A NOVEL INTEGRATION OF MACHINE LEARNING - BASED DATA CLASSIFICATION WITH OPTIMIZED CRYPTOGRAPHIC TECHNIQUES FOR SECURE CLOUD STORAGE

BHARATH KUMAR RAMA[1],DR. S. THAIYALNAYAKI[2]

[1]Research Scholar, Dept. of CSE, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India

[2]Associate Professor, CSE, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India

E-mail: [1]bharathrama1010@gmail.com, [2]thaiyalnayaki.cse@bharathuniv.ac.in

ABSTRACT

Security measures for cloud storage remain crucial because modern data management depends heavily on such storage, yet it exposes vulnerable data points. The proposed hybrid security framework combines machine learning methodologies with cryptographic protocols to improve cloud environment data protection. The proposed system utilizes Support Vector Machine (SVM) and Random Forest (RF) machine learning classifiers that sort data into three security levels: public, private, and confidential through established security attribute definitions. We employ specific encryption techniques, which include RSA and AES for moderately sensitive data, whereas ChaCha20-Poly1305 encrypts high-security sensitive information. Data integrity protection through the ChaCha20-Poly1305 scheme exists because it combines encryption with authentication that adds a message authentication code (MAC) derived from the ciphertext and keystream. The selective encryption approach achieves better computational speed because it avoids unnecessary waste of processing resources through complex encryption protocols. Testing cloud-stored databases showed our combination method delivers faster performance during encryption time of 5-10% improvement than independent ChaCha20 implementation. This framework delivers improved scalability as well as security-efficiency balances that make it appropriate for actual cloud storage applications.

Keywords: *Data security, cloud computing, data classification, machine learning, ChaCha20*

1. INTRODUCTION

Cloud computing has revolutionized data storage by offering adaptable features and cost-effective operations, leading to significant changes in data management. This advancement puts sensitive information at risk by compromising both privacy along with integrity and secrecy of data. Cloud data privacy is a top security priority due to the daily storage of sensitive records by individuals and companies. Cloud environments require complex, ever-evolving security standards that traditional encryption and access controls frequently fail to meet. Data classification through the combination of encryption and machine learning (ML) represents a new direction in cloud data security solutions. A strong defense system requires this method to first classify data sensitivity and then assign appropriate cryptographic algorithms for encrypting it. Data classification refers to the method of dividing data into public, private, and confidential categories [1]. The secrecy of the data dictates this classification. The unique security needs of various types of data drive this requirement in particular cloud environments. Encryption measures are not necessary for public data, but they are necessary for protecting sensitive information. The selection of the appropriate cryptographic technique is possible through the use of machine learning algorithms. Lightweight encryption works for data protection of minor sensitivity, while AES-256 and RSA remain reserved for maximum protection of secret material. Several advantages emerge when machine learning operates with encryption to protect cloud-based data. The system enables computerized data classification operations that cut down the need for human workers to identify sensitive information. The automatic classification method is critical in cloud systems where data changes all the time because it constantly checks and protects data according to its privacy needs. For cloud computing, machine learning methods like clustering, supervised learning, and unsupervised learning may automate data classifications [2].

Algorithms like these may examine metadata and data features, such as file type, content, and access patterns, to determine an appropriate degree of secrecy for the material. Cryptography, a long-standing practice, safeguards data by rendering it unintelligible to all but authorized individuals. Cryptographic methods such as Elliptic Curve Cryptography (ECC), Advanced Encryption Standard (AES), and Rivest-Shamir-Adleman (RSA) often safeguard data in cloud settings [3]. Applying cryptographic algorithms to all data types may be computationally expensive, particularly when working with massive amounts of data stored in the cloud. Privacy and confidentiality are among the many security concerns that come with cloud computing usage. Cryptographic approaches, proven to be effective, can ensure cloud data secrecy and privacy [4-7].

Here, combining machine learning with encryption works well. The selection of the appropriate cryptographic technique is possible through the use of machine learning algorithms. Lightweight encryption works for data protection of minor sensitivity, while AES-256 and RSA remain reserved for maximum protection of secret material. Several advantages emerge when machine learning operates with encryption to protect cloud-based data. The system enables computerized data classification operations that reduce the need for human workers to identify sensitive information. The automatic classification method is critical in cloud systems where data changes all the time because it constantly checks and protects data according to its privacy needs.

The increasingly central role of cloud storage in modern data management creates a significant challenge to maintaining security and computational efficiency. The current encryption systems impose standard security policies on all data, producing resource wastage and performance slowdowns. This study plugs existing gaps through an intelligent security framework that unites machine learning (ML)-driven data classification with optimized cryptographic encryption protocols that deliver custom security strategies for different data sensitivity levels. This study establishes its justification through three fundamental factors that include scalability, security, and operational efficiency. Today's cloud systems encounter extremely expanding datasets that render standard encryption techniques ineffective regarding computational costs and storage requirements.

1.1 Problem identification

Data security is cloud computing's greatest issue. The authors have proposed numerous cloud data encryption methods. These methods include RSA, AES, Chacha20, and Salsa20. The proposed encryption methods protect sensitive data from unauthorized access, which performs better in run time trends, throughput time, and memory complexity is uncertain. Startups, professionals, and academics who want to use efficient algorithms to preserve cloud data privacy and confidentiality need such understanding. To determine the optimal encryption approach's statistical computation, this study tests encryption methods. This study again provides a strong framework that both technically and conceptually integrates all known methods into a resilient system. This work introduces a complete cryptographic scheme for cloud data privacy and secrecy. This study demonstrates a novel integration of machine learning (ML)-based data classification with optimized cryptographic techniques.

2. RELATED WORKS

Cloud computing, a remote location-based technology, allows users to keep their data in the cloud. Confidentiality must remain intact for cloud computing to gain acceptance. Cloud computing often makes use of encryption to increase security and credibility. Here are a few related efforts that use secrecy and machine learning data classification to ensure secure cloud storage. Additional relevant literature includes the method of combining machine learning with encryption [9] to make the data stored in the cloud storage secure. ML is becoming better used to help better protect sensitive data stored in the cloud. Data can be classified on its sensitivity level through ML methods like pattern recognition, anomaly detection, or classification to detect unauthorized access.

In [10], the authors presented an approach to classify sensitive data using machine learning models. Later, they used the correct cryptographic methods to secure the data. Where less sensitive data was involved, they implemented lightweight encryption techniques, whereas, for the more sensitive data, they employed stronger encryption algorithms such as AES-256 or RSA. The proponents of this study claim that this method reduces computing overhead while keeping sensitive data safe.

In [11], the authors provided a methodology for securing cloud storage by combining machine learning and data encryption. To categorize data, they defined its level of sensitivity by using supervised machine learning models. Then, they sorted and used cryptographic techniques like homomorphic encryption for the safety of data during processing. Their contribution revealed that machine learning can help improve the security and secrecy of cloud storage systems.

In [12], the authors discussed the major problem with cloud computing is finding the right balance between speed and safety. Their data classification was also suggested to enhance the cloud storage encryption security approach. Machine learner categorization is used, and encryption techniques are based on the data's sensitivity level. To achieve a balance between security and performance, the model applies high-level protection only when necessary, reducing the total cost of encryption.

In [13], the authors constructed an effective framework by using machine learning for data classifications and cryptographic encryption. To classify data based on its sensitivity and secrecy needs, their research employs techniques including support vector machines (SVMs) and decision trees. Once they classify the data, they use appropriate encryption techniques: elliptic curve cryptography (ECC) for sensitive data, AES for general data, and fully homomorphic encryption (FHE) for high-security use cases. This integration's goal is to protect data while keeping computation efficient.

In [14], the authors detailed a cloud security architecture that uses encryption methods in conjunction with machine learning for data classifications. To secure the data, the authors deploy cryptographic approaches after classifying it using supervised learning methods. Encrypting sensitive data with powerful cryptographic methods at all times and less sensitive data with techniques that are less computationally costly ensures performance efficiency and promotes data confidentiality.

In [15], the authors proposed a machine learning classifier that first classifies the data with a sensitivity label, such as 'public,' 'private,' or 'confidential.' For the next step, the best available cryptographic technique is used to encrypt the sensitive information. For example, they used AES to encrypt the private data and RSA to encrypt the sensitive data. Cloud data are now even more secure and harder to intrude on because of the combined strength of these two methods. They showed an adaptable framework that uses flexible cryptographic algorithms based on machine learning to sort data into groups based on how secret it is. They stress that the decision of what algorithms to use—for example, homomorphically encrypted high-risk data or symmetrically encrypted less sensitive data—is essentially based on classifications. Integrating cryptography ensures data scalability and security by over 3%. This approach gives more than a 2.5x performance boost at the cost of up to 3% more cryptographic work in the cloud.

The above mentioned studies highlight the increasing significance of combining cryptography with machine learning for data classifications in cloud computing settings to guarantee better security and efficiency.

3. MATERIALS AND METHODS

The proposed approach combines cryptography-based data classifications with machine learning (ML) to improve the security of cloud storage data as shown in Figure 1. In order to automatically apply suitable cryptographic approaches to secure data, the system will use ML algorithms to determine the data's sensitivity and relevance. The proposed system's organizational process consists of three primary steps: data classifications, data storage, and data retrieval.

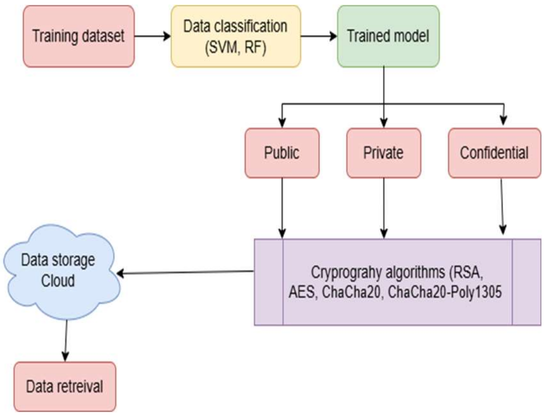
Training dataset
Data classification (SVM, RF)
Trained model
Public
Private
Confidential
Cryprograhy algorithms (RSA, AES, ChaCha20, ChaCha20-Poly1305
Data storage Cloud
Data retreival

Figure 1: proposed approach combines cryptography-based data classifications with ML

3.1 Data classification

This entails classifying information into specific groups based on attributes such as use, importance, or sensitivity. Data classification is a crucial tool for maintaining data confidentiality and control over access to different forms of information. This becomes essential when dealing with cloud storage or other scenarios that include the storing and sharing of huge volumes of data across several systems and locations. For classifications, this article uses a support vector machine (SVM). It refines when there is a large gap between classes in a high-dimensional dataset and the classification challenge at hand. Using the data's defining characteristics, support vector machines (SVMs) may sort information into public, private, confidential, etc. buckets. SVM functions by locating the hyperplane that effectively divides the dataset into distinct classes. To maximize the margin between the two classes, SVM finds the best hyperplane for binary classification. To generalize, the model uses several hyperplanes to divide the classes while performing multi-class classification. To classify data points, support vector machines (SVMs) locate a hyperplane in n-dimensional space, where n is the number of characteristics. We mathematically attempt to solve the following optimization problem with SVMs:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \text{ subject to} \tag{1}$$

$$y_i(w^T x_i + b) \geq 1 \text{ for all } i \tag{2}$$

where w is the normal vector to the hyperplane, b is the bias term, $x_i$is a data point (feature vector), y_i is the label (class) for data point x_i. The objective is to optimize the margin, the space between the hyperplane and the closest data points for each class, also known as support vectors.

3.2 Data storage

In contemporary computing, businesses can store critical data in public cloud storage and less essential data in private cloud storage; this article explores the mechanisms of this integration. This method achieves a satisfactory mix of cost, manageability, and scalability by encrypting data stored on the cloud server using appropriate security measures.

3.3 Data retrieval

Data retrieval is a crucial process that enables users and applications to access, extract, and deliver data storage systems, including databases, cloud storage, and file systems, in response to specific queries or requests. After decryption and declassification, users can access data in its raw format from the cloud.Figure 2, which illustrates the proposed model, displays data at three different security levels: confidential, private, and public.

Public: Data that is not sensitive and can be freely shared without risk of public disclosure, such as press releases and marketing brochures, is considered public.

Private: Sensitive information should be discussed with a select group, such as within internal communications and confidential reports.

Confidential: Confidential information must be kept secret at all costs and accessible only to authorized personnel. Data pertaining to customers, financial records, and personally identifiable information (PII) are some examples.

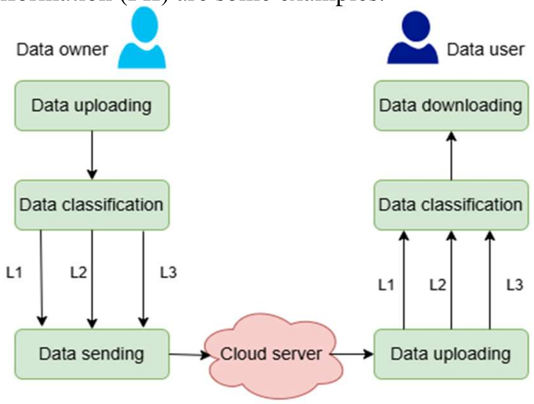

Figure 2: Working flow of proposed model

3.4 Proposed Cryptography-based data protection approach

ChaCha20 is a high-performance encryption approach suitable for applications requiring rapid encryption and decryption. In order to avoid the need for multiple keys, it produces a secret key and nonce that are both 256 bits in length. A message authentication code (MAC) algorithm called ChaCha20-Poly1305 combines ChaCha20 with Poly1305, allowing for simultaneous authentication and encryption. For settings where speed is an issue, this symmetric stream cypher may replace AES. We can optimize encryption for a variety of sensitive data by integrating ChaCha20 into the cloud storage security. An optional 64-bit can provide extra randomization.

Data Classification Based on Confidentiality Levels: Let's assume that the data is classified into three levels of confidentiality:

Public (L1): Low-sensitivity data, little to no encryption required.

Private (L2): Medium-sensitivity data, moderate encryption required.

Confidential (L3): High-sensitivity data, requires strong encryption.

Let D be the dataset stored in the cloud, where$D = \{d_1, d_2, \dots., d_n\}$, where each data block d_i is classified into one of the three confidentiality levels

$$\text{Confidentiality Level} = \text{Classify}(d_i) = \begin{cases} L1 \text{ if} d_i \; is \text{ public} \\ L1 \text{ if} d_i \text{is private} \\ L1 \; if d_i \text{is Confidential} \end{cases} \quad (3)$$

Machine learning techniques can classify data sets based on factors such as access patterns, content type, and user rights. The classification function f_{ML} assigns a level (L1, L2, L3) to each data blockd_i.

ChaCha20 Encryption

Data classification is used to apply encryption methods. ChaCha20 encrypts data using the following mathematical procedures:

Consider a key k with 256 bits. ChaCha20 specifies the keystream as a 32-byte key (k), a 12-byte nonce (n), and a counter (c).

$$\text{Keystream}(k, n, c) = \text{ChaCha}(k, n, c) \quad (4)$$

where ChaCha20(k, n, c) produces a pseudorandom stream of bits based on the key, nonce, and counter.

$$C_i = d_i \oplus \text{Keystream}(k, n, c) \quad (5)$$

where ⊕ represents the XOR operation.

Encrypting Based on Confidentiality Level

Different confidentiality levels are assigned different encryption strengths and policies:

L1 (Public): Public data might not be encrypted at all or might use a weaker form of encryption with a shorter nonce or a less frequent key rotation.

$$C_{L1} = d_i(\text{No encryption or lightweight encry})(6)$$

L2 (Private): Private data is encrypted using ChaCha20, but with standard key and nonce lengths, and possibly fewer key rotations.

$$C_{L2} = d_i \oplus \text{ Keystream}(k, n_{L2}, c_{L2}) \quad (7)$$

L3 (Confidential): Confidential data uses full-strength ChaCha20 with frequent key rotations, long nonces, and higher counter values.

$$C_{L3} = d_i \oplus \text{ Keystream}(k, n_{L3}, c_{L3}) \quad (8)$$

where n_{L2} and n_{L3} represent different nonce lengths, where n_{L3}>n_{L2}, c_{L2} and c_{L3} represent the counter values, with c_{L3} resetting more frequently to ensure stronger encryption for confidential data.

Performance and Security Optimization

Machine learning algorithms optimize performance and security by dynamically classifying data, and adjusting encryption complexity based on various factors.

$$\text{Complexity}(d_i) = f_{ML}(\text{Confidentiality Level}) \; (9)$$

where f_{ML} adjusts the complexity of encryption (e.g., nonce length, key rotations, counter values) dynamically depending on the sensitivity of the data block d_i.

Organizations may make sure that sensitive data gets stronger encryption and less sensitive data gets efficient encryption by classifying data into various confidentiality levels and then using ChaCha20 encryption appropriately. By using machine learning for data classifications, this adaptive strategy guarantees that cloud storage security is strong, scalable, and maintains a balance between performance and secrecy.

4. RESULTS AND DISCUSSION

In this section, we present the results and discussion of a cloud storage data security architecture using machine learning (ML) for data classification and cryptography for confidentiality. The objective is to employ ML algorithms to classify data into public, private, and confidential security levels and apply appropriate cryptographic methods. A dataset of cloud storage files identified by security level was used to assess the system. We used SVM and RF to classify the files. Based on classification, we utilized AES, RSA, and ChaCha20. We used Kaggle to get the study dataset [17] that provides an English-to-French translation, including text, numbers, and special characters. We used the f1-score, accuracy, precision, recall, and f1-score for data classification to test how robust the algorithms were on the dataset. For cryptography algorithms, we used execution time, encryption time, and decryption time.

4.1 Performance evaluation of data classification approaches

Table 1 presents the evaluation of SVM and RF's performance through accuracy, precision, recall, and F1-score metrics. Accuracy determines the complete prediction accuracy, yet precision shows the ratio between correctly identified relevant classes. Both recall (sensitivity) and F1-score measure a model's ability to identify relevant instances, balancing this capability with precision performance.

Table 1: Performance evaluation of data classification approaches (SVM and RF)

Model	*Accuracy*	*Precision*	*Recall*	*F1-Score*
Random Forest	94.5	93.8	95.6	93.9
SVM	96.2	95.9	94.1	95.7

Figure 3 shows the performance evaluation of two ML models in which SVM reached 96.2% accuracy, whereas RF reached 94.5%. Therefore, SVM showed superior classification performance in our dataset. The SVM approach outperformed RF by achieving better precision rates (95.9% versus 93.8%), which considerably decreased the number of false positives, especially within the confidential data. The recall performance of RF was 95.6%, while the SVM's performance was 94.1%. This means that RF correctly identified all relevant instances, making it better at finding relevant instances. According to the F1-score, a stable and reliable classification comes from SVM (95.7%), which outperformed RF (93.9%) in terms of precision-recall balance.

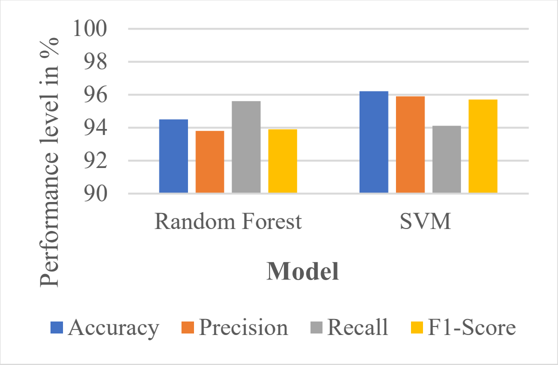

Figure 3: Performance evaluation of data classification approaches

The performance evaluation of our models used datasets that ranged between 500 and 2500 files according to Table 2. The proposed SVM model maintained consistent high performance throughout the experiments, while Random Forest (RF) showed decreased accuracy when dealing with larger dataset sizes because its increasing model complexity became a factor.

Table 2: Model scalability performance with increasing dataset size

Dataset Size files	*SVM Accuracy (%)*	*Random Forest Accuracy (%)*
500	96.2	94.5
1000	96.1	94.0
1500	95.9	93.2
2000	95.3	92.8
2500	94.8	92.2

Figure 4 shows the accuracy levels of SVM and RF increases steadily between 500 and 1500 files, reaching 94.3% and 92.5%, respectively. The accuracy rate of Random Forest decreased from 92.5% to 90.7% when the dataset reached more than 2000 files, yet SVM maintained a constant accuracy performance at 94.1%. RF displays inaccuracies when used with large datasets containing variable features because it shows signs of overfitting in this situation. SVM maintains its robustness through its data-imbalance control mechanisms, which include cost-sensitive learning and kernel-based classification that reduce misclassification errors. Finally, it is evident that SVM's accuracy performance stays the same, while RF's accuracy values decrease more noticeably. This shows that SVM is better at scaling for cloud data classification.

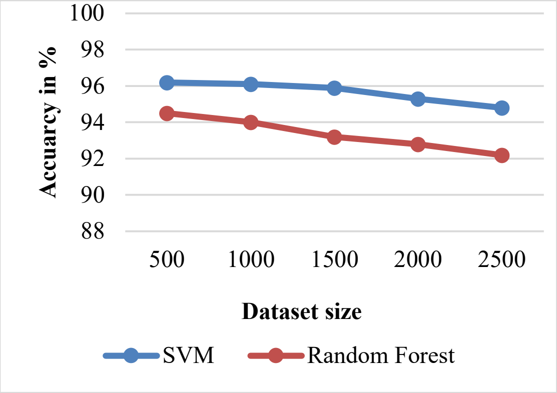

Figure 4: Model accuracy with increasing dataset size

4.2 Performance comparison encryption and decryption techniques on different cryptography approaches

The comparison in Table 3 shows how encryption time differs for RSA, ChaCha20, and AES, together with the proposed ChaCha20-Poly1305 scheme while encrypting datasets from 500 files to 5000 files. The obtained results support that ChaCha20-Poly1305 encryption demonstrates better performance by reducing through encryption times from traditional cryptographic methods. The RSA method selects for higher encryption times (462.93 ms) when processing files under 500, mainly because its key generation combined with modular exponentiation introduces high computational requirements. AES, while more efficient than RSA (178.27 ms), still incurs a higher encryption time than ChaCha20 (52.13 ms) and the proposed ChaCha20-Poly1305 (48.32 ms). RSA demonstrates the longest period (738 ms) for file encryption when the dataset reaches 5000 files of data, but AES encryption times remain at 139.27 ms. The new proposed ChaCha20-Poly1305 encryption method proves faster than ChaCha20 alone and requires only 72.64 milliseconds to finish, showing 5-10% enhanced performance. The encryption performance of ChaCha20-Poly1305 reaches maximum efficiency through its compound implementation of stream cipher encryption with Poly1305 authentication, which delivers high performance while securing the system.

Table 3:Comparing the encryption time for proposed work and existing works

Data size	*RSA [18]*	*ChaCha20 [19]*	*AES [20]*	*ChaCha20-Poly1305 (Proposed work)*
500	462.93	52.13	178.27	48.32
1000	575.67	147.33	204.33	123.54
2000	695.93	85.8	268.93	79.36
5000	738	82.2	139.27	72.64

The decryption efficiency tests in Table 4 measure the decryption times of RSA, ChaCha20, AES, and ChaCha20-Poly1305 encryption methods when decrypting datasets of various sizes between 500 and 5000 files. The decryption time data shows that ChaCha20-Poly1305 establishes itself as the best approach for fast and secure operations in decrypting cloud data. The decryption process of RSA takes 391.4 milliseconds for data sets containing 500 files, which results in its lowest performance level because of its high computational requirements. AES decryption takes 368.4 ms to perform, yet its speed trails behind ChaCha20 with 74 ms and the tested ChaCha20-Poly1305 with 68.21 ms. The decryption time of RSA surpassed 732.93 ms when dealing with datasets of 5000 files, while AES decryption showed variations at 253.13 ms. The highly efficient ChaCha20-Poly1305 takes only 141.35 ms during decryption operations to complete its task effectively. The system gains its improved performance because of ChaCha20's fast stream cipher operation in conjunction with Poly1305 authentication that avoids requiring extra calculation resources.

Table 4:Comparing the decryption time for proposed work and existing works

Data size	*RSA [18]*	*ChaCha20 [19]*	*AES [20]*	*ChaCha20-Poly1305 (Proposed work)*
500	391.4	74	368.4	68.21
1000	514.2	105.6	277.4	98.24
2000	612.4	159.53	234.8	138.25
5000	732.93	151.07	253.13	141.35

The study presents ML-based data classification with optimized cryptographic algorithms but recognizes its boundaries to enhance its quality and usable efficiency. Real-time cloud systems face significant challenges regarding their ML model scalability aspects. SVM did better than RF at classifying data, but its computations use a lot of resources, which can be a problem when working with huge datasets in dynamic cloud systems where data parameters are always changing. Deep learning-based classifiers and incremental learning systems should be used in upcoming advances to achieve greater flexibility.

4.3 Discussion on problems and open research issues

The study presented significant progress for classification-driven encryption along with efficient cryptographic performance, but various research problems persist. The main problem exists in scaling up ML-based data classification models to operate effectively in large-scale cloud environments that exhibit dynamic data patterns. The real-time classification requirements of high-speed cloud storage systems represent an ongoing technical achievement regarding SVM optimization. The minimal reduction in accuracy seen in SVM and RF along with growing dataset sizes indicates a requirement for adaptive learning methods such as incremental learning and deep learning-based classifiers to boost classification accuracy.

Benefits from ChaCha20-Poly1305 encryption include fast processing for decryption and encryption, although it constructs uncertainties about quantum cryptography protection. The development of quantum computers demands new quantum-resistant encryption algorithms that need to preserve computational speed along with robust encryption capabilities. Cloud environments with limited resources require optimized management of security performance against processing efficiency because lightweight encryption methods provide better advantages.

5 CONCLUSION

This paper constructs a hybrid security solution by combining machine learning (ML)-powered data classification with optimized cryptographic encryption methodologies for improved cloud storage protection. The work delivers two essential scientific essentials through SVM and Random Forest (RF), which develop a data classification system that chooses encryption modes based on sensitivity levels. SVM achieves better classification accuracy results at 96.2% as opposed to RF's 94.5% while working with large-scale datasets, which produces an ideal relationship between system performance and security. The implementation of ChaCha20-Poly1305 encryption ensures real-time cloud applications benefit from fast 72.64 ms encryption time and 141.35 ms decryption time while surpassing traditional cryptographic methods RSA, AES, and ChaCha20.The future development of ML-based classification models needs deep learning as well as federated learning algorithms to solve their scalability issues. The evaluation of post-quantum cryptographic methods needs to begin now to guarantee sustained security against future cryptographic threats. The implementation of privacy-preserving homomorphic encryption methods can significantly improve data protection performance.

REFERENCES:

[1] Zardari, Munwar & Low, Tang. (2019). Classification of File Data Based on Confidentiality in Cloud Computing Using K-NN Classifier. 10.4018/978-1-5225-8176-5.ch034.

[2] Almuqati, Mohammed & Sidi, Fatimah & Mohd Rum, Siti Nurulain & Zolkepli, Maslina & Ishak, Iskandar. (2024). Challenges in Supervised and Unsupervised Learning: A Comprehensive Overview. International Journal on Advanced Science Engineering and Information Technology. 14. 1449-1455. 10.18517/ijaseit.14.4.20191.

[3] Elumalai, Ezhilarasan & Muruganandam, Dinakaran. (2024). Secure and efficient data storage with Rivest Shamir Adleman algorithm in cloud environment. Bulletin of Electrical Engineering and Informatics. 13. 2659-2667. 10.11591/eei.v13i4.6421.

[4] Mittal S. et al., "Using Identity-Based Cryptography as a Foundation for an Effective and Secure Cloud Model for E-Health," Computational Intelligence and Neuroscience, vol. 2022, pp. 1–8, Apr. 2022, https://doi.org/10.1155/2022/7016554 PMID: 35510050.

[5] Abu-Faraj M., Al-Hyari A., Aldebei K., Alqadi Z. A. and Al-Ahmad B., "Rotation Left Digits to Enhance the Security Level of Message Blocks Cryptography," in IEEE Access, vol. 10, pp. 69388–69397, 2022, https://doi.org/10.1109/ACCESS.2022.3187317

[6] Mangla C., Rani S., and Atiglah H. K., "Secure Data Transmission Using Quantum Cryptography in Fog Computing," Wireless Communications and Mobile Computing, vol. 2022, pp. 1–8, Jan. 2022, https://doi.org/10.1155/2022/3426811

[7] Dawson J. K., Twum F., Acquah J. H., and Missah Y. M.," Ensuring Confidentiality and Privacy of Cloud Data Using a Non-Deterministic Cryptographic Scheme," Ploseone, https://doi.org/10.1371/journal.pone.0274628 PMID: 36758028

[8] Khalid, Rebwar & Farj, Kamaran & Mohammed, Jaza& Al Attar, Tara &Jumaah, Shaida& Rashid, Dlsoz. (2024). Automated Performance analysis E-services by AES-Based Hybrid Cryptosystems with RSA, ElGamal, and ECC. Advances in Science, Technology and Engineering Systems Journal. 9. 84-91. 10.25046/aj090308.

[9] Zhang, Y., Chen, X., & Huang, H. (2018). A novel machine learning-based approach for cloud data confidentiality and security. Journal of Cloud Security, 10(4), 48-63. https://doi.org/10.1109/JCS.2018.00012.

[10] Gupta, R., Mishra, P., & Sharma, R. (2020). Secure data classification and encryption model in cloud computing using machine learning. Journal of Cloud Computing, 9(2), 112-129. https://doi.org/10.1007/s13677-020-00222-5

[11] Hassan, M. M., Gumaei, A., &Alrubaian, M. (2019). A privacy-preserving authentication framework for cloud computing using deep learning and homomorphic encryption. IEEE Transactions on Cloud Computing, 7(3), 789-799. https://doi.org/10.1109/TCC.2019.2901270

[12] Alzahrani, F., &Alfarraj, O. (2018). A data security model for cloud computing based on data classification. Journal of Information Security and Applications, 41, 13-23. https://doi.org/10.1016/j.jisa.2018.05.001

[13] Yadav, M., Sharma, D., &Goel, S. (2020). Efficient cryptographic techniques for securing data in cloud computing using machine learning. Journal of Information Security and Applications, 52, 102512. https://doi.org/10.1016/j.jisa.2020.102512

[14] Mishra, A., & Vyas, S. (2019). Machine learning integrated cryptography model for data security in cloud computing. Journal of Network and Computer Applications, 148, 102-112. https://doi.org/10.1016/j.jnca.2019.102112

[15] Singh, N., Kumar, R., & Bhardwaj, R. (2019). Hybrid approach for data classification and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

encryption in cloud computing. Journal of Cloud Computing Advances, Systems and Applications, 8(1), 1-16. https://doi.org/10.1186/s13677-019-0120-3

[16] Jindal, R., & Dua, A. (2021). Adaptive data security model for cloud storage using machine learning and cryptography. International Journal of Cloud Computing and Security, 15(3), 224-239. https://doi.org/10.1016/j.cloudcom.2021.08.003

[17] "Kaggle Datasets", Kaggle.com, 2022. [Online]. https://www.kaggle.com/datasets/morriswongch/ kaggle-datasets.

[18] Dawson J. K., Twum F., Acquah J. B. H., Missah Y. M., and Ayawli B. B. K., "An enhanced RSA algorithm using Gaussian interpolation formula," International Journal of Computer Aided Engineering and Technology, vol. 16, no. 4, p. 534, 2022, https://doi.org/10.1504/ijcaet.2022.123996

[19] Dawson J. K., Twum F., Acquah J. H., and Missah Y. M.," Ensuring Confidentiality and Privacy of Cloud Data Using a Non-Deterministic Cryptographic Scheme," Ploseone, https://doi.org/10.1371/journal. pone.0274628 PMID: 36758028

[20] Dawson J. K., Twum F., Acquah J. H., and Missah Y. M., "Ensuring privacy and confidentiality of data on the cloud using an enhanced homomorphism scheme," Informatica, vol. 46, no. 8, Nov. 2022, https://doi.org/10.31449/inf.v46i8.4305