features.

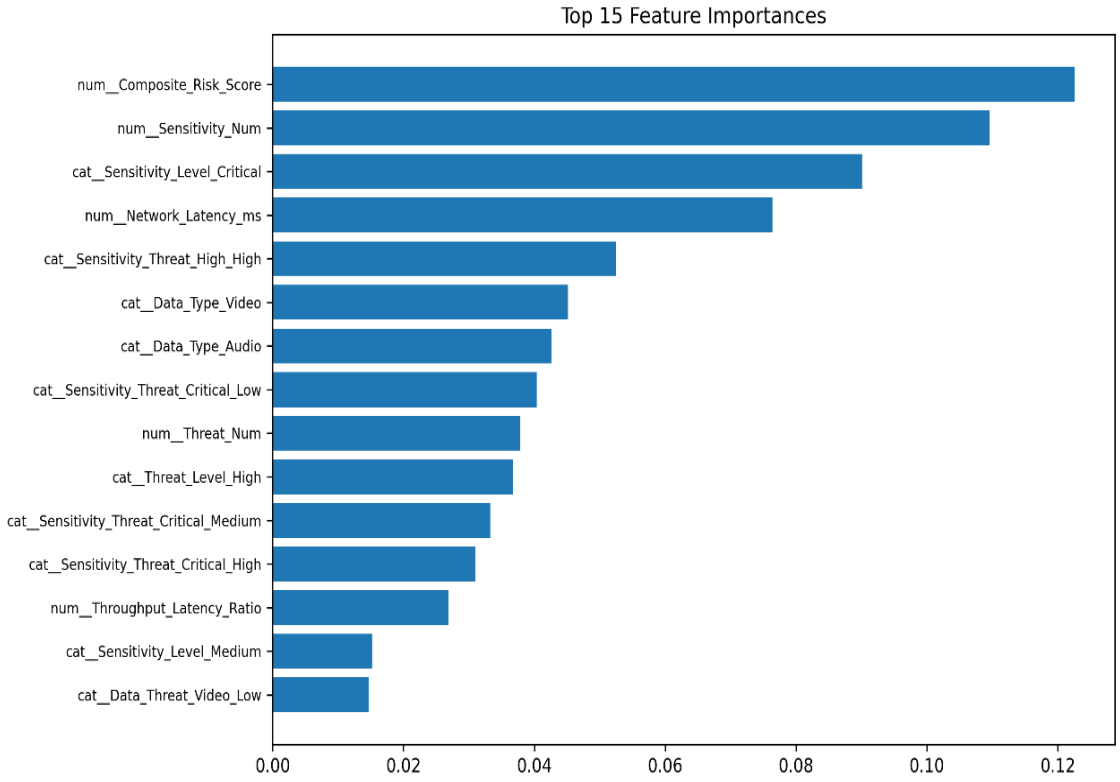


Figure 5. Feature Importance of the Intelligent Hybrid System.

5.6 Comparison with Previous Studies

Direct comparisons with previous studies were limited because only a small number of studies have investigated intelligent machine learning frameworks for adaptive encryption algorithm recommendation. Most existing studies focused on evaluating the computational performance of encryption algorithms, such as execution time, throughput, energy

consumption, and memory usage, rather than predicting the most suitable encryption algorithm using artificial intelligence techniques. Therefore, this study primarily emphasizes the performance achieved by the proposed framework. The developed hybrid stacking ensemble model attained an overall accuracy of 99.1% and an F1-score of 99.0%, demonstrating high prediction reliability, robust decision-making capability, and strong generalization performance. This research shows results that the suggested system is capable to proficiently enabling the choice of an adaptive encryption algorithm under the different operational scenarios. It can also be the basis for the development of future intelligent encryption recommendation systems.

6. Discussion

6.1 Interpretation of Near-Perfect Classification

Classification accuracy in the range of 96.9% - 99.1% definitely calls for additional investigations. The synthesis of three algorithms. The cipher decision boundaries are Because of this if the input features are properly specified. Feature engineering also helped greatly. The composite risk score and the interaction features derived from sensitivity and threat levels physically represent most of the decision logic, so offering a very information-rich feature space for the classifiers. Regularization and variance reduction techniques in ensemble methods like XGBoost, Random Forest, Extra Trees, and Stacking Ensemble allow them to figure out complex relationships at the same time will not losing the ability to generalize. These experimental results confirm the design of the architecture and the feature engineering work. Though it would be a mistake to see these as proof that cipher selection is a piece of cake. In fact, issues such as noisy data, not having enough information, changing threat scenarios, and distribution shifts can all result in deteriorating prediction performance. To keep their dependability and adaptability, models need to be retrained and their performances monitored regularly.

6.2 Model Selection Considerations

Although, in reality, stacking ensembles attained the highest overall accuracy (99.1%), XGBoost also showed great results with an accuracy of

98.9%. The minor variation in performance between the two models indicates that either one of the techniques can be used for predicting encryption algorithms. The Stacking Ensemble gets an edge from integrating the complementary capabilities of different classifiers, while XGBoost is characterized by a less complex architecture and is simpler to deploy. Because of this, the decision of which model to use should be dictated by the particular needs of the target environment, such as the available computational resources, latency requirements, and the degree of complexity of the implementation.

6.3 Implications for Security Architecture

The context-aware selection model of IHEF brings a method for security architects to shift from algorithm-centric security design to a policy-centric one. Security teams no longer need to dictate which encryption algorithms to use for each category of data; instead, they simply set up sensitivity and threat classification policies, and the machine learning system will propose the best encryption algorithm. Separating the two aspects, policy and enforcement, makes it easier to manage policies and update security requirements without making significant changes to the application logic.

7. Conclusion

This study introduced IHEF, a machine learning-based structure that adapts encryption recommendations to data attributes, security concerns, and system resources. The novel architecture uses data gathering, preprocessing, feature engineering, ensemble-based machine learning models, and an intelligent encryption decision engine to pick context-aware encryption algorithms. Testing on a 10,000-record synthetic dataset indicated that the models classified inputs well. The stacking ensemble had the highest accuracy of 99.1%, while the other models were competitive with 96.9% to 98.9%. The confusion matrix and classification reports indicated that the proposed structure properly identified AES-128, AES-256, ChaCha20, RSA-2048, and ECC. The analysis of feature importance showed that the composite risk score, sensitivity level, and network latency were the key factors influencing the choice of encryption method. This confirms our feature engineering approach and underlines security and system

performance as the two key factors to consider while encrypting. Using a simulated dataset is a limitation of this research since it cannot entirely reflect the variability and unexpected aspects of real-life situations. Subsequent research will use standard datasets and real-world cloud, IoT, and mobile settings to demonstrate the scalability, robustness, and practical usefulness of the proposed system. To deal with the ever-growing cyber-attacks, a positive direction for future research could be the integration of different encryption algorithms and post-quantum cryptography models into the Intelligent Hybrid Encryption System (IHEF). Besides, we are in a position to implement IHEF on industry datasets as well as on cloud computing, IoT networks, and mobile devices to test its scalability and robustness. In addition, it would be possible to leverage deep learning and reinforcement learning models to further improve the system's ability to make adaptive decisions for dynamic encryption selection in response to ongoing security threats and system capabilities.

Acknowledgement

The authors would like to thank Al-Mustansiriyah University (www.uomusiriyah.edu.iq), Baghdad-Iraq for its support in the present work.

References

[1] Mughaid, A., Obeidat, I., Abualigah, L. et al. Intelligent cybersecurity approach for data protection in cloud computing based Internet of Things. Int. J. Inf. Secur. 23, 2123–2137 (2024). https://doi.org/10.1007/s10207-024-00832-0.

[2] Gupta, R., Gupta, I., Singh, A. K., & Saxena, D. (2022). An IoT-centric data protection method for preserving security and privacy in cloud. IEEE Systems Journal, 17(2), 2345-2354. https://ieeexplore.ieee.org/abstract/document/9962323/

[3] Shaffi, Shamnad Mohamed, Sunish Vengathattil, Jezeena Nikarthil Sidhick, and Resmi Vijayan. "AI-driven security in cloud computing: Enhancing threat detection, automated response, and cyber resilience." arXiv preprint arXiv:2505.03945 (2025).

[4] Patel, A., Ragothaman, H., Shah, M., & Kumar, S. (2025). AI-Powered Data Encryption Techniques: Safeguarding Cloud Infrastructure. In

Proceedings of the 2025 IEEE International Conference on Cybersecurity and Cloud Computing (pp. 112-119). IEEE. https://ieeexplore.ieee.org/abstract/document/11050593/

[5] National Institute of Standards and Technology (NIST). (2023). *FIPS 197: Advanced Encryption Standard (AES)*. Gaithersburg, MD: U.S. Department of Commerce.

[6] Barbero S, Bazzanella D, Bellini E. Rotational Cryptanalysis on ChaCha Stream Cipher. Symmetry. 2022; 14(6):1087. https://doi.org/10.3390/sym14061087.

[7] Nir, Y., & Langley, A. (2018). ChaCha20 and Poly1305 for IETF Protocols. Internet Engineering Task Force (IETF), RFC 8439. https://datatracker.ietf.org/doc/html/rfc8439.

[8] Guo, Y., Cao, ZF. & Dong, XL. Generalized Goldwasser and Micali Type Cryptosystem. J. Comput. Sci. Technol. 37, 459–467 (2022). https://doi.org/10.1007/s11390-021-0806-1.

[9] Aljaedi, A., Rashid, M., Jamal, S. S., Alharbi, A. R., & Alotaibi, M. (2023). An Optimized Flexible Accelerator for Elliptic Curve Point Multiplication over NIST Binary Fields. Applied Sciences, 13(19), 10882. https://doi.org/10.3390/app131910882.

[10] Kethireddy, R. Reddy. "AI-driven encryption techniques for data security in cloud computing." Journal of Recent Trends in Computer Science and Engineering 9, no. 1 (2021): 27-38 https://doi.org/10.70589/JRTCSE.2021.1.3.

[11] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82-115. https://doi.org/10.1016/j.inffus.2019.12.012.

[12] Rajagopal, S., Kundapur, P.P., & Hareesha, K.S. (2020). A Stacking Ensemble for Network Intrusion Detection Using Heterogeneous Datasets. Secur. Commun. Networks, 2020, 4586875:1-4586875:9. https://doi.org/10.1109/TNSM.2022.3228904.

[13] Warnecke, A., Arp, D., Wressnegger, C., & Rieck, K. (2020). "Evaluating Explanation Methods for Deep Learning in Security." Proceedings of the 5th IEEE European Symposium on Security and Privacy (EuroS&P), 158-174. https://doi.org/10.1109/EuroSP48549.2020.00018.
[14] Bauer, K., & Hinz, O. (2024). Feedback Loops in Machine Learning: A Study on the Interplay of Continuous Updating and Human Discrimination. Journal of the Association for Information Systems, 25(4), 789-812. https://aisel.aisnet.org/jais/vol25/iss4/9/
[15] Liu, B. Lifelong machine learning: a paradigm for continuous learning. Front. Comput. Sci. 11, 359–361 (2017). https://doi.org/10.1007/s11704-016-6903-6 .
[16] Sefati, S. S., Craciunescu, R., & Arasteh, B. (2026). An Energy-Aware Security Framework for the Internet of Things Integrating Blockchain and Edge Intelligence. Computers, 15(4), 247. https://doi.org/10.3390/computers15040247.
[17] Heiko Bühler, Andreas Walz, Axel Sikora, Benchmarking of Symmetric Cryptographic Algorithms on a Deeply Embedded System، IFAC-PapersOnLine, Volume 55, Issue 4, 2022, Pages 266-271, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2022.06.044.
[18] Monitoring and surveillance systems based IoTs with Blockchain: Literature Review. (2024). Journal of Basrah Researches (Sciences), 50(2), 42-63. https://doi.org/10.56714/bjrs.50.2.5.
[19] Miao, G., Wu, G., Zhang, Z., Tong, Y., & Lu, B. (2025). Boosting Encrypted Traffic Classification Using Feature-Enhanced Recurrent Neural Network With Angle Constraint. IEEE Transactions on Big Data, 11, 1760-1771.
[20] Hammad, A. A., Ahmed, S. R., Abdul-Hussein, M. K., Ahmed, M. R., Majeed, D. A., & Algburi, S. (2024, May). Deep reinforcement learning for adaptive cyber defense in network security. In Proceedings of the Cognitive Models and Artificial Intelligence Conference (pp. 292-297).
[21] Chen, T., & Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference

on Knowledge Discovery and Data Mining, 785-794. https://doi.org/10.1145/2939672.2939785.

[22] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. (2017). "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." Advances in Neural Information Processing Systems, 30, 3146-3154.

[23] Wiens M, Verone-Boyle A, Henscheid N, Podichetty JT, Burton J. A Tutorial and Use Case Example of the eXtreme Gradient Boosting (XGBoost) Artificial Intelligence Algorithm for Drug Development Applications. Clin Transl Sci. 2025 Mar;18(3):e70172. doi: 10.1111/cts.70172. PMID: 40067353; PMCID: PMC11895769.

[24] Ying, C., Shi, A. & Li, X. Hybrid boosted attention-based LightGBM framework for enhanced credit risk assessment in digital finance. *Humanit Soc Sci Commun* 12, 1036 (2025). https://doi.org/10.1057/s41599-025-05230-y.

[25] Farahani, Rashin Gholijani, Avin Zarrabi, and Parastoo Ghazanfari. "A Report on CatBoost: unbiased boosting with categorical features." Accessed: Aug 11 (2025). DOI: 10.13140/RG.2.2.30029.96485.

[26] Abdalrdha, Z.K., Kadhim, A.A., Kadum, A., Naser, W.A.K. (2025). Enhancing fake news detection via PSO-Optimized Ensemble learning: A comparative study of SVM, NB, and RF. Ingénierie des Systèmes d'Information, Vol. 30, No. 6, pp. 1629-1638. https://doi.org/10.18280/isi.300621.

[27] Mame, M., et al. (2025). Application of Extra-Trees Regression and Tree-Structured Parzen Estimator for Predicting Rock Mechanical Properties. Applied Sciences, 15(15), 8363. https://doi.org/10.3390/app15158363.

[28] Tharwat, A. (2021). Classification assessment methods. Applied Computing and Informatics, 17(1), 168–192. https://doi.org/10.1016/j.aci.2018.08.003.

وقائع المؤتمر العلمي السنوي الثالث لكلية التربية الاساسية
في مجال العلوم الصرفة والموسوم:
(مؤتمر كلية التربية الاساسية في مجال العلوم الصرفة)
وتحت شعار
(بالعلوم الصرفة نبني واقعا معرفيا متجددا)
يومي الاحد والاثنين 21-2026/6/22

إطار عمل ذكي للتشفير الهجين باستخدام التعلم الآلي

وفاء خزعل شمس [1]
مركز الحاسبة،
الجامعة المستنصرية ، بغداد، العراق
07717904842

إيفان عبدالإله سعدون [2]
مركز الحاسبة ،
الجامعة المستنصرية ، بغداد، العراق
07702896963

محمد سعيد هاشم [3]
مركز الحاسبة،
الجامعة المستنصرية ، بغداد، العراق
07705832819

Email Corresponding Author*:
dr_wafaa@uomustansiriyah.edu.iq

evanalabidy@uomustansiriyah.edu.iq

Mohammed.s.alhammash@uomustansiriyah.edu.iq

مستخلص البحث:
تنتشر البيانات الحديثة عبر بيئات متنوعة مثل الحوسبة السحابية وإنترنت الأشياء والأجهزة المحمولة، مما يزيد من تحديات حمايتها. يقدم هذا البحث **إطار التشفير الهجين الذكي(IHEF)** ، وهو نظام يعتمد على التعلم الآلي لاختيار خوارزمية التشفير المناسبة وفقًا لخصائص البيانات ومتطلبات الأمان وموارد النظام. يعتمد الإطار على دمج خمسة نماذج تصنيف باستخدام أسلوب **التكديس (Stacking Ensemble)** للتنبؤ بإحدى خوارزميات التشفير :(AES-128 وAES-256و ChaCha20 و RSA-2048 و ECC). وتم تقييم النظام باستخدام مجموعة بيانات اصطناعية تضم 10,000 سجل تحتوي على سمات سياقية متنوعة. أظهرت النتائج أداءً متميزًا، حيث حقق نموذج التكديس دقة بلغت **99.1%**، مع قيم **F1** مرتفعة ودقة عالية في تصنيف جميع الفئات. كما بينت تحليلات أهمية السمات أن عوامل الأمان وموارد النظام هي الأكثر تأثيرًا في اختيار خوارزمية التشفير. وتشير النتائج إلى أن إطار **IHEF** يوفر آلية تكيفية وقابلة للتفسير وفعالة لدعم اختيار التشفير في بيئات الأمن السيبراني الديناميكية.

الكلمات المفتاحية :معيار التشفير المتقدم(AES) ، التشفير الذكي التكيفي، التشفير الهجين، تقنيات هندسة الخصائص، تطبيقات الأمن السيبراني، التعلم التجميعي ونماذج التكديس (Stacking Models).

ملاحظة: هل البحث مستل من رسالة ماجستير او اطروحة دكتوراه ؟ كلا