SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I7P130

Volume 12 Issue 7, 390-399, July 2025

Original Article

Bot Detection and Multi-Level Encryption in Multiple Cloud Storage for Recommendation Systems

J. Chitra[1], S. K. Piramu Preethika[2]

[1,2]*Department of Computer Science, Vels Institute of Science, Technology & Advanced Studies, Chennai, Tamil Nadu, India*

[1]*Corresponding Author : chitrakarthik452@gmail.com*

Received: 16 May 2025 Revised: 17 June 2025 Accepted: 18 July 2025 Published: 31 July 2025

Abstract *- The explosive growth of Cloud Computing (CC) and recommendation systems has created novel challenges in maintaining system dependability, privacy, and data security. This study proposes a novel method called Enhanced Encryption and Bot Detection for Multiple to solve these issues. Cloud Storage and efficient Recommendation Systems. Multilevel encryption is used in the suggested system to improve data resilience and secrecy. Each piece of data is encrypted using cutting-edge cryptographic techniques to provide strong security against breaches and unwanted access. The information is safely dispersed over many cloud platforms. To improve the integrity of recommendation systems, the framework also includes an advanced Bot Detection mechanism. This mechanism detects bot-driven actions that may jeopardize the accuracy of recommendations by applying Machine Learning (ML) algorithms. This two-tiered strategy preserves the precision and reliability of recommendation systems while fortifying the security of multi-cloud storage.*

Keywords *- Bot detection, Cloud computing, Multilevel encryption, Multiple cloud storage, Recommendation systems.*

1. Introduction

In recent years, cloud services and encrypted cloud storage systems have become more and more popular because of their accessibility and availability. Cloud storage services are the best place to obtain such information, whether personal or not, because of the large number of users and data stored there. The user might take the easy step of adding a local layer of encryption to maintain total anonymity. This will make it impossible for the cloud provider to decode the information [1]. Multi-cloud storage is the usage of storage solutions from several cloud providers to satisfy the performance, pricing, data management, or compliance requirements of an enterprise. Businesses distribute their information and workloads among two or more cloud service providers, utilizing each one's advantages rather than depending on just one [2]. In cloud settings, multi-level encryption and bot detection are crucial elements of a strong cybersecurity approach.

Multi-level encryption guarantees that sensitive data is safe throughout the whole storage and transmission process, whereas bot detection concentrates on locating and preventing unwanted automated activity. Together, these two technologies can defend cloud systems against data breaches, bot-driven assaults, and other security risks. In multi-cloud setups, recommendation systems are susceptible to a number of risks since they depend on machine learning models, APIs, and sensitive user data. In order to guarantee the integrity, confidentiality, and efficient operation of these systems, multi-cloud storage must incorporate bot detection and multi-level encryption.

1.1. Overview of Multi-Cloud Storage

Cloud storage allows individuals and companies to store, manage, and retrieve data online instead of depending on local on-premises storage options. It is an essential part of modern IT infrastructures because it offers data management scalability, flexibility, and cost-effectiveness. It has emerged as a key component of today's data management, providing both people and businesses with readily accessible, scalable, and affordable alternatives. In contrast with typical storage systems, cloud storage uses dispersed server networks to store data remotely, giving users remote access to their files from any location at any time. The main advantages of cloud storage are

- Scalability: The ability to adjust storage capacity to user demands.
- Cost-effectiveness: Pay-as-you-go schemes save money up front.
- Increased dependability by replication across several servers and locations is known as data redundancy.
- Accessibility: Secure internet connections allow data to be accessed anywhere in the world.

Cloud Service Providers (CSPs) use a Storage-as-a-Service (StaaS) model to provide application providers access to essentially limitless computing and storage capacity while also offering cost-effectiveness and other Quality of

Service (QoS) attributes. Companies also employ hybrid or multi-cloud systems, which combine many public and/or private cloud service providers, to save costs, avoid vendor lock-in, and achieve high availability and performance [3]. Multi-cloud storage is the process of combining storage options from several cloud platforms (such as AWS, Google Cloud, Azure, and others) into a unified storage plan. Utilizing the finest services from several sources helps businesses save expenses, boost resilience, and avoid vendor lock-in.

1.2. Multi-Level Encryption

Multilevel encryption is a system that uses many overlapping encryption techniques to safeguard sensitive data and information. Multiple layers of encryption are added in this system to enhance security and make it more difficult to decipher the data. Several encryption methods are employed on the data, one after the other, while utilizing the multilayer encryption system. For instance, data can be encrypted using digital encryption techniques as a first layer, and then the data that was encrypted is encrypted once again using a different encryption approach [4]. Further, multi-level encryption protects data security at several levels.

- Encrypt user data on the client side before uploading it to the cloud.
- Encrypt data while it is being sent between clouds and via APIs using Transport Layer Security (TLS).
- On the server side, the cloud providers use techniques like AES-256 to encrypt data while it is at rest.
- Field-level encryption is performed on certain user data fields (such as names and email addresses).
- Using encryption for sharding involves dividing the user data into encrypted chunks and saving it on several cloud storage services. The information is still lacking, even if one supplier is hacked [5].

1.3. BoT Threats and Recommendation Systems

Recommendation systems are seriously endangered by bots, which are fabricated programs created to carry out actions on behalf of malevolent people. They have the ability to distort suggestions, falsify interactions between users, evaluations, and ratings, and compromise the system's integrity.

Sensitive user data is maintained and analyzed in dispersed multi-cloud settings by recommendation systems (such as streaming services, tailored advertisements, or e-commerce product suggestions). The challenges faced by the recommendation systems are as follows.

- Harmful Bot Activity: Data and recommendation results can be stolen by bots via scraping techniques. ML techniques can be distorted by fraudulent requests or phoney traffic. Bots have the potential for DDoS assaults, which prevent users from accessing services.
- Compliance and Data Privacy: Regulations such as the CCPA, GDPR, and HIPAA require that user data be protected.
- Management of Distributed Data: Attack surfaces have grown since data is frequently hosted across several cloud providers (such as AWS, Google Cloud, and Azure).
- Managing Encryption Keys: It might be difficult to secure and manage encryption keys across multiple vendors.
- Performance in Real Time: Low-latency performance must be retained while recommendation systems maintain security.

Recommendation systems are seriously threatened by bots, especially in settings that depend on extensive data processing, customised outputs, and interaction from users. The following are some of the types of Bots in recommendation systems [6].

- DataScraping: Bots employ front-end interfaces or recommendation system APIs to get confidential data, including consumer preferences, copyrighted material, and tailored suggestions.
- False User Contacts: In order to influence the recommendation algorithms, bots mimic user actions (such as clicks, searches, or purchases).
- Credential stuffing: Bots obtain sensitive information or personalized recommendations by logging into user accounts using stolen credentials.
- Denial of Service (DoS/DDoS): Bots overload storage infrastructure or recommendation system APIs with queries, causing the system to break down.
- Model poisoning occurs when bots introduce erroneous or modified data into the system (such as phoney reviews, search terms, or interactions) in order to lower the caliber of suggestions.
- API Abuse: Bots take advantage of semi-private or public APIs to get recommendations data or bombard endpoints with queries.
- Competitive Attacks: Rivals use bots to scrape suggestions, tamper with ranks, or find flaws in algorithms.
- Bots that breach a system, encrypt user data or sensitive recommendation models, and then demand a ransom are known as ransomware bots.

Existing options for protecting recommendation systems have a number of drawbacks, even with the notable improvements in bot detection and multi-level encryption. These vulnerabilities may affect how well the systems handle bot attacks and guarantee data security across various cloud storage infrastructures. Because of their automated-like conduct, legitimate users may be labelled as bots. Vulnerabilities might arise because data saved on several cloud platforms could not necessarily adhere to the same encryption standards. It might be difficult and prone to human error to manage encryption keys across several cloud

storage services. Risks like data loss or illegal access might result from improper key management.

Although recommendation systems are at risk of malicious bot attacks and data breaches in multi-cloud environments, existing approaches have not achieved the necessary combination of multi-layered encryption and intelligent options for bot detection. In doing so, they may destroy data privacy and consistency as well as the trusted networks. A viable, symmetry framework is urgently needed to ensure such systems' confidentiality, integrity, and judgment.

Most of the existing approaches focus on encryption or bot detection, often without an integrated approach. Besides these barriers, challenges exist with respect to encryption that engages across a number of task-based cloud platforms, low latency, and detection of advanced bot attacks. Few studies merge deep AI options for bot detection in real-time recommendation systems, to check against underpinning volatility in multi-cloud schemas. The issues will be solved by creating hybrid models to increase accuracy by fusing conventional approaches with AI-powered strategies. In real-time settings, optimize encryption techniques to lower latency and boost efficiency. Further, a unified security framework has to be designed that incorporates encryption and bot detection effortlessly.

1.4. Objectives of the Proposed Framework

- To provide a multi-level encryption approach for recommendation systems security in a multi-cloud setting.
- To utilize ML techniques to improve Bot detection capabilities in recommendation systems to identify and stop bot activity and guarantee that only authorized users engage with the site.

2. Related Works

Cloud security has recently gravitated toward extra secure multi-layered encryption processes and bot detection systems relying upon machine learning to protect the cloud data and cloud systems. Previous studies have reviewed encryption types within cloud storage or addressed AI technologies to identify malicious bots intelligently. This review paper highlights current concepts, limitations, and increased demand for interoperable, scalable systems in a multi-cloud recommendation framework.

Sharma et al. (2024) suggest multi-level encryption as a way to guarantee data security during network transmission. Compared to ordinary encryption, which uses numerous rounds of encryption using identical or different keys to produce a complex or powerful encryption, multi-level encryption protects data. Many solutions have been developed based on Attribute-Based Encryption (ABE), which is widely acknowledged as the best access control technique for protecting the cloud storage environment [7]. Only a small number of schemes for multi-cloud environments fail to incorporate the data security categorization on the cloud side, and most current research is carried out inside a single cloud provider [8].

Kumar et al. (2024) introduce a dynamic identity-based system that minimizes vendor lock-in problems by offering a multi-copy based on several cloud storage servers. For the purpose of storing user data, several copies of multi-homomorphic verified tags are supplied to various cloud storage servers. The K Center Diffie-Hellman (KCDH) technique has been used to show the system's security. Block hashing, merging and data integrity audits for cloud computing are also carried out. Additionally, two Secure Networking Protocols (SNP)-based data centers are offered for the development of effective cloud storage strategies for dynamic data. Using the airport dataset, the performance of the suggested HMAC-Rijndael architecture was assessed, and it was discovered to outperform current encryption techniques across a number of performance metrics [9].

Raj et al. (2024) suggest a better attribute-based encryption method for data stored in lightweight distributed clouds. Storage support is offered by many cloud computing providers in a single heterogeneous architecture. The suggested approach uses a multi-cloud architecture to provide file sharing, safe data storage, and secrecy. Quantitative results indicate that the suggested method is feasible when compared to conventional approaches [10].

The ability of ML to handle enormous volumes of data has made it a prominent solution to the bot identification challenge. However, evaluating and comparing the effectiveness of suggested approaches is challenging due to the variation among research in terms of issue statements, suggested procedures, datasets, and assessment measures.

Dimitriadis et al. (2024) propose CALEB, a powerful end-to-end proactive framework that simulates bot development by building realistic synthetic instances of various bot kinds. It is built on the Conditional Generative Adversarial Network (CGAN) and its enhanced version, Auxiliary Classifier GAN (AC-GAN). These simulated evolved bots improve the identification of new generations of bots before they even manifest by supplementing current bot datasets [11]. Shukla et al. (2024) provide a method for detecting bots that makes use of GANs. It describes how to use several discriminators to train against a single generator, detaching the discriminator to identify social media bots and using the generator for data augmentation, thus overcoming the mode collapse problem. The method exceeds the most advanced approaches in this sector in terms of classification accuracy [12].

Randhawa et al. (2024) suggest a new GAN model that uses Deep Reinforcement Learning (DRL) to investigate semantically aware samples while also making detection more difficult. The discriminator of the GAN, which serves as a botnet detector, is attacked by a DRL agent. In order to assist the GAN generator to converge sooner than it would in the absence of DRL, the agent trains the discriminator on the designed perturbations during the GAN training [13].

Zeng et al. create a behavior representational learning model that concurrently extracts global and local characteristics of behavioral data using a Transformer and a CNN encoder–decoder. Additionally, a network representation learning approach is put forth that uses random walks oriented both within and outside of communities to extract community connections and structural elements from the relationship graph. Lastly, fused representations for bot identification are produced by combining the behavioral representation and relational representation learning models. The suggested approach has several benefits over the most advanced detection techniques in this area, as shown by the experimental results of four publicly accessible social network datasets [14].

Table 1. Recent studies for multi-cloud security and bot detection

Author & Year	Method	Strengths	Limitations
Sharma et al., 2024	Multi-level encryption during network transmission	Strong data protection; enhances complexity through layered encryption	Focused on single cloud environments; lacks integration of multi-cloud data classification
Yang et al., 2023	Attribute-Based Encryption (ABE)	Effective access control for cloud storage	Limited multi-cloud support; no categorization of security levels on the cloud provider side
Kumar et al., 2024	Identity-based system, KCDH, SNP-based data centers	Reduces vendor lock-in; supports dynamic data with verified tags and integrity audits	Implementation complexity: performance may vary with different datasets
Raj et al., 2024	Enhanced ABE in multi-cloud lightweight architecture	Secure file sharing and data secrecy across distributed environments	Less tested under real-time or high-throughput conditions
Nguyen, 2024	Supervised ML approaches (DL & shallow models) for social bot detection	Broad coverage of supervised models; tweet-based bot classification	Lack of standardization in datasets and evaluation metrics; not tailored to recommendation systems
Dimitriadis et al., 2024	CALEB framework using CGAN and AC-GAN	Proactively detects evolving bots; improves dataset robustness with synthetic instances.	Computationally expensive; needs regular updates to stay relevant with evolving threats.
Shukla et al., 2024	Multi-discriminator GAN with data augmentation	Solves mode collapse; high accuracy in social bot detection	Limited scalability; mainly tested on social media data
Randhawa et al., 2024	GAN combined with Deep Reinforcement Learning (DRL)	Boosts generator performance; explores adversarial behavior for stronger detection	Training complexity; risk of adversarial overfitting
Zeng et al., 2024	Behavioral and relational representation using Transformers and CNN	Strong feature fusion; captures global/local and structural patterns	Requires large, diverse datasets; computationally intensive for real-time implementation

While there are a multitude of forward-looking advancements in bot detection and encryption, the existing approaches and solutions provide a disconnected multi-cloud environment, and/or limit attack prevention to bot activity. In this regard, no framework that encompasses broad, scalable multi-cloud encryption and cloud shielding with adaptive AI-based bot detection has yet been realized, and especially not one that delivers such options for real-time recommendation systems. As a result, researchers and developers have had to choose between encryption and bot detection as part of larger defensive approaches to cybersecurity or make tradeoffs concerning performance, interoperability, and effectiveness of their defensive controls.

3. Materials and Methods

Strong security measures are necessary for recommendation systems, which frequently handle sensitive user data, particularly in multi-cloud settings. By creating phoney traffic, manipulating ratings, or scraping data, bots can influence recommendation algorithms.

To guarantee the integrity of the suggestions, effective bot identification is essential.

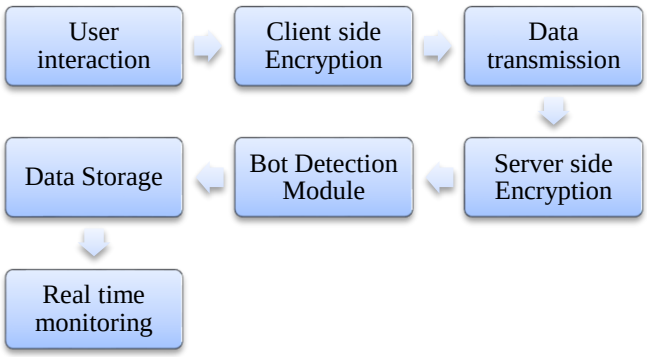

Fig. 1 Workflow for the multi-level encryption and bot detection

3.1. Multi-Level Encryption

A method for applying varying degrees of encryption to data according to its sensitivity is called multi-level encryption. In a variety of scenarios, it aids in maintaining data confidentiality.

The multi-level encryption is shown in Figure 2. The broad-level steps involved in the multi-level encryption are as follows.

3.1.1. Input

- Sensitive information, such as user preferences, past exchanges, and suggestions
- Encryption keys and rules for varying security levels

3.1.2. Data Classification

Sort the data into categories such as general, moderately sensitive, and very sensitive. Then, determine the encryption level according to the sensitivity of the data

- Level 1: Low-sensitivity data (public metadata, for example) is encrypted using a lightweight technique.
- Level 2: For data that is fairly sensitive, strong encryption (such as AES-256) is used
- Level 3: For extremely sensitive data, an advanced encryption algorithm is used.

3.1.3. Key Generation

Key Management System (KMS) to generate encryption keys and to improve security, making sure different keys are used for every cloud storage provider.

Data Partitioning: Divide the data into sections according to storage needs and sensitivity. Divide data chunks across many CSPs to provide security and redundancy.

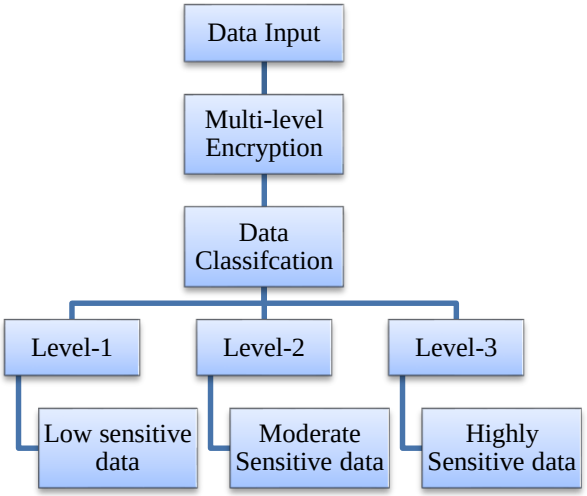

Fig. 2 Multi-level encryption

3.1.4. Encryption Procedure

Use the designated encryption level and matching key to encrypt every data segment. Encrypt very sensitive data using many levels:

- First layer: Use a symmetric key to encrypt.
- Second layer: For safe sharing, re-encrypt using an asymmetric key.
- Third layer: If computation on encrypted data is required, use a homomorphic encryption technique.

3.1.5. Storage and Backup

Keep secure information segments in the cloud storage providers that have been assigned to you. To avoid data loss, make sure duplicates are encrypted and kept in several places.

3.1.6. Decryption Process

The process of decryption (during the creation of recommendations):

- Use token-based validation or Multi-Factor Authentication (MFA) to authenticate the user or the requesting system.
- Retrieve the relevant keys and encrypted data segments from the key management system.
- Decrypt information gradually:
- Using the security hierarchy as a guide, decode at each level.
- At every level of decryption, make sure that access policies are confirmed

3.1.7. Output

Securely stored encrypted data on several cloud platforms; only approved individuals or systems may decrypt it. The encryption algorithms used to encrypt low, moderate, and high-sensitivity data are GIFT, AES256, and GFHE, and they are described as follows.

Galois Field and Integrated Field-Effect Transistor(GIFT)

A lightweight cryptographic method called GIFT was created especially for situations with constraints, such as embedded systems, RFID systems, and Internet of Things (IoT) devices. While retaining a high level of security, lightweight cryptography is tailored for gadgets with constrained resources, such as low computational capacity, memory, and power consumption.

The structure of the GIFT cipher is derived from Galois Field arithmetic, Substitution Permutation Networks (SPNs), and basic XOR operations. Substitution-Permutation Network (SPN): GIFT employs an SPN structure, a popular symmetric cypher architecture that uses several rounds of permutation and substitution to make the cypher impervious to cryptanalysis. To increase the complexity of any potential assaults, the data is spread out throughout each round by being permuted (rearranged) and then substituted (using a substitution table).

Rounds: GIFT typically operates in rounds, performing XOR operations, bitwise rotations, and substitution processes in each round. These procedures increase the dispersion and complexity of the data, making the encrypted message more difficult to anticipate or decode. The number of rounds depends on the block and key sizes.

Scheduling of Keys: The original key is extended into many round keys that are utilized for encryption through GIFT's key expansion technology. The size of the round keys and the number of rounds are determined by the key size (e.g., 80 or 128 bits).

Galois Field Operations: Galois Field arithmetic, a type of finite field arithmetic that is frequently utilized in cryptographic methods, is incorporated into GIFT. In hardware, Galois fields are very helpful for carrying out effective mathematical operations [15].

Advanced Encryption Standard (AES-256)

AES is a symmetric block cipher used to safeguard confidential information. AES-256 encryption encrypts and decrypts a block of data using a 256-bit key length. Each of the 14 rounds of 256-bit keys includes processing stages that convert plaintext into ciphertext, such as mixing, transposition, and substitution. The foundation of AES-256 encryption is a substitution-permutation network, or SP network. Instead of using a Feistel cipher structure, which employs the same fundamental technique for both encryption and decryption, the encryption operates on an SP network structure. In the initial step, the algorithm picks up a single key. Later on, this was extended to include many keys utilized in every round. Rather than using bit data, the AES encryption technique uses byte data. This indicates that throughout the encryption process, the 128-bit block size is treated as 16 bytes. The length of the key used to encode data determines how many encryption rounds must be performed. There are 14 rounds in the 256-bit key size.

Every data unit in encryption is swapped out with a different unit based on the security key that is being utilized. AES is an SP network that creates new keys, known as round keys, by using the original key in a key expansion process. The spherical keys are produced after several iterations of alteration. The encryption becomes more difficult to crack with each round. There are 14 of these rounds in the AES-256 encryption [16].

Gentry's Fully Homomorphic Encryption(GFHE)

A cryptographic technique called Gentry's Fully Homomorphic Encryption (FHE) enables calculations to be done on encrypted material without first decrypting it. This implies that when addition, multiplication, and other more intricate operations are carried out, data can stay encrypted, and the outputs will still be encrypted. One significant development in privacy-preserving technology is the capacity to do calculations on encrypted data. This capability might open the door for secure cloud computing, privacy-enhancing innovations, and other use cases where secrecy is essential [17]. The steps in GFHE are as follows.

Step 1: Key Generation in Gentry's FHE Process

Two keys are generated by the cryptosystem: a public key to encrypt information and a private key. Additionally, keys are utilized for bootstrapping and other homomorphic procedures.

Step 2: Encryption

The public key encrypts data using the following formula: c = Encrypt (m, pk), where cc is the ciphertext, m is the plaintext message, and pk is the public key.

To assure semantic security, a little quantity of noise is added to the ciphertext during encryption, making it computationally impossible to deduce m from c.

Step 3: Homomorphic Operations

The computations are performed directly on the encrypted data

(i) Ciphertext Addition: Add two ciphertexts C1 and C2.

$$C_{sum} = C_1 + C_2 \, mod \, q \quad (1)$$

(ii) Ciphertext multiplication: Multiply two ciphertexts C1 and C2

$$C_{Product} = C_1 . C_2 \, mod \, q \quad (2)$$

Step 4: Bootstrapping

In order to minimize noise and facilitate additional calculations, refresh the ciphertext.

Re-encrypt Ciphertex by applying the encryption method once again to the noisy ciphertext. There is less noise in the updated ciphertext. To decrypt and re-encrypt (conceptually), an encrypted copy of the secret key is used to

carry out a homomorphic decryption of the ciphertext. A ciphertext with noticeably less noise is the result of the updated ciphertext that is prepared for additional calculations.

Step 5: Decryption:

To get the plaintext result, decryption is done on the final ciphertext. Next, eliminate noise from the ciphertext using the private key sk. Lastly, to retrieve the original plaintext, use modular arithmetic:

$$m = c - e \; mod \; q \quad (3)$$

Where e is the noise term, and the plain text is denoted by m.

3.2. Machine Learning Methods

The machine learning models used in this study are RF, XGBoost and LSTM for detecting the bots in the recommendation systems. The general workflow of ML-based Bot detection is shown in Figure 3. The algorithm steps in the ML-based bot detection are as follows.

- Input: The inputs are the information about user engagements (clicks, session length, patterns of activity, etc._, data about network traffic, such as IP addresses, device characteristics, and geolocation and the historical information (such as known bot habits, historical activity patterns, etc).
- Data Preprocessing: The Data Collection and preprocessing involve gathering the 396network traffic, clickstream information, and user interaction records from the recommendation system to get rid of noise or inconsistencies, organize and normalize the data.

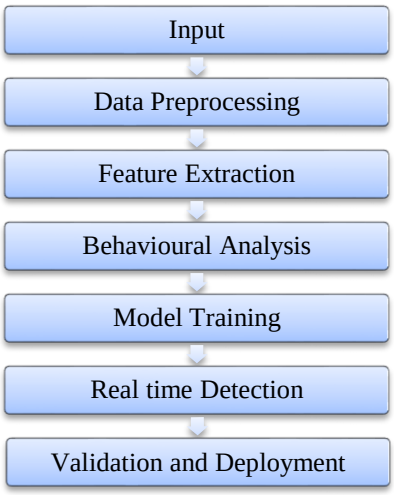

Fig. 3 Workflow of bot detection

- Feature Extraction: This process gathers essential characteristics for bot identification, like: request frequency, mouse movements and scrolling patterns, interaction speed, response times, user agent, browser attributes, geolocation and IP address consistency.
- Behavioral Analysis: It examines user habits in relation to established profiles of human and bot behavior and identifies any discrepancies or deviations from the typical patterns of human interaction.
- Model Training: Apply a supervised or unsupervised ML model that has been trained on labeled datasets of human and Bot interactions, such as Random Forest, Gradient Boosting, or LSTM networks.
- Real-Time Detection: Monitor active user sessions, update the detection system with current data, and use the trained model or rule-based decision thresholds to categorize users as human or Bot.
- Validation and Adaptation: To reduce false positives and false negatives, check the bot classification findings regularly and use feedback loops to modify the detection method to address recently discovered bot activities.
- Output: Bots were identified and marked for blocking or prevention; legitimate users were permitted to continue using the system without interruption.

3.2.1. Random Forest

An RF model is useful for identifying bots in recommendation systems as these systems frequently display unique patterns in user behavior or activity. A random forest is a supervised ML technique that generates a single final result by combining the computations of several decision trees. It is well-liked since it is straightforward but efficient. As an ensemble approach, RF combines a number of base-level models to provide better outcomes. The steps in RF include the following.

- Divide the dataset into smaller groups: An ensemble of decision trees is called a random forest. We must separate our dataset into subgroups in order to generate several decision trees. There are two primary methods for selecting features at random (random feature subspaces) and taking a bootstrap sample, which is a sample of the selected features with replacements.
- Train decision trees: The dataset is divided into subgroups, and decision trees are then trained on these subsets.
- Combine the outcomes: Every tree has a single outcome that is dependent on the original data. Then the output is merged into a single result to eliminate the reliance on the original data and get a more precise estimate. There are several ways to aggregate the findings. For instance, voting by effectiveness is frequently used in classification, yet averaging models are utilized in regression.
- Verify the model: The hold-out validation process is carried out following the completion of the training process using the training data and the execution of the tests using the test dataset. This entails using the same

hyperparameters to train a new model. These specifically include the split function, the number of trees, and the training and pruning methods.

Finding the best training process in terms of metrics, such as accuracy, overfitting resistance, memory, and other generic characteristics, is the aim of creating a general model without pretrained parameters [18].

3.2.2. XGBoost Algorithm

XGBoost is an effective method for detecting bots in recommendation systems since it can handle unbalanced datasets and understand intricate patterns.

An ML approach called XGBoost classifier is used with tabular and structured data. A gradient boosted decision tree implementation made for speed and efficiency is called XGBoost. This indicates that it is a complex ML algorithm. Large, complex datasets are compatible with XGBoost. It is a method for ensemble modeling.

Extreme Gradient Boosting, or XGBoost, is a scalable distributed Gradient-Boosted Decision Tree (GBDT) ML toolkit. It is the top machine learning package for tasks including regression, classification, and ranking and offers parallel tree boosting.

In supervised ML, a model is trained using algorithms to identify patterns in a dataset that contains features and labels. The learned model is then used to predict the labels on features in a new dataset [19].

3.2.3. Long Short-Term Memory (LSTM)

Recommendation systems can benefit greatly from the use of LSTM networks for Bot identification, particularly where temporal behavior or action sequence is a crucial factor in differentiating Bots from people, because they are better at identifying patterns in sequential and time-dependent data than tree-based models like XGBoost. LSTMs are perfect for identifying behavioral variations in clickstreams or session logs.

LSTM networks are a kind of recurrent network designed to improve the vanishing gradient issue with conventional RNNs, which makes them less effective at storing long-term memories. LSTM networks are now the most widely used models for Natural Language Processing (NLP) because of their exceptional memory capacity.

The forget, input, and output gates are the three gates that make up the LSTM network. This resolves RNNs' primary drawback, which is their limited memory. The information that enters and exits the memory cells or refreshes the network memory is determined by these gates [20].

4. Results and Discussion

The experiments were performed using Python. The dataset used in this study is a Twitter bot detection dataset, which is a publicly available dataset and can be freely downloaded from Kaggle. Twitter accounts that have been categorized as either human or Bot are beneficial for creating Bot detection ML models. Some of the features in the sample Bot detection dataset are listed in Table 2.

Table 2. Features in the twitter bot detection dataset

Feature Name	Description
Age of the account	How many days have passed since the account creation
Verified	Boolean value to verify whether the account has been validated
Profile image	Profile image is uploaded or not.
Name similarity	The degree to which the username and account are similar
Followers count	Number of followers
Friends count	Number of friends
Growth rate	Followers rate over time.
Tweets count	Total number of tweets
Tweets per day	Average number of tweets per day
URL ratio	Percentage of liked tweets

Table 3. Performance analysis

Method /Measures	Accuracy	Precision	Recall	F1-Score
RF	91.5	89	86	87
XGBoost	95.3	91	92	91
LSTM	92	.89	93	91

Table 3 shows the performance indicator values for methods like RF, XGBoost, and LSTM. The random Forests recall is somewhat lower than comparable models, but accuracy is good at 91.5%. It has a somewhat weaker recall and could miss certain Bots, but its precision is balanced.

In XGBoost, all indicators, including accuracy (95.3%), precision (0.91), recall (0.920), and F1-score (0.91), were high for the best overall performance, and it excels at managing engineering features and class imbalance. LSTM: has a high recall (0.93), indicating that it is quite good at spotting Bots. Although the F1-score is competitive (0.91), overfitting or extended training sessions may prevent it from reaching XGBoost's accuracy.

Figure 4 shows that the XGBoost outperforms the other two methods in classifying the Bots in the recommendation systems. Bot identification in recommendation ensures that automated Bots or malevolent individuals do not tamper with ratings, interactions, or suggestions. The encryption algorithms used in this study to classify the different types of data are shown in Table 4.

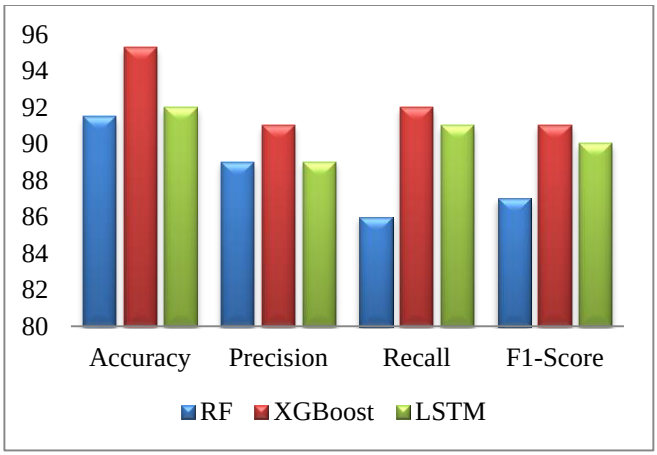

Fig. 4 Metrics vs ML methods

Table 4. Encryption algorithms used in this study

Feature	GIFT	AES-256	GPHE
Sensitivity of data	Lowly sensitive data	Moderate sensitive data	Highly sensitive data
Type	Lightweight block cipher	Symmetric block cipher	Fully Homomorphic Encryption (FHE)
Key size	128 bits	256 bits	Longer than 256
Algorithm type	Small Block size cipher	Large block size cipher	FHE computation
Operations	Substitution and permutation	Substitution and permutation	Arithmetic operations on encrypted data
Speed	Very fast	Fast	Extremely slow
Cryptanalysis Resistance	Strong	Extremely resistant to cryptanalysis	Very strong

Table 1 provides a comparison of encryption algorithms for encrypting the low, moderate and high sensitive data. GIFT provides a basic defense against t frequent attacks, but it is insufficient for more sensitive information or complex threats. AES 256 provides a balance between security and computing performance, offering better protection than low-level encryption. It offers maximum protection, safeguarding extremely important and sensitive data.

4.1. Discussions

The findings of this study exceed many of the approaches we identified in the existing literature because of several key features that enhanced Bot score accuracy and data security in the multi-cloud recommendation system. The most notable factor for overall accuracy comes from the introduction of XGBoost, which had the greatest accuracy (95.3%), precision (91%), recall (92%), and F1 score (91%) of all models tested. An advantage of utilizing XGBoost is its use of a gradient boosting framework. It is more suited for structured data and is versed in inferring feature interaction and class imbalance, which are persistent issues for Bot data (discussed in Section 6). XGBoost also made an impact because it can understand large feature spaces and facilitates overfitting control through regularization, which provides model superiority over traditional classifiers, such as Random Forest, and deep learning models such as LSTM**.**

Although LSTM is an excellent method for learning temporal sequences and achieving a high recall value of 0.93, it underperformed slightly in accuracy, potentially due to its complexity and tendency to overfit when trained on small datasets. Regardless of lower accuracy than a few other models, recall was extremely high and desirable in minimizing the chances of false negatives (not finding), so that not many Bots are being missed.

This is important because this is what the security of recommendation systems cares about; false negatives (i.e., Bot accounts not detected) often lead to serious ongoing repercussions for user trust and system integrity. Moreover, arguably, the addition of a rich feature dataset that included profile metadata, behavioral attributes (i.e., frequency of tweeting, URL ratio), and account verification rates allowed for greater discrimination between human and Bot accounts in training.

On the encryption side, the new multi-level encryption strategy, based on sensitivity levels, was able to provide strong protection with a limited tradeoff in performance. GIFT was employed for low-sensitivity data to ensure speed, AES-256 provided acceptable performance for moderately sensitive data, and GPHE provided the maximum level of security for highly sensitive data, even at its highest computational cost. As an integrated hybrid framework for optimized ML detection and a tiered encryption system, this will provide a comprehensive, secure, and scalable solution that outperforms and surpasses traditional methods on both performance and protection.

5. Conclusion

This framework protects recommendation systems' availability, secrecy, and integrity by fusing strong multi-level encryption with real-time Bot detection. The algorithmic processes offer a complete solution for contemporary recommendation platforms by emphasizing scalability, flexibility, and the safe management of private information across multi-cloud settings. A safe, reliable, and user-focused solution is guaranteed when recommendation systems are supported by several cloud storage systems,

including bot detection and multi-level encryption. This architecture opens the door for contemporary recommendation systems that put user happiness and data security first, despite some obstacles. Further developments in this field will be fueled by ongoing research and developments in distributed storage, AI-driven Bot identification, and encryption. It is anticipated that future studies will concentrate on creating flexible, scalable, and privacy-preserving solutions that combine strong encryption methods with sophisticated Bot identification to improve the security and effectiveness of cloud-based recommendation systems.

References

[1] Chinnadurai Manthiramoorthy, K. Mohamed Sayeed Khan, and A. Noorul Ameen, "Comparing Several Encrypted Cloud Storage Platforms," *International Journal of Mathematics, Statistics, and Computer Science*, vol. 2, pp. 44-62, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Dinesh Reddy Chirra, "Secure Data Sharing in Multi-Cloud Environments: A Cryptographic Framework for Healthcare Systems," *Revista de Inteligencia Artificial en Medicina*, vol. 15, no. 1, pp. 821-843, 2024. [Google Scholar] [Publisher Link]

[3] Akif Quddus Khan et al., "Cloud Storage Cost: A Taxonomy and Survey," *World Wide Web*, vol. 27, pp. 1-54, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Mais Irreem Kamal, and Laheeb Ibrahim, "The Multilevel Encryption Model: A Review," *Al-Rafidain Journal of Computer Sciences and Mathematics*, vol. 18, no. 1, pp. 40-49, 2024. [Google Scholar]

[5] Zhaowei Hu, Kaiyi Hu, and Milu Md Khaled Hasan, "A Bidirectional Reversible and Multilevel Location Privacy Protection Method Based on Attribute Encryption," *Plos One*, vol. 19, no. 9, pp. 1-26, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Zijian Cai et al., "LMBot: Distilling Graph Knowledge into Language Model for Graph-less Deployment in Twitter Bot Detection," *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, Merida Mexico, pp. 57-66, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] Isha Sharma, and Monika Saxena, "A Review of Lightweight Cryptography Algorithm for Healthcare Using Multi-Level Encryption," *SSRN*, pp. 1-7, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Guangcan Yang et al., "An Efficient Attribute-Based Encryption Scheme with Data Security Classification in the Multi-Cloud Environment," *Electronics*, vol. 12, no. 20, pp. 1-19, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] P. Hari Kumar, and G.S. AnandhaMala, "HMAC-R: Hash-Based Message Authentication Code and Rijndael-Based Multilevel Security Model for Data Storage in Cloud Environment," *The Journal of Supercomputing*, vol. 79, pp. 3181-209, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Shani Raj, and B. Arunkumar, "Enhanced Encryption for Light Weight Data in a Multi-Cloud System," *Distributed and Parallel Databases*, vol. 41, pp. 65-74, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Ilias Dimitriadis, George Dialektakis, and Athena Vakali, "CALEB: A Conditional Adversarial Learning Framework to Enhance Bot Detection," *Data & Knowledge Engineering*, vol. 149, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Anant Shukla, Martin Jureček, and Mark Stamp, "Social Media Bot Detection Using Dropout-GAN," *Journal of Computer Virology and Hacking Techniques*, vol. 20, pp. 669-680, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Rizwan Hamid Randhawa et al., "Deep Reinforcement Learning Based Evasion Generative Adversarial Network for Botnet Detection," *Future Generation Computer Systems*, vol. 150, pp. 294-302, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Fanrui Zeng, Yingjie Sun, and Yizhou Li, "MRLBot: Multi-Dimensional Representation Learning for Social Media Bot Detection," *Electronics*, vol. 12, no. 10, pp. 1-24, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Furqan Zahoor et al., "Design implementations of Ternary Logic Systems: A Critical Review," *Results in Engineering*, vol. 23, pp. 1-23, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Shubham Verma et al., *Soldier Strap: An Iot-Based Safety and Security Band for Soldiers*, 1st ed., IoT-Enabled Healthcare Systems, Apple Academic Press, pp. 1-20, 2024. [Google Scholar] [Publisher Link]

[17] Ramalingam Praveen, and Parameswaran Pabitha, "Improved Gentry–Halevi's Fully Homomorphic Encryption-Based Lightweight Privacy Preserving Scheme for Securing Medical Internet of Things," *Transactions on Emerging Telecommunications Technologies*, vol. 34, no. 4, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Peace Azugo, Hein Venter, and Mike Wa Nkongolo, "Ransomware Detection and Classification Using Random Forest: A Case Study with the UGRansome2024 Dataset," *Arxiv Preprint*, pp. 1-12, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Malak Aljabri et al., "Machine Learning-Based Social Media Bot Detection: A Comprehensive Literature Review," *Social Network Analysis and Mining*, vol. 13, pp. 1-40, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Vajratiya Vajrobol et al., "Adversarial Learning for Mirai Botnet Detection Based on Long Short-Term Memory and XGBoost," *International Journal of Cognitive Computing in Engineering*, vol. 5, pp. 153-160, 2024. [CrossRef] [Google Scholar] [Publisher Link]