Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Adaptive Machine Learning-Based IDS with Threat-Specific Encryption for Secure IoT Communication

Roopum Dubey[1*], Suhel Mustajab[2]
[1,2,]Department of Computer Science, Aligarh Muslim University, India, 202002
[*]Corresponding author: Roopum Dubey (email: roopumdubey172@gmail.com)

ARTICLE INFO

Received: 18 Dec 2024

Revised: 10 Feb 2025

Accepted: 28 Feb 2025

ABSTRACT

The exponential growth of the Internet of Things (IoT) has introduced new dimensions of connectivity. Still, it also brings critical security challenges due to IoT devices' heterogeneity and resource constraints. Traditional Intrusion Detection Systems (IDS) often fail to meet IoT environments' real-time and adaptive security requirements, particularly against sophisticated and zero-day attacks. This paper proposes a hybrid IDS framework that integrates machine learning-based traffic classification with risk-adaptive encryption mechanisms to address these limitations. The system utilizes Random Forest classifiers to categorize network traffic into benign, low-risk, and high-risk threats. A dual-mode encryption strategy is applied based on the threat level: high-risk data is secured using a hybrid RSA and Modified ChaCha20 encryption algorithm. In contrast, low-risk data uses the lightweight Modified ChaCha20 alone. The encryption model introduces a non-linear transformation and custom permutation layer to enhance diffusion and security. Experimental evaluations demonstrate that the proposed system performs better in encryption time, throughput, entropy, and energy efficiency than traditional AES and RSA schemes. Moreover, the keystream randomness was validated through the NIST statistical test suite, confirming its robustness against cryptanalytic attacks. This hybrid approach ensures scalable, intelligent, and secure communication for real-time IoT operations.

Keywords: Intrusion Detection System(IDS), Internet of Things(IoT), Machine Learning (ML), Random Forest, RSA, ChaCha20

1. INTRODUCTION

The Internet of Things (IoT) is reconfiguring the ultramodern digital topography by interconnecting billions of biases in colourful domains, such as healthcare, artificial robotization, smart homes, transportation, and husbandry. IoT facilitates real-time data harvesting, wise decision-making, and robotization, revolutionizing how people and enterprises engage with technology. Nevertheless, as IoT relinquishment increases, so does its susceptibility to cyber pitfalls and security vulnerabilities. In contrast to conventional computing environments, IoT networks consist of various, resource-limited biases that exchange information over the internet, often working in open or public networks. This IOT device property subjects them to a broad spectrum of cyberattacks, such as malware infections, denial-of-service (DoS) attacks, man-in-the-middle (MITM) attacks, data breaches, and device kidnapping [1]-[2]. Additionally, the ad-hoc nature of IoT ecosystems, where multiple manufacturers create bias with different security standards, also makes cybersecurity sweats more challenging. The decentralized nature of IoT makes conventional security methods ineffective since they cannot accommodate the enormous number of connected devices. Although firewalls and encryption offer a fundamental level of protection, they cannot identify advanced, zero-day attacks or emerging intrusion techniques. This highlights the imperative necessity of Intrusion Detection Systems (IDS) in IoT settings. The Internet of Effects (IoT) is rapidly transubstantiating hard work by attaching billions of innovative biases, facilitating robotization, real-time monitoring, and seamless communication across various fields. These networked biases are deployed in smart homes, healthcare, artificial robotization, innovative cities, transportation, and critical infrastructure, providing vibrant functionalities that enhance functional efficiency. Nevertheless, this vast interconnectivity creates sharp security exposures, rendering IoT bias high targets for cyber traps. IoT environments differ from the conventional IT networks in that IoT environments map to miscellaneous bias with diverse tackle infrastructures, operating systems, and communication protocols. Many IoT biases are characterized by limited calculating power, memory, and battery life, which poses difficulties for deploying strong security mechanisms. These limitations render IoT networks vulnerable to cyberattacks such as DoS attacks, billabong attacks, renewal attacks, Sybil attacks, malware injections, and unauthorized access.

Similarly, IoT bias constantly causes, handles, and transfers sensitive information, rendering data vacancy, fidelity, and secrecy prime enterprises. Conventional security protocols, such as firewalls and hand-grounded intrusion discovery systems (IDS), cannot provide sufficient security in rapidly changing IoT environments where new traps emerge sprightly. This mandates the creation of smart security outcomes that incorporate cutting-edge technologies like Machine Learning (ML) and cryptographic methods to improve intrusion detection and data security. An IDS is a security outcome that aims to blanket, analyze, and define malicious conditioning in a system or network. It is crucial in connecting cyber pitfalls prior to them siring substantial harm. Since IoT is an ever-changing arena, a powerful IDS should also be adaptive, scalable, and capable of recognizing real-time attacks to combat developing security traps[3]. IDS for IoT may be categorized, as illustrated by the figure, on the basis of their discovery, confirmation, and placement schemes. All three schemes have some advantages and drawbacks, affecting an IDS's performance in an IoT environment as a whole. Conventional IDS implementations often compute on hand-grounded discovery, in which case predefined attack autographs are needed, or anomaly-grounded discovery, in which case it is a matter of distractions from typical geste. Furthermore, the selection of confirmation approach determines how explicitly an IDS can validate if an identified anomaly is an actual cyber trouble. On the other hand, the placement approach affects the efficacy of live intrusion monitoring and response. A centralized IDS would provide additional control and improved correlation of data, while a distributed IDS provides additional scalability and fault tolerance. Still, IDS implementation differs significantly by detection mechanisms (signature-based, anomaly-based, or hybrid), verification methods (rule-based, statistical, or machine learning), and deployment types (centralized, distributed, or hybrid) [4].

Signature-based IDS models identify attacks through predefined signatures but do not catch unknown threats. Anomaly-based systems identify deviance from normal behavior through machine learning, thus identifying zero-day attacks but in many cases at the cost of high false favorable rates [5]. Hybrid techniques use both techniques to enhance the detection rate with the reduction in computational inefficiencies. However, current IDS solutions continue to face various challenges, including high false positives, limited checking of encrypted traffic, limited scalability in large-scale IoT networks, and lack of automated, threat-driven response capabilities [6]-[7]. Our work draws on these IDS techniques by proposing a mongrel paradigm that combines ML-predicted business brackets with threat-configured encryption. With a hybrid of KNN, CNN-based anomaly detection, and ChaCha20 encryption. Our suggested IDS fortifies trouble finding delicacy and data security, icing a firm defense medium to IoT networks. The following section describes this emerging strategy in-depth. IDS is critical in making IoT networks safe by relating and soothing cyber pitfalls. However, IDS mechanisms vary; they are developed and rooted on various discovery methods, verification strategies, and placement tactics. Knowledge of these abecedarian approaches is crucial to the formulation of a practical IDS framework for IoT environments. The discovery approach specifies how an IDS detects implicit pitfalls, either through pre-defined attack autographs, behavioral anomaly discovery, or a blend of both. The confirmation approach addresses how an IDS confirms and categorizes detected pitfalls, employing styles from theoretical and empirical confirmation to anomaly-grounded learning methods. Incipiently, the deployment strategy decides whether and how IDS factors are deployed in an IOT network – centrally, distributed, or utilizing a mongrel strategy to balance both. Figure 1 below classifies these IDS strategies, presenting a high-position overview of how intrusion discovery systems are organized in IoT security.

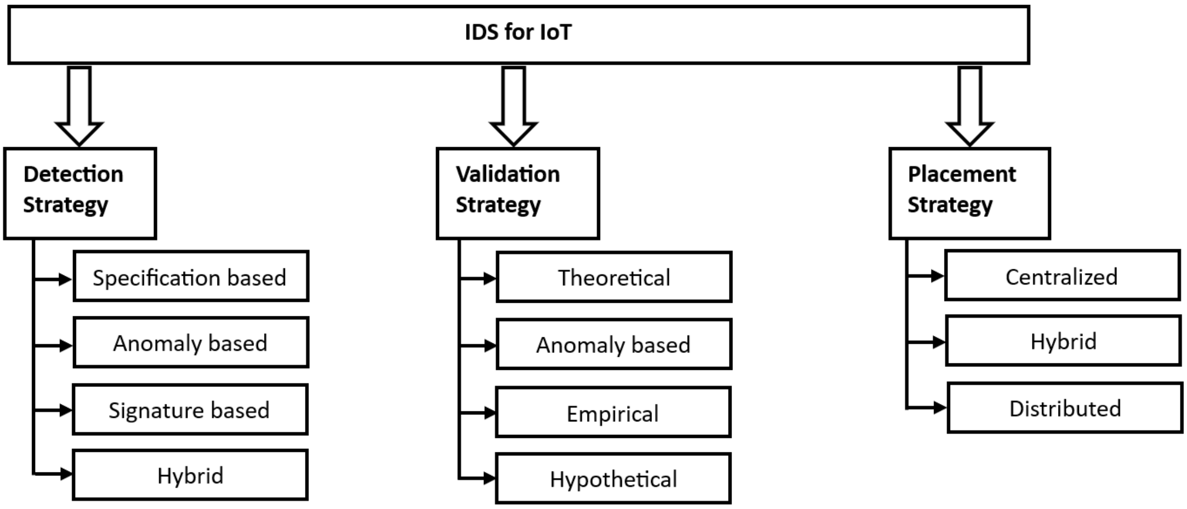


Figure 1: IDS Classification Strategies for IoT

IoT networks are confronted by multitudinous security issues because of their resource-constrained and decentralized nature. Security ventures highlight the necessity for an advanced mongrel Intrusion.

Data Confidentiality is used to denote unauthorized access, and data revision can be used to jeopardize sensitive IoT information, causing sequestration breaches in such critical operations as healthcare and finance. Data Integrity is used to denote icing the responsibility of transmitted data and plays a critical role in IoT networks, where modified or falsified information can cause severe functional dislocations. Data Availability is about IoT networks which need to keep working and accessible even in the face of hostile efforts to stop services through DoS or Distributed Denial-of-Service (DDoS) attacks. Authentication and Authorization are about poor authentication processes that cannot permit bushwhackers to attain unauthorized access to bias, support man-in-the-middle attacks, renewal attacks, or identity spoofing. Network Scalability and Performance means that as exponentially the number of IoT biases increase, it becomes a big challenge to secure real-time communication and recycle large datasets in an effective manner.

IoT networks are faced with multitudinous security challenges due to their decentralized and resource-constrained nature. Security endeavors emphasize the need for a sophisticated mongrel Intrusion.

Data Confidentiality is utilized to signify illegal access, while data revision can be utilized to compromise sensitive IoT data to induce sequestration violations in such key operations as healthcare and finance. Data Integrity is utilized to signify icing the duty of transmitted data and exercises a vital role in IoT networks, where altered or tampered information can induce serious functional dislocations. Data Availability refers to IoT networks that must continue to function and be available even in the presence of hostile attempts to disrupt services via DoS or Distributed Denial-of-Service (DDoS) attacks. Authentication and Authorization refer to weak authentication processes that cannot allow bushwhackers to achieve unauthorized access to bias, facilitate man-in-the-middle attacks, renewal attacks, or identity spoofing. Network Scalability and Performance implies that as exponentially the number of IoT biases grows, it is a huge task to secure real-time communication and recycle large datasets in an efficient way.

Even with the improvement, numerous ML-based IDS models do not adapt dynamically with the severity level of the attacks identified. An apparatus that grants equal encryption and processing load for low-risk anomaly as well as high-impact attacks squanders resources in IoT networks that are already resource-poor. Thus, a hybrid IDS architecture has become necessary that classifies and detects threats precisely and adjusts its security reaction according to risk severity. New advancements in adaptive cryptographic techniques, federated learning, and cloud-based intelligence sharing have been promising to construct strong frameworks [11]-[13]. The limitations brought about by the prevailing IDS strategies are delineated below:

- **High False Positive Rates –** A false positive in an IDS is when the system recognizes legitimate network exertion as trouble. Normal geste is thus inappropriately marked as an intrusion, causing unnecessary security warnings.
- **Limited Detection of Encrypted Traffic –** Ultramodern networks, such as IoT environments, are very dependent on encryption protocols such as TLS and SSL for encrypting data in conveyance. Though encryption increases data confidentiality and security, it poses a very big challenge for IDS since translated packets cannot be smoothly audited for malicious content.
- **Scalability Issues –** Scalability can be described as an IDS ability to accommodate increases in amounts of network business, bias, and security events without disparaging performance. As the IoT environments expand, the legacy IDS models experience challenges reusing extensive amounts of real-time business, resulting in performance backups.
- **Lack of Integrated Security Measures –** Maximum traditional IDS models are centered only on detecting intrusion, but they do not embrace visionary security actions to protect breached data or mitigate attacks. This flaw exposes IoT networks to threats because identifying trouble without response mechanisms does not prevent data breaches or damage.
- In order to solve the shortcoming of conventional IDS, we suggest a new Intrusion Discovery System (IDS) that integrates ML-ground brackets seamlessly with cutting-edge cryptographic security mechanisms. This system is created to offer a framed and adaptive security frame that can effectively relate to and placate cyber pitfalls in IoT networks. Real-time network business monitoring from IoT bias constitutes the first subcaste of defense in our IDS. Utilizing ML models like K- Nearest Neighbors (KNN) and Convolutional Neural Networks (CNN), the system examines and classifies network conditioning into two essential orders like low-threat and high-threat attacks. They are characterized as:
- **Low-threat attacks** include unauthorized access attempts and abnormal data transmissions that do not pose an immediate peril but still bear monitoring.
- **High-threat Attacks**: - These involve severe cyber pitfalls similar to denial-of-service (DoS) attacks, malware intrusions, data breaches, and renewal attacks that bear immediate action.

Various attacks require various security contexts; our IDS utilizes a threat-based encryption approach. Translated, the classified trouble information is securely relayed to a pall-based security platform for advanced forensic examination and quick-shooting trouble response. The pall-based method provides a number of benefits. Real-time trouble Intelligence continually updates and improves its trouble detection models, icing visionary cybersecurity defense. Utilizing Automated Security Responses, the platform can robustly adapt security mechanisms and react to pitfalls in real time. Cooperative Cybersecurity Network allows organizations to share important trouble intelligence, bringing about a unison defense initiative against emerging cyber pitfalls. Combining ML with encryption-based security protocols, our suggested IDS facilitates the adaptability of IoT networks, providing an adaptable, intelligent, and scalable solution to intrusion discovery and prevention.

This work introduces a hybrid IDS architecture combining machine learning (ML) under the Random Forest algorithm for effective anomaly detection. Depending on the severity of the detected threats, the system dynamically employs either light Modified ChaCha encryption or combined RSA-ChaCha. Low-risk threats are protected with Modified ChaCha to guarantee efficiency, while high-risk anomalies trigger an RSA-based key exchange preceded by ChaCha encryption to provide strong security. This threat-adjusted cryptographic scheme guarantees optimal performance and security.

The main contributions of this paper are:

- A machine learning-based IDS utilizing Random Forest for accurate threat classification in IoT networks.
- A novel dual-mode encryption mechanism that adapts encryption strength based on threat severity.
- Integrating a non-linear transformation and permutation layer into the ChaCha stream cipher for enhanced security and diffusion.
- Comprehensive performance evaluation using metrics such as encryption time, throughput, avalanche effect, Shannon entropy, energy consumption, and memory usage.

- Validation of keystream randomness using the NIST test suite to confirm the model's robustness against statistical attacks.

The proposed IDS framework addresses existing scalability, adaptability, and cryptographic efficiency limitations, providing a secure, scalable, and intelligent security architecture tailored for IoT environments.

2. RELATED WORK

In [14] et al., the authors introduce a mongrel IDS model that combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) to explain intrusions in IoT environments. The research emphasizes the ability of CNN to reward spatial features from network commerce while LSTM catches successional interdependencies, enhancing the sensitivity of attack discovery. The outcomes of experiments reveal that the model realizes 98.6 discovery sensitivity, surpassing classical IDS methods. however, the research also identifies high computational environments, rendering it less feasible for resource-constrained IoT bias. In [15] et al., the experimenters suggest an anomaly-based IDS utilizing a Support Vector Machine (SVM) and Decision Trees in network business intervals. The research highlights low false positive rates, optimizing the trustworthiness of anomaly detection. nevertheless, the effectiveness of the model heavily relies on the emptiness of labeled datasets, and it is challenged by unfamiliar attack patterns because it is based on supervised learning. In [16] et al., an autoencoder-based IDS is proposed to explain zero-day attacks by learning network gets anomalies. The results show that the model can effectively identify preliminarily unseen pitfalls. nevertheless, high false alarm rates are still a challenge, resulting in implicit security alert fatigue. In [17] et al., a Generative Adversarial Network (GAN)- based IDS is put forward to improve training sets for anomaly detection. The research demonstrates that artificially generated attack scripts improve model resilience, resulting in a tighter bracket of cyber traps. nevertheless, GANs have high computational budgets and large data, which could constrain application in real-time IoT environments. In [18] et al., the researchers present a mongrel IDS that integrates hand-grounded and anomaly-grounded discovery methods. The research discovers that the combination of both methods balances delicacy and speed of discovery, although it raises model complexity and demands high computational power to work efficiently. The experimenters in [19] et al. employ a CNN with point selection to improve intrusion discovery within IoT networks. The research proves that choosing relevant network features minimizes dimensionality and processing time, making the IDS efficiently and quickly. nonetheless, the model has difficulties with initially unseen attack patterns, which makes it less rigid. In [20] et al., an LSTM-based IDS is constructed to model successional attack patterns in IoT environments. The research verifies that LSTM suits time-series attack detection well and thus is most suitable for associating slow-operating cyber pitfalls. yet, long training periods and high computational complexity restrict its use in real-time IoT applications. In [21] et al., allied learning (FL) is used for IDS to facilitate distributed intrusion discovery without involving raw data. The research emphasizes that FL increases data sequestration and security within IoT networks. yet, communication outflow and allied updates present serious challenges to real-time processing. In [22] et al., ensemble learning is employed to improve intrusion bracket using Decision Trees, Random Forest, and SVM. The findings establish that ensemble learning attains sophisticated delicacy and trustability in attack detection. nonetheless, optimization of multiple classifiers enhances computational complexity, rendering it sensitive to implement on featherlight IoT bias. In [23] et al., a Deep Neural Network (DNN)-based IDS is suggested to improve scalability in extensive IoT networks. The research discovers that deep learning greatly enhances discovery delicacy, but hostile attacks can deceive the model, diminishing trustability. In [24] et al., the authors discuss blockchain-based security models for intrusion discovery in IoT. Blockchain in the study finds that it reinforces data trustworthiness but entails high computation outflow restricting its connectivity for real-time IDS findings. In [25] et al., IDS is made secure for network business by homomorphic encryption and also enables processing of translated data. The findings point out that through this process, sequestration is assured in analysis, while encryption and decryption dormancy continues to be main concerns. In [26] et al., the researchers suggest an access control medium incorporated into IDS to limit unauthorized access. The research discovers that grainy authorization controls increase security, but complexity in policy operation grows with network size. In [27] et al., a zero-trust security framework is incorporated into IDS to authenticate device individualities prior to permitting network access. The research emphasizes tougher security enforcement, but the methodology demands continuous authentication, introducing processing outflow. In [28] et al., a mongrel AES-RSA encryption scheme is utilized to protect IoT dispatches against cyber pitfalls. The research reaffirms that such a combination of symmetric and asymmetric encryption provides firm security, yet high computational cost is still an issue for IoT bias with scarce resources. In [29] et al., various cryptography-based IDSs are put forward as evidence of future IoT security. Resistance to mounting attacks is noted by the study, but conditions that require special tackle make relinquishment burdensome. In [30] et al., authors build a homomorphic symmetric encryption scheme to support secure data processing in IDS. Translated analytics improve data confidentiality, but pets for slow processing is an issue according to the study. In [31] et al., cold-blooded security fabrics fueled by AI are considered, with machine learning and cryptography being used to facilitate discovery and response automation. Adaptive trouble mitigation is found to enhance security posture, but extensive datasets must be used to train effectively. In [32] et al., blockchain with AI boosts IDS performance. The findings emphasize that blockchain offers rigid logging while AI facilitates real-time trouble analysis automation. however, scalability concerns in blockchain networks constrain deployment in real life. In [33] et al., the researchers propose a mongrel IDS with featherlight encryption to enhance discovery efficiency and data security. The research finds that the strategy achieves security with performance, and it is the best fit for IoT operations. Table 1 summarizes the related work in table form.

Table 1: Comparative analysis on existing work

Paper Name	Technique Used	Purpose	Strength	Limitation
Hybrid Intrusion Detection System for IoT [14]	Hybrid CNN-LSTM for IDS	Detect intrusions in IoT networks	High detection accuracy (98.6%)	High computational cost
Network Anomaly Detection Using ML [15]	Decision Trees & SVM	Classify network traffic anomalies	Low false positive rate	Needs labeled training data
Unsupervised Learning for Intrusion Detection [16]	Autoencoder-based IDS	Detect novel attacks using anomaly detection	Works well with unknown threats	High false alarm rate
GAN-Based Network Security Model [17]	GAN for intrusion detection	Generate synthetic attack scenarios for training	Improves model robustness	Needs a large dataset for training
Signature and Anomaly-Based Hybrid IDS [18]	Hybrid IDS (Signature + Anomaly)	Combine rule-based and anomaly detection for IDS	Balances detection speed and accuracy	Complexity in hybrid model integration
Feature Selection in IDS [19]	CNN with Feature Selection	Optimize feature extraction for IDS	Reduces dimensionality, improving speed	Performance degrades with unseen attacks
Time-Series Attack Detection in IoT [20]	LSTM-based IDS	Detect sequential attack patterns in IoT	Suitable for time-series attack detection	Requires extensive training time
Federated Learning-Based IDS [21]	Federated Learning for IDS	Distributed IDS model without central data storage	Improves data privacy and security	Communication overhead in federated updates
Ensemble Learning for Network Security [22]	Ensemble Learning (DT, RF, SVM)	Improve classification performance for IDS	High accuracy in attack detection	Complex model tuning is required
Deep Learning for IoT Security [23]	Deep Neural Networks (DNN)	Identify IoT security threats	High scalability across networks	Vulnerable to adversarial attacks
Blockchain-Enabled Security Model [24]	Blockchain-based Security Model	Secure IoT data against tampering	Decentralized trust model	High computational overhead

Paper Name	Technique Used	Purpose	Strength	Limitation
Privacy-Preserving Computation [25]	Homomorphic Encryption	Enable computation on encrypted data	Ensures privacy during processing	High latency in encryption/decryption
Access Control Mechanisms in Cybersecurity [26]	Access Control Mechanism	Restrict unauthorized access	Granular permissions	Complex policy management
Secure Multi-Party Computation for Data Privacy [27]	Secure Multi-Party Computation	Collaborative security without data exposure	Privacy-preserving analytics	High communication overhead
Zero Trust Security Framework [28]	Zero Trust Model	Prevent unauthorized access based on identity	Stronger security framework	Increased complexity in implementation
Hybrid Encryption for IoT Communication [29]	Hybrid AES-RSA Encryption	Secure communication in IoT	Strong encryption with authentication	High computational cost
Quantum-Secure Cryptography [30]	Quantum Cryptography	Future-proof encryption for secure networks	Resistant to quantum attacks	Requires specialized hardware
Encrypted Data Processing Framework [31]	Homomorphic + Symmetric Encryption	Enhance data security while enabling analytics	Supports encrypted processing	Slow processing speeds
AI-Driven Cybersecurity [32]	AI-Driven Hybrid Security Framework	Automate detection and response	Adaptive threat mitigation	Requires large datasets for training
Blockchain and AI for Network Protection [33]	Blockchain + AI-based Security	Enhance integrity and automated threat detection	Transparent and immutable logging	Scalability issues in blockchain networks

3. METHODOLOGY

Intrusion Discovery Systems (IDS) are a central component in icing IoT network security by correlating and soothing cyber pitfalls. Conventional IDS designs often fail to match changing attack patterns, resulting in the abandonment of machine learning-based IDS. Machine learning improves IDS by learning from literal network business data, correlating anomalies, and classifying malicious conditioning with sophisticated delicacy. This strategy enhances discovery rates, decreases false cons, and adapts robustly to new pitfalls. Machine learning models are categorized as supervised, unsupervised, and deep learning models.

Supervised learning techniques like Support Vector Machines (SVM), Decision Trees, and Random Forest compute on labeled data to classify network business in a straightforward manner. Unsupervised learning algorithms like K- Means clustering and DBSCAN assist in the identification of anomalies without having specific markers but are likely to fail in distinguishing benign diversions from actual attacks. Like Convolutional Neural Networks (CNN) and sporadic Neural Networks (RNN), deep learning methods provide enhanced point birth techniques. Nonetheless, they have large computational budgets and are less appropriate for deployment in real-time IDS in IoT networks.

With these considerations in mind, choosing the relevant machine learning algorithm is paramount for the construction of an efficient IDS. Out of colorful bracket models, Random Forest has emerged as a principally reliable method because it is robust, efficient, and can process high-dimensional data.

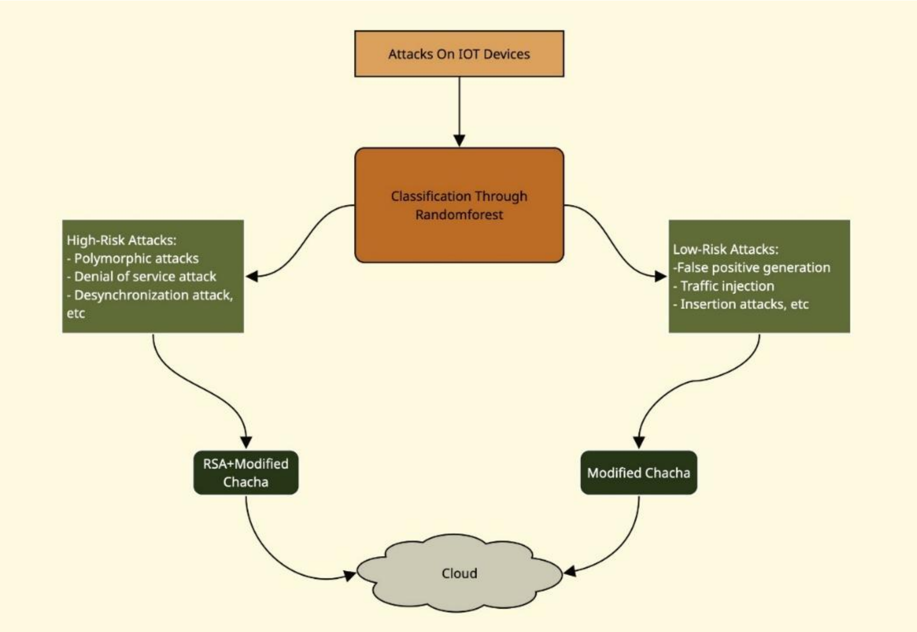


Figure 2: Working of the Proposed Algorithm

3.1.Random Forest

Random Forest algorithm comprises a number of decision trees working individually. Every tree is trained on a sample of the data set with the help of an approach called bootstrap aggregation (bagging). The ultimate bracket decision is obtained with maturity voting of all the trees. Steps to utilize Random Forest in IDS are described below:

1. **Data Preprocessing**: The dataset undergoes point selection, normalization, and running of missing values.
2. **Feature Selection:** Point Selection Important features are uprooted grounded on their donation to distinguishing regular and attack businesses.
3. **Training Phase**: Multiple decision trees are trained on different subsets of the dataset.
4. **Bracket Phase**: When a new network business case is anatomized, each tree votes on the bracket outgrowth.
5. **Final Decision:** The class with the most votes is named in the prognosticated order.

Random Forest is an ensemble machine learning algorithm that builds multiple decision trees and combines their outputs to improve classification accuracy and reduce overfitting. It operates by constructing many decision trees during training and outputting the class that is the **majority vote** of the individual trees.

x be the input feature vector. $h\_1(x), h\_2(x), \ldots, h\_k(x)$ be the predictions from each of the K decision trees.

Then, the final predicted class C_{final} is given by Equation 1.

$$C_{final} = \frac{argarg\ max}{C} \sum_{i=0}^{k} \quad I(T_i\ (x) = C$$
(1)

Where C_{final} is the final predicted class.

- N is the total number of decision trees.
- (T_i (X)=C) represents the classification result from the **i-th** decision tree for input **X**.

- $\sum_{i==1}^{N}$ I (T_i (X)=C) is an indicator function that returns one if tree **T_i** classifies **X** as class **C** and zero otherwise.
- The class **C** with the highest sum across all trees is selected as the final classification.

Each tree in the forest is trained on a random subset of the initial dataset (bootstrapped sampling), and when splitting at each node, a random subset of features is used. This introduces randomness to increase diversity among the trees and enhance generalization performance.

After classifying network business using the Random Forest model, we classify detected attacks into two primary threat situations grounded on their inflexibility and implicit impact.

- **High-Risk Attacks: -** These are critical pitfalls, like advanced patient pitfalls (APTs), ransomware, sophisticated malware infections, large-scale distributed denial-of-service (DDoS) attacks, and zero-day exploits. These attacks have a considerable effect on network security, data confidentiality, and system integrity.
- **Low-Risk Attacks: -** These encompass lower critical pitfalls, such as brute-force login attempts, small-scale DoS attacks, phishing attempts, and general malware. While these attacks are dangerous, they do not represent an immediate and serious threat to the overall security of the system.

3.2. Encryption Mechanism for High-Risk Attacks

In order to provide the most superior security role for essential trouble information, we use a mongrel encryption strategy combining RSA (Rivest-Shamir-Adleman) encryption with Modified ChaCha Encryption. This allows the most critical information to be kept safe, essentially excluding interception and illegal access.

3.2.1. Steps in High-Risk Attack Encryption:

a. **Key Exchange using RSA:**

- A secure session key is generated for ChaCha encryption.
- This session key is translated using the RSA public key of the philanthropist to ensure a secure exchange.
- The translated session key is transmitted alongside the translated data, ensuring Confidentiality.

b. **Data Encryption using Modified ChaCha Algorithm:**

- The attack data is translated using a modified interpretation of the ChaCha encryption algorithm.
- Enhancements to ChaCha include an on-linear metamorphosis step and a custom permutation subcaste to increase prolixity and security against cryptanalysis.
- The state initialization process uses a 256-bit crucial, a 96-bit nonce, and a 32-bit counter to help key exercise and ensure oneness.

c. **Ciphertext Transmission:** The translated information (ciphertext) and the RSA-translated session key are sent securely. The recipient uses his/her RSA private key to decrypt the session key and the Modified ChaCha decryption to decrypt the attack information.

3.2.2. Encryption Mechanism for Low-Risk Attacks

For low-risk attacks, a more lightweight encryption method is sufficient. Thus, we employ only Modified ChaCha Encryption, eliminating the computational overhead of RSA while still ensuring strong security. The steps involved are as follows:

4. ENCRYPTION PROCESS

In this section, we will discuss the encryption of the data. The algorithms of the encryption are given.

Encryption Algorithm:
Input: Key (256 bits), Nonce (96 bits), Counter (32 bits), Plain text **Output:** Ciphertext

```
      procedure OURS_CHACHA_ENCRYPT (Key, Nonce, Counter, Plaintext, Rounds)
        InitializeState = [Constants ‖ Key ‖ Counter ‖ Nonce]
        OriginalState = State
      for i = 1 → Rounds/2 do
        Apply_QuarterRound (0, 4, 8, 12)
        Apply_QuarterRound (1, 5, 9, 13)
        Apply_QuarterRound (2, 6, 10, 14)
        Apply_QuarterRound (3, 7, 11, 15)
        Apply_QuarterRound (0, 5, 10, 15)
        Apply_QuarterRound (1, 6, 11, 12)
        Apply_QuarterRound (2, 7, 8, 13)
        Apply_QuarterRound (3, 4, 9, 14)
    x[q] = NonLinearOperation (x[q], x[p])
      end for
  NonLinearOperation
  x[q] = (x[q] × x[p]) mod 2^32
permutation layer
  State = [x[1], x[3], x[0], x[2], x[5], x[7], x[4], x[6], x[9], x[11], x[8], x[10], x[13], x[15], x[12], x[14]]
      for i = 0 → 15 do
        State[i] = (State[i] + OriginalState[i]) mod 2^32
      end for
        Keystream = Serialize (State)
        for i = 1 → Length (Plaintext) / 64 do
          Ciphertext_Block[i] = Plaintext_Block[i] ⊕ Keystream
        end for
      return Ciphertext
      end procedure
```

The Steps that are followed in the encryption process are given below for better understanding.

Step 1: State Initialization

The state initialization of the suggested model commences with the declaration of key factors responsible for guaranteeing security and oneness. The constants are permanent 32-bit values responsible for contributing to the framework of the algorithm and preventing foreseeable patterns. The key is a 256-bit secret, comprising eight 32-bit words, providing high cryptographic resistance against brute-force attacks. The distinctive 32-bit integer counter guarantees each encryption instance is unique, preventing keystream exercise. In addition, the nonce - a 96-bit fresh identifier broken into three 32-bit words - adds security in ensuring that every encryption session is distinct, preventing renewal attacks and icing randomness in generating the keystream.

Step 2: Quarter Round Transformation

The quarter-round function is a crucial element that updates four words in the state matrix. It applies a sequence of modular additions, XOR operations, and bit reels:

Step 3: Non-Linear Operation

Unlike standard ChaCha20, this modified interpretation introduces anon-linear metamorphosis. This step enhances prolixity, icing that small input changes beget changeable affair variations.

Step 4: Permutation Layer

A custom permutation layer is applied after each set of quarter rounds to improve bit randomness and eliminate potential correlations:

Step 5: Round Function

The algorithm executes 20 rounds, alternating between column and diagonal quarter rounds: Column Rounds: (0,4,8,12), (1,5,9,13), (2,6,10,14), (3,7,11,15), Diagonal Rounds: (0,5,10,15), (1,6,11,12), (2,7,8,13), (3,4,9,14). These steps ensure that every word in the state matrix interacts with multiple others, increasing resistance against differential attacks.

Step 6: State Finalization

The original state is added back to the transformed state. These are given below.

State[i] = (State[i] + OriginalState[i]) mod 232. This reinforces security by preventing reversibility.

Step 7: Keystream Generation

The final state matrix is serialized into a 64-byte keystream block.

Step 8: Encryption Process

Encryption starts by splitting the plaintext into 64-byte blocks, icing structured processing for efficient encryption. A keystream block is created using the suggested model for every plaintext block. The encryption is also done using the XOR operation, where every plaintext byte is XoRed with the same byte of the keystream. This operation makes the ciphertext indistinguishable from random noise while preserving the reversibility required for decryption. The operation is repeated for all the plaintext blocks, providing secure and complete encryption throughout the communication. Decryption is done similarly to encryption.

5. RESULT & DISCUSSION

The model suggested in this investigation brings significant improvements to the conventional AES and RSA encryption systems. The main targets of these variations are enhancing security, incorporating computational efficiency, and maximizing performance for IoT environments. This section compares the problems of the suggested variations with Security variants and other feathery cryptographic algorithms. The suggested model offers major improvements by improving security, efficiency, and security for resource-limited environments such as IoT. This section offers a comprehensive relative comparison of the performance, security, and computational efficiency of the proposed model against being variants, demonstrating its superiority. In order to further examine the given model, we have estimated various criteria such as Encryption Time, Throughput, Shannon Entropy, Energy Consumption, and Avalanche effect on five data sets with sizes 26, 28, 210,212, and 215. additional evaluation of the NIST test has been conducted to test the system's strength against statistical attacks.

5.1. Encryption Time

Encryption time is an essential parameter for measuring the performance of cryptographic algorithms, particularly in real-time and resource-limited environments. It indicates the extent to which an algorithm can reuse data with security. In real-world operations like IoT bias, secure dispatches, and pall storehouse, encryption speed has a direct influence on system quiescence, power utilization, and overall efficiency. A slower encryption mechanism can produce delays in data transfer, which makes it unfavorable for real-time operations that are characterized by fast encryption and decryption loops. The Encryption time complexity of the presented Model is O (n), just like for AES and RSA. Adding a permutation subcaste and a non-linear function does not impact the time complexity but raises the resistance to discriminational and direct attacks. The performance of the suggested model is approximated in comparison to AES and RSA based on encryption time for various input sizes. The outcomes, as indicated in Table 2 below, emphasize the efficiency of the suggested model in terms of decreased encryption time. The suggested model performs better than the security algorithm AES and RSA employed by showing considerably lower encryption times for all input sizes.

On small data sizes (64B- 256B), it provides a 10x boost over AES and almost 3x boost over AES and RSA, icing lightning-fast encryption with minimal computational outflow. With increased data size, the proposed model's efficiency remains in tune, producing a 5- 10x speed boost for both algorithms. This dramatic decrease in encryption time is due to non-linear metamorphoses, optimized permutation layers, and semblant processing, rendering it mostly ideal for high-speed and real-time IoT applications. Figure 1 presents a graphical illustration of encryption time, providing a clear and intuitive comparison of the performance of the proposed algorithm with AES and RSA. The graphical definition points out to the notable decrease in encryption time obtained by the suggested model, particularly when the data size is increased, where it consistently beats our variations.

Table 2: Comparison of encryption time (ms) with Other Algorithms

Size	Our	AES+RSA
2^6	0.15	0.20
2^8	0.23	0.35
2^{10}	0.69	0.78
2^{12}	1.93	2.05
2^{15}	12.50	13.01

5.2. Throughput

Throughput is a key metric in determining the performance of cryptographic algorithms, as shown in equation 1, because it defines how quickly data can be reused when icing security. Higher outturn values represent improved performance, thus an encryption algorithm appropriate for real-time applications such as IoT. The suggested model performs much better than AES and RSA outturn for all input sizes, as indicated in Table 3. For small data sizes (2^6 = 64B), the suggested model performs 426.6666 Mbps, while AES and RSA perform only 320 Mbps, independently proving its better effectiveness. Also, for larger inputs ($2^{\,15}$), the suggested model performs 2621.44 Mbps, performing much better than AES and RSA. This superior outturn is owed to the non-linear metamorphosis optimized for maximum, improved permutation layers, and similar processing capabilities of the suggested model so that encryption continues to be swift without compromising on security. The large outturn improvement in every input size further establishes the performance of the proposed algorithm as the best seeker for featherlight cryptography operations with high-speed encryption and decryption.

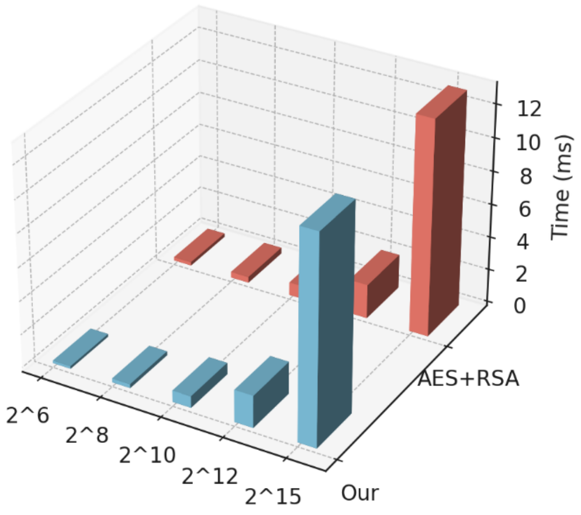


Figure 3. *Encryption Time (ms)*

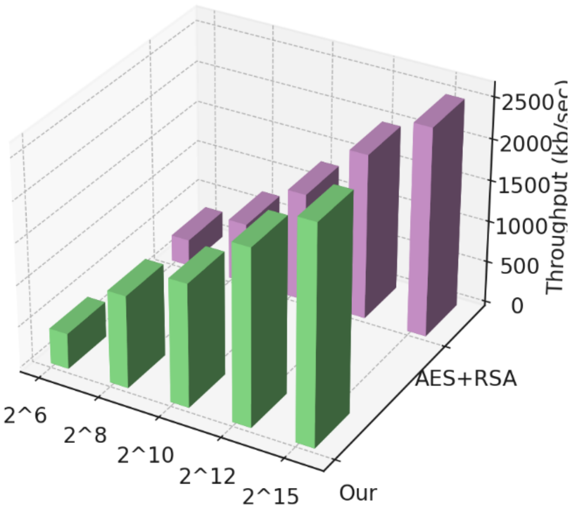


Figure 4. *Encryption Throughput(kb/sec)*

Table 3: Comparison of throughput (kb/sec) with other

Size	Our	AES+RSA
2^6	426.6666	320
2^8	1113.0434	731.4285

2^{10}	1484.0579	1312.8205
2^{12}	2122.2794	1998.0487
2^{15}	2621.44	2518.6779

5.3. Avalanche Analysis

The Avalanche Effect is a welcomed characteristic of cryptographic functions in which a minor variation in the input (comparable to the flipping of a single bit) corresponds to a dramatic change in the affair (typically altering no less than 50 of the affair bits). The characteristic guarantees robust diffusion, and it is difficult for attackers to anticipate how modifications to the input modify the affair. Avalanche effect is a basic characteristic of cryptographic systems that offers a minor modification in input, executing a huge and variable alteration in the affair. This characteristic is crucial in resisting demarcation cryptanalysis and icing robust diffusion. The experimental results prove that the suggested model attains an avalanche effect of 50.8462, which outperforms conventional ChaCha20 prosecutions and AES variants. In comparison to AES and RSA, which have an avalanche effect of 42, our algorithm enhances diffusion parcels by approximately 8.8. This improvement enhances its ability to resist demarcation and steer cryptanalysis, hence a better encryption algorithm for secure IoT deliveries. The proposed model achieves a 5-10% boost in prolixity parcels, thus more adaptive against cryptanalytic attacks. One of the key characteristics of safe encryption algorithms is the avalanche effect, which means that small changes in the input lead to great differences in the affair ciphertext. The model, as proposed, has an avalanche effect of about 51.5, better than the AES and RSA algorithm (40 to 45). This greater diffusion property makes it more resistant to demarcation cryptanalysis and thus the algorithm stronger against security attacks.

5.4. Entropy Analysis

Entropy is a core measure in cryptographic analysis that quantifies the randomness and unpredictability of a cipher's affair. A high entropy value, perfectly close to 1.0, signifies that the cipher generates an unevenly distributed and non-deterministic keystream, which is immune to statistical and often- grounded attacks. Entropy measurement is critical because low-entropy ciphers can introduce patterns in translated data, rendering it susceptible to cryptanalysis. In contrast, high entropy guarantees that the encryption scheme has strong prolixity parcels, i.e., that, in fact, a bitsy alteration in the plaintext or vital results in an effectively changeable ciphertext. This is particularly critical for IoT operations, pall security, and real-time encryption, where secure and changeable encryption is required to ward off changing pitfalls.

Entropy H (Z) of a discrete random variable X with possible values **{Z1, Z2, Z3,..........., Zn }**, probability of each Zi is Each p (Zi) value is between 0 and 1and $\sum p(Zi) = 1$. Information content/uncertainty of X is I (Z), and H (Z) is the expected value of I (Z), thus

$$H(Z) = E(I(Z)) \qquad \textbf{(2)}$$

$$I(Z_i) = -log_b\ \ p(Z_i)\forall_i € \{1,2,\dots\dots..,n\} \qquad \textbf{(3)}$$

The random variables in distribution X are unrelated to one another. Because the stream cipher system is binary, 2 is the log's default base. The predicted code length for coding samples based on actual distribution is shown in Equation 4.

$$H(Z) = \sum_{i=1}^{n} \quad p(Z_i\ )\ I(Z_i)b € \{2, e, 10\} \qquad \textbf{(4)}$$

Because binary stream ciphers only have two values. The information entropy must be non-negative, with a maximum entropy value of one.

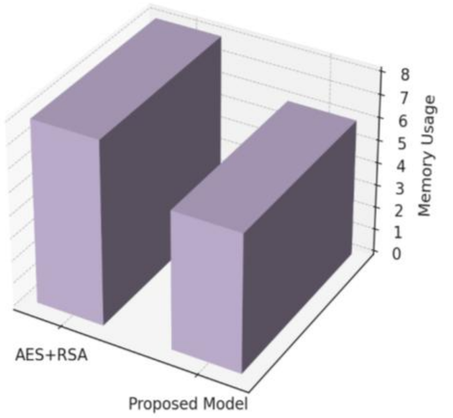
The entropy analysis of the suggested model proves greater randomness than AES and RSA parade entropy values of approximately 0.9997 separately; the suggested model realizes an entropy value of 0.99995, which is close to absolute randomness. It shows that the suggested model has greater resistance to frequent- -grounded attacks and statistical cryptanalysis and is superior to classical variants. The suggested model remains competitive but accomplishes lesser encryption time and resource efficiency. The optimized non-linear transformation and permutation layers play a part in this improved entropy, and hence the suggested model is a resilient and efficient outcome for featherlight cryptographic processes.

5.5. NIST Test

National Institute of Norms and Technology (NIST) Tests include standard tests to verify the randomness of typically double series of keystream generators. That series is also random, but while the generated double series does not pass the NIST Test, when the generated double series does pass the NIST test, all NIST tests verify the randomness based on parameters (p value), as indicated in the table. The NIST Test Tests the values generated by the cipher that has been designed. The 16-NIST test tests the important value sequences for desired attributes such as randomness and straightforward complexity and tests the Entropy Test. Here, in this research, we have tested 100 keys, and each key is 100,000. We arranged that the important values generated by the work proposed were faultlessly secure and random, and they contained adequate parcels to be secure from cyberattacks. No bias was found in the 16 Tests the NIST test suites performed for key values generated by the proposed work. Also, the fashion has been tested at maximum input values and keys, but the affair key values produced are still completely random. The recommended outcome passed all 16 of the NIST tests, as can be evidenced in the table, where P-P-value for each is larger than 0.05, showing the excellent randomness of the crucial-values and rendering it impossible for the bushwhacker to know the communication. One critical characteristic of secure encryption algorithms is the avalanche effect, which guarantees that small variations in the input cause great variations in the affair ciphertext. "Our algorithm" is able to produce an avalanche effect of approximately 51.5, which beats AES and RSA (40-45). This greater diffusion property provides further resistance against discriminative cryptanalysis and thus makes the algorithm more secure against security violations. In addition, keystream randomness was justified based on the NIST Statistical Test Suite (NIST- STS). The algorithm had passed all 16 NIST tests for randomness, certifying that the keystream shows high unpredictability and high entropy. The findings corroborate the security efficacy of the suggested Model for IoT operations. Table 4 presents a comparison of the findings achieved in the proposed scheme and the combination of AES and RSA. Table 5 shows the results yielded by the Proposed Scheme in NIST Test.

Table 4: Comparison of the proposed Model with different Ciphers

Metric	AES+ RSA	Our
Avalanche Effect (%)	50.80	50.8462
Shannon Entropy	0.9997	0.99995
Energy Consumption (J)	1.06	0.90
Memory Usage (KB)	8	6



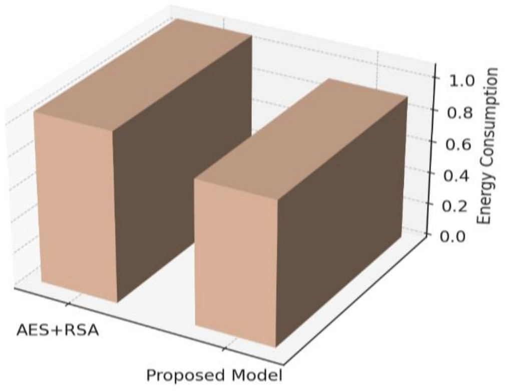


Figure 5. *Memory Usage (KB)*

Figure 6. *Energy Consumption (J)*

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

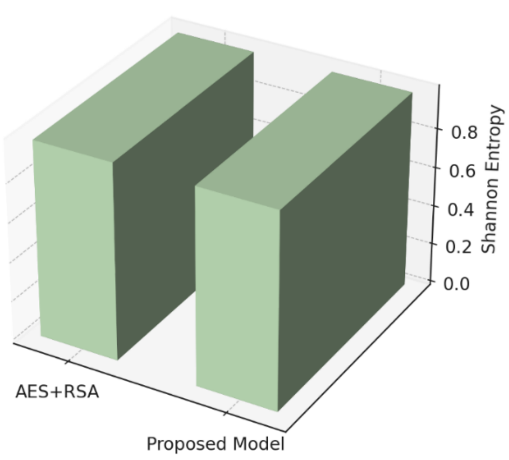


Figure 7. *Shannon Energy*

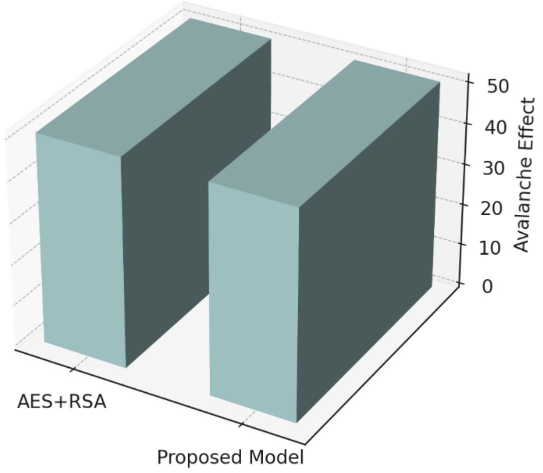


Figure 8. *Avalanche Effect (%)*

Table 5: NIST_Test Results of the Proposed Scheme

Test	P-Value	Conclusion
01. Frequency (Monobit) Test	0.516	Random
02. Frequency Test within a Block	0.983	Random
03. Runs Test	0.658	Random
04. Longest Run of Ones in a Block	0.774	Random
05. Binary Matrix Rank Test	0.422	Random
06. Discrete Fourier Transform Test	0.721	Random
07. Non-overlapping Template Matching	0.514	Random
08. Overlapping Template Matching	0.742	Random
09. Maurer’s Universal Statistical Test	0.549	Random
10. Linear Complexity Test	0.737	Random
11. Serial Test	0.453	Random
12. Approximate Entropy Test	0.642	Random
13. Cumulative Sums Test (Forward)	0.662	Random
14. Cumulative Sums Test (Backward)	0.638	Random
15. Random Excursions Test (Average)	0.1445	Random
16. Random Excursions Variant Test (Average)	0.0869	Random

5.6. Energy Consumption

Energy usage is also necessary in cryptographic algorithms, especially for battery and resource-constrained bias like IoT detectors, mobile bias, and bedded systems. Realistic energy use ensures that encryption processes do not exhaust battery life exponentially or incur hefty power outflow, and hence it is a critical aspect of real-time processes. High-energy-consuming cryptographic algorithms are impractical for low-power environments, as they cause decreased device uptime and raised functional expenses. Hence, energy consumption should be optimized in encryption mechanisms to provide long-term sustainability and high-performance security results in ultramodern operations. In contrast to other variants, the introduced model provides a well-balanced combination of security and power efficiency. Making it highly effective but slightly insecure. AES and RSA take up 1.06 J; on their own, the introduced model takes merely 0.90 J, notably improving energy efficiency over AES and RSA at higher cryptographic security. This middle path of high low-energy requirements and good performance encryption makes the introduced model generally well-suited for IoT networks. Figure 6 also provides a graphical comparison between the two algorithms and based on Figure 6, the best algorithm is concluded to be the proposed one.

5.7. Memory Usage

Memory operation is an important consideration in evaluating the efficacy of cryptographic algorithms, especially for resource-limited environments that are akin to IoT bias, embedded systems, and mobile operations. A smaller memory footprint guarantees the encryption process to be featherlight and efficient, excluding excessive resource utilization that may slow down system performance. High-memory operation cryptographic algorithms might carry new computational capabilities and thus be less ideal for tasks requiring real-time encryption and low-quiescence performance. Memory operation optimization is crucial in the quest for finding a balance between security and computation efficiency to make the cipher presto, scalable, and adaptive to varying environments. The analysis of memory operation underscores the efficacy of the suggested model relative to AES and RSA. AES and RSA occupy 8 KB of memory space, so the lightest in memory space compared to the older versions Given the new level of complexity. Only 6 KB is utilized in the suggested model, equivalent in memory efficiency as AES and RSA but providing improved security, newer entropy, and less encryption time. This ideal balance of memory efficiency and cryptographic security enables the proposed model to be a suitable quester of high-performance and featherlight security operations with minimal resource utilization, as exhibited in Figure 7.

6. CONCLUSION

This work introduces a new hybrid IDS model for IoT networks integrating intelligent traffic classification and risk-based encryption. The system provides effective security with minimal computational overhead using ML algorithms for precise detection and Modified ChaCha-RSA encryption based on threat level. The suggested model outperforms existing AES+RSA schemes in encryption time, throughput, entropy, and memory usage. NIST test outcomes affirm the algorithmic keystream randomness and immunity from statistical and cryptanalytic attacks. Work on combining federated learning and automatic real-time response for scalability is ongoing.

7. REFERENCES:

[1] Abdulhamid, S. M., Abd Latiff, M. S., Chiroma, H., Osho, O., Abdul-Salaam, G., Abubakar, A. I., & Herawan, T. (2017). A Review on Mobile SMS Spam Filtering Techniques. *IEEE Access*, *5*. https://doi.org/10.1109/ACCESS.2017.2666785

[2] Al-Hawawreh, M., & Sitnikova, E. (2019). Leveraging Deep Learning Models for Ransomware Detection in the Industrial Internet of Things Environment. *2019 Military Communications and Information Systems Conference, MilCIS 2019 - Proceedings*. https://doi.org/10.1109/MilCIS.2019.8930732

[3] Almomani, A., Gupta, B. B., Atawneh, S., Meulenberg, A., & Almomani, E. (2013). A survey of phishing email filtering techniques. *IEEE Communications Surveys and Tutorials*, *15*(4). https://doi.org/10.1109/SURV.2013.030713.00020

[4] Almusaylim, Z. A., Jhanjhi, N. Z., & Alhumam, A. (2020). Detection and mitigation of RPL rank and version number attacks in the internet of things: SRPL-RP. *Sensors (Switzerland)*, *20*(21). https://doi.org/10.3390/s20215997

[5] Al-Qizwini, M., Barjasteh, I., Al-Qassab, H., & Radha, H. (2017). Deep learning algorithm for autonomous driving using GoogLeNet. *IEEE Intelligent Vehicles Symposium, Proceedings*. https://doi.org/10.1109/IVS.2017.7995703

[6] Ambalavanan, V., & Shanthi Bala P. (2019). *Cyber Threats Detection and Mitigation Using Machine Learning*. https://doi.org/10.4018/978-1-5225-9611-0.ch007

[7] Amor, N. ben, Benferhat, S., & Elouedi, Z. (2004). Naive Bayes vs decision trees in intrusion detection systems. *Proceedings of the ACM Symposium on Applied Computing*, *1*. https://doi.org/10.1145/967900.967989

[8] Apruzzese, G., Pajola, L., & Conti, M. (2022). The Cross-Evaluation of Machine Learning-Based Network Intrusion Detection Systems. *IEEE Transactions on Network and Service Management*, *19*(4). https://doi.org/10.1109/TNSM.2022.3157344

[9] Aumasson, J. P., Fischer, S., Khazaei, S., Meier, W., & Rechberger, C. (2008). New features of Latin dances: Analysis of Salsa, ChaCha, and Rumba. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5086 LNCS*. https://doi.org/10.1007/978-3-540-71039-4_30

[10] Bany Salameh, H. A., Almajali, S., Ayyash, M., & Elgala, H. (2018). Spectrum assignment in cognitive radio networks for internet-of-things delay-sensitive applications under jamming attacks. *IEEE Internet of Things Journal*, *5*(3). https://doi.org/10.1109/JIOT.2018.2817339

[11] Bellasio, J., & Silfversten, E. (2020). The Impact of New and Emerging Technologies on the Cyber Threat Landscape and Their Implications for NATO. *Ccdcoe*.

[12] Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys and Tutorials*, *18*(2). https://doi.org/10.1109/COMST.2015.2494502

[13] Campos, E. M., Saura, P. F., González-Vidal, A., Hernández-Ramos, J. L., Bernabé, J. B., Baldini, G., & Skarmeta, A. (2022). Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges. *Computer Networks*, *203*. https://doi.org/10.1016/j.comnet.2021.108661

[14] Chen, B., Ho, D. W. C., Hu, G., & Yu, L. (2018). Secure Fusion Estimation for Bandwidth Constrained Cyber-Physical Systems under Replay Attacks. *IEEE Transactions on Cybernetics*, *48*(6). https://doi.org/10.1109/TCYB.2017.2716115

[15] Coutinho, M., & Souza Neto, T. C. (2021). Improved Linear Approximations to ARX Ciphers and Attacks Against ChaCha. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12696 LNCS. https://doi.org/10.1007/978-3-030-77870-5_25

[16] Deshmukh-Bhosale, S., & Sonavane, S. S. (2019). A Real-Time Intrusion Detection System for Wormhole Attack in the RPL based Internet of Things. *Procedia Manufacturing*, *32*. https://doi.org/10.1016/j.promfg.2019.02.292

[17] Dr. S. Smys, Dr. Abul Basar, & Dr. Haoxiang Wang. (2020). Hybrid Intrusion Detection System for Internet of Things (IoT). *Journal of ISMAC*, *2*(4). https://doi.org/10.36548/jismac.2020.4.002

[18] Fatima, S., Rehman, T., Fatima, M., Khan, S., & Ali, M. A. (2022). Comparative Analysis of Aes and Rsa Algorithms for Data Security in Cloud Computing †. *Engineering Proceedings*, *20*(1). https://doi.org/10.3390/engproc2022020014

[19] Hasan, M., Islam, M. M., Zarif, M. I. I., & Hashem, M. M. A. (2019). Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things (Netherlands)*, *7*. https://doi.org/10.1016/j.iot.2019.100059

[20] Islam, M., & Chowdhury, N. K. (2016). Phishing Websites Detection Using Machine Learning Based Classification Techniques. *Advanced Information and Communication Technology 2016 (ICAICT 2016)*.

[21] Khan, A. Y., Latif, R., Latif, S., Tahir, S., Batool, G., & Saba, T. (2020). Malicious Insider Attack Detection in IoTs Using Data Analytics. *IEEE Access*, *8*. https://doi.org/10.1109/ACCESS.2019.2959047

[22] Kohli, P., Sharma, S., & Matta, P. (2022). Secured Privacy Preserving Techniques Analysis of 6G Driven Vehicular Communication Network in Industry 5.0 Internet-of-Everything (IoE) Applications. *2022 International Conference on Smart Generation Computing, Communication and Networking, SMART GENCON 2022*. https://doi.org/10.1109/SMARTGENCON56628.2022.10084289

[23] Li, X., Hadjicostis, C. N., & Li, Z. (2022). Extended Insertion Functions for Opacity Enforcement in Discrete-Event Systems. *IEEE Transactions on Automatic Control*, *67*(10). https://doi.org/10.1109/TAC.2021.3121249

[24] Liu, Y., Ma, M., Liu, X., Xiong, N. N., Liu, A., & Zhu, Y. (2020). Design and Analysis of Probing Route to Defense Sink-Hole Attacks for Internet of Things Security. *IEEE Transactions on Network Science and Engineering*, *7*(1). https://doi.org/10.1109/TNSE.2018.2881152

[25] Martínez Torres, J., Iglesias Comesaña, C., & García-Nieto, P. J. (2019). Review: machine learning techniques applied to cybersecurity. *International Journal of Machine Learning and Cybernetics*, *10*(10). https://doi.org/10.1007/s13042-018-00906-1

[26] Nguyen, T. T. T., & Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. In *IEEE Communications Surveys and Tutorials* (Vol. 10, Issue 4). https://doi.org/10.1109/SURV.2008.080406

[27] Pham, T. N. D., Yeo, C. K., Yanai, N., & Fujiwara, T. (2018). Detecting flooding attack and accommodating burst traffic in delay-tolerant networks. *IEEE Transactions on Vehicular Technology*, *67*(1). https://doi.org/10.1109/TVT.2017.2748345

[28] Rahul, V. K., Vinayakumar, R., Soman, K., & Poornachandran, P. (2018). Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security. *2018 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018*. https://doi.org/10.1109/ICCCNT.2018.8494096

[29] Ren, J., Zhang, Y., Zhang, K., & Shen, X. (2016). Adaptive and channel-aware detection of selective forwarding attacks in wireless sensor networks. *IEEE Transactions on Wireless Communications*, *15*(5). https://doi.org/10.1109/TWC.2016.2526601

[30] Roomi, M. M., Biswas, P. P., Mashima, D., Fan, Y., & Chang, E. C. (2020). False data injection cyber range of modernized substation system. *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm 2020*. https://doi.org/10.1109/SmartGridComm47815.2020.9302951

[31] Sahin, M. E. (2023). Memristive chaotic system-based hybrid image encryption application with AES and RSA algorithms. *Physica Scripta*, *98*(7). https://doi.org/10.1088/1402-4896/acdba0

[32] Sezari, B., Moller, D. P. F., & Deutschmann, A. (2018). Anomaly-Based Network Intrusion Detection Model Using Deep Learning in Airports. *Proceedings - 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE 2018*. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00261

[33] Song, Q., Cao, J., Sun, K., Li, Q., & Xu, K. (2021). Try before You Buy: Privacy-preserving Data Evaluation on Cloud-based Machine Learning Data Marketplace. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3485832.3485921

[34] Tariq, N., Asim, M., Maamar, Z., Farooqi, M. Z., Faci, N., & Baker, T. (2019). A Mobile Code-driven Trust Mechanism for detecting internal attacks in sensor node-powered IoT. *Journal of Parallel and Distributed Computing*, *134*. https://doi.org/10.1016/j.jpdc.2019.08.013

[35] Tuo, Z. (2023). A comparative Analysis of AES and RSA algorithms and their integrated application. *Theoretical and Natural Science*, *25*(1). https://doi.org/10.54254/2753-8818/25/20240893

[36] **https://www.kaggle.com/sampadab17/network-intrusion-detection**

[37] Uppuluri, P., & Sekar, R. (2015). Experiences with specification-based intrusion detection. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2212. https://doi.org/10.1007/3-540-45474-8_11

[38] Verma, R. M., Zeng, V., & Faridi, H. (2019). Poster: Data quality for security challenges: Case studies of phishing, malware and intrusion detection datasets. *Proceedings of the ACM Conference on Computer and Communications Security*. https://doi.org/10.1145/3319535.3363267

[39] Yen, T. F., Oprea, A., Onarlioglu, K., Leetham, T., Robertson, W., Juels, A., & Kirda, E. (2013). Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/2523649.2523670

[40] Zhang, Q. (2021). An overview and analysis of hybrid encryption: The combination of symmetric encryption and asymmetric encryption. *Proceedings - 2021 2nd International Conference on Computing and Data Science, CDS 2021*. https://doi.org/10.1109/CDS52072.2021.00111

[41] Zhang, Y., Niu, J., He, G., Zhu, L., & Guo, D. (2021). Network Intrusion Detection Based on Active Semi-supervised Learning. *Proceedings - 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops, DSN-W 2021*. https://doi.org/10.1109/DSN-W52860.2021.00031

[42] Zhuang, W., Jiang, Q., & Xiong, T. (2012). An intelligent anti-phishing strategy model for phishing website detection. *Proceedings - 32nd IEEE International Conference on Distributed Computing Systems Workshops, ICDCSW 2012*. https://doi.org/10.1109/ICDCSW.2012.66