

Full Length Article

A hybrid machine learning and cryptography-based predictive probability model for enhancing security and privacy in cloud-IoT environment

Mayank Pathak^a, Kamta Nath Mishra^{a,*}, Satya Prakash Singh^a, Alok Mishra^{b,c,*} 

^a Department of CSE, Birla Institute of Technology, Ranchi, Jharkhand, India

^b Faculty of Engineering, Norwegian University of Science and Technology, Trondheim, Norway

^c Faculty of Logistics, Molde University College—Specialized University in Logistics, Molde, Norway

ARTICLE INFO

Keywords:

Cryptography

Data privacy

Data security

Distributed Denial of Service (DDoS) attack detection

Predictive probability

ABSTRACT

In the era of rapid technological advancement, many organizations shift their data to the cloud. At the same time, valuable data should be made available to various stakeholders when required for analysis, storage, and data consumption. Conversely, to attain security and preserve privacy while data is present on the cloud, data sharing involves significant challenges. In this research study, the authors have integrated cryptography technologies with probabilistic frameworks and machine learning (ML) to introduce a unique model that assists in secure data distribution. The model describes the rules and protocol based on which data can be allocated to various parties from the cloud. The proposed *HyMLCPP* (*Hybrid Machine Learning Cryptography Predictive Probabilistic*) model facilitates minimizing the computational time in digital image encryption. Further, it enhances data distribution and sharing practices among users while safeguarding privacy by integrating differential privacy techniques and ML models. The proposed model minimizes the risk by simultaneously detecting vulnerabilities in the cloud-IoT environment. The experimental outcomes demonstrate the proficiency of the proposed model over different self-generated and standard data sets. The proposed model was tested by the authors on standard datasets such as the *CICIDS 2017 Dataset*, *Kaggle Dataset*, *AIND DDoS Dataset*, and *CTU-13 Dataset* for DDoS attack detection. The proposed model shows accuracy and precision values of about 100% and 99.98% respectively, for standard and self-generated data sets. The proposed *HyMLCPP* model was tested using various accuracy measurement parameters such as *F1 score*, *recall value*, and *precision score*. The probabilistic features of the proposed model tend to identify the future events as a DDoS assault, which further makes the *HyMLCPP* model unique and stronger in comparison to the prior works that have proved the effectiveness.

1. Introduction

These days, cloud computing is one of the powerful technologies that makes data analysis, sharing, and storage easier. When it comes to processing, updating, and storing information more effectively and economically, cloud services are a huge benefit. Most government agencies and commercial enterprises migrate their data to the cloud due to its reliability and lower operational costs. These are sensitive medical, military, and defense data or images. Security is a highly demanding component of network and internet applications. Risks may arise after the data is moved to the cloud. After data is outsourced, a variety of entities have access to it for trade and other purposes. Because a third party manages the cloud platform, it is therefore impossible to completely trust it (Gupta et al., 2021). For security purposes, data must

therefore be encrypted by the data owner before being sent to the cloud. To achieve this security and put mathematical procedures into practice to encrypt data before uploading it to the cloud, cryptography is essential. Cryptography is considered one of the most popular techniques for providing better information-protecting operations in the form of encryption (ciphertext) and decryption (plaintext) over the network. AES (Advanced Encryption Standard), DES (Data Encryption Standard), 3DES (Triple DES), IDEA (International Data Encryption Algorithm), RSA (Rivest, Shamir, and Adleman), ECC (Elliptic Curve Cryptography), Homomorphic, and Blowfish are some examples of advanced encryption techniques that exist in contrast to some traditional ones (Sharma et al., 2022). The majority of currently available solutions for Cipher text policy attribute-based encryption (CP-ABE) and Key Policy attribute-based encryption (KP-ABE) rely on the idea that a

* Corresponding authors.

E-mail addresses: knmishra@bitmesra.ac.in (K.N. Mishra), alok.mishra@ntnu.no (A. Mishra).

<https://doi.org/10.1016/j.cose.2026.104870>

Received 2 June 2025; Received in revised form 24 September 2025; Accepted 22 February 2026

Available online 25 February 2026

0167-4048/© 2026 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

single key management center selects a master key at random and uses it to produce decryption keys for users (Shaikh et al., 2024). Unfortunately, these advances exhibited the uniqueness of information protection but did not address the key constraints. Chaotic techniques are similar to encryption techniques, which enable a high level of security for digital data. When the bulk of data is transmitted over the cloud, balancing between computational and security over data becomes a big challenge (Brunetta and Picazo-Sanchez, 2022). Computation over encrypted data is challenging (Fu et al., 2018). Hence, there is a requirement to protect the data of the owner as well as the data available on the cloud while carrying out the computation effectively. In addition to this, data ought to be mutually shared with all participants for its impartiality. Although the data is shared among legitimate enterprises, we cannot guarantee that our data will not be disclosed in the cloud (Gupta et al., 2019a). Therefore, in the context of the cloud, where a large volume of data is continuously transmitted, it is essential to safeguard that data from enterprises or entities interconnected with the cloud to prevent them from acquiring data for their own benefit. In IoT applications, where a large volume of data is transmitted, the trade-off between security and computational efficiency becomes particularly challenging. The challenges lie in addressing this time-intensive characteristic of chaotic systems to ensure their practicality in real-time scenarios (Shafique et al., 2024). So, there is an immediate requirement for efficient and computational encryption algorithms that can become a comprehensive analysis for data security. Here, we introduce a novel solution to data security that leverages Machine Learning (ML) and cryptography, facilitating secure cloud-based mutual data sharing, privacy-preserving data storage, data analysis, fault tolerance, and user detection. The integration of ML and cryptography is significant within the cloud domain. Both ML and cryptography are required to refine the big data that is uploaded to the cloud. ML approaches in cryptography are substantial within the cloud for mutual sharing and security of information. Adaptive learning techniques from a data-driven encryption model strengthen the security of data on the cloud. This correlation between ML and cryptography is the growing requirement for advanced encryption techniques to provide security and protection to the owner's data over the cloud. This correlation between ML and cryptography is the growing requirement for advanced encryption techniques to provide security and protection to the owner's data over the cloud. The hybrid combination of the ECC (Elliptic curve cryptography) + AES (Advanced Encryption Standard) algorithm and the application of logistic regression is applied to the data for security, and the system's performance is measured in terms of encryption and decryption time, and avalanche effect (Kumar and Goel, 2023). The full image takes a very long time to encrypt. One method that aims to reduce processing power, network traffic, and execution time is selective encryption (Kelur et al., 2019). In many applications, such as the military and medical fields, image security is crucial. ML models have demonstrated excellent results in various tasks, including the recognition of body organs from medical images, classification of lung diseases, detection of lung nodules, medical image reconstruction, and brain tumor segmentation (Qayyum et al., 2021). Additionally, since more medical data and pictures are private and sensitive, having them stored on a third-party platform will increase security risks. The confidential image leakage problem is resolved using the CNN (convolutional neural network) and enhanced by coupling with pre-processing (Liu et al., 2022). Hence, considering the vulnerability of sensitive medical images in the public domain, there is an urgent requirement to create a safer, effective, and efficient system for sharing and access across various stakeholders (Chenthara et al., 2019). Together, ML technologies and neural networks can solve difficult tasks that are essential for calculation and analysis, like the human brain. A goal of the suggested model is to provide a deep learning framework for privacy preservation by combining the Restricted Boltzmann Machine (RBM) with Instance Reduction (IR) Techniques (Alshammari and El Hindi, 2024). A framework for privacy-preserving credit risk analysis that utilizes Homomorphic Encryption aware

Logistic Regression (HELRL) on encrypted data was proposed (Divakar Allavarpu et al., 2024). Even with numerous encryption standards to safeguard sensitive data, a single encryption is insufficient to handle systems with high processing speeds. We have presented a novel approach in this research study to enhance privacy and data security. The presented approach offers a novel and effective solution to the complex problems of image feature classification in big data and cloud computing environments.

The main contributions of this research study are as follows:

- To enable secure sharing of digital images over the cloud. Images are encrypted using advanced technologies such as bit-plane and chaotic maps. It strengthens security in a cloud-based, distributed environment.
- Advanced access control mechanisms and noise insertion are implemented for better data distribution over the cloud with the help of differential privacy techniques. It enhances the security of data over the cloud.
- Several experiments are conducted using well-known standard data sets to analyze the significance of the proposed model in real-time. Additionally, the comparative analysis of the proposed model with various prior state-of-the-art models is studied and explained with various classifiers and datasets. The results demonstrate the model's effectiveness in multiple scenarios.
- To ensure secure and reliable cloud data distribution by applying appropriate protocols and guidelines, thereby safeguarding data transferred by the data steward.
- *Hybrid ML with privacy preservation*: We propose the HyMLCPP model, which uniquely integrates hybrid machine learning with collaborative privacy-preserving mechanisms to address both detection and secure cloud data distribution.
- *Performance superiority*: Compared to state-of-the-art methods (Table 6), HyMLCPP achieves 99.9% accuracy, 99.50% precision, 99.90% recall, and only 1.20% false positives, outperforming existing models in terms of detection reliability and efficiency.
- *Novel balance of performance and security*: Unlike prior works that emphasize either accuracy or encryption, our model ensures a balanced framework that delivers high detection rates while safeguarding data through protocol- and guideline-based mechanisms in cloud environments.
- *Adaptive Learning and Attack Generalization*: The proposed model can automatically learn patterns from data and adapt to new attacks during retraining.

The vulnerability is detected in the cloud when the data steward is unable to upload data to the cloud, which the proposed model identifies. Also, it can show predictions of future occurrences of such events and alert the data owner. The rest of this research study is structured and organized as follows: Section 2 presents the existing literature on cryptographic approaches for information security and privacy-enhancing methods in machine learning. In Section 3, we describe the problem identification in a cloud environment. Section 4 describes the proposed model along with the results and discussion of the experiment performed. Finally, this current research study ends with a statement of conclusion and future direction in Section 5.

2. Existing literature

2.1. Cryptographic approaches for information security: A review of encryption and decryption techniques

The ciphertext-policy attribute-based encryption (CP-ABE) is one method of data encryption that enables the specification of access policies in the ciphertext by the encryption provider (Wang et al., 2016). However, two shortcomings of typical CP-ABE techniques limit their applicability: the decryption time is linear with policy complexity,

meaning that significant processing overheads result; and the access policy is public, meaning that privacy is revealed. Using keyword search (KS) approaches, the author presents a novel way to safeguard the confidentiality of the CP-ABE system (Meng et al., 2021). Two access policies—one public and the other sensitive and hidden—are placed in the ciphertext to form a novel security paradigm we name the chosen sensitive policy security. The proposed model by the researcher has the same characteristics as CP-ABE and has a faster processing speed than when only CP-ABE is used (Lee et al., 2021). The author Kaur et al. (2023) employs a triple encryption approach before uploading data to the cloud, incorporating a hash value to ensure confidentiality, data integrity, and authenticity. A framework implemented by the author utilizes a multi-encryption and OTP (One-Time Password) verification technique (Anil et al., 2020). The research study explains what watermarking is and how digital watermarking may be used to improve data security. The author provides Layer privacy and data security to preserve confidential data over the cloud. In this approach, data are categorized according to the level of confidentiality. The watermarking technique is used for secret information (Mishra et al., 2023; Gupta et al., 2019b). The author developed a novel technique by using a Hybrid Cat-Boot Algorithm (HCBA) in which data is uploaded to the cloud using the Bring Your Own Encryption Key (BYOEK) (Priya and Yamuna, 2022). Many researchers have worked on secure image transmission over the cloud. The authors provide a steganography model called HIGAN in their research study. It utilizes an encoding network comprising residual blocks to conceal a color secret image within another color image of the same size, resulting in a steganographic image with reduced distortion and improved visual clarity (Fu et al., 2020). Another author (Jia et al., 2023) introduced a method for classifying image features by segregating multilevel homomorphic encryption and partitioning data images. The author introduced a model, which is known as the cloud-based buyer-seller watermarking protocol (CB-BSWP), where a semi-trusted third party is used for preserving privacy in the cloud environment (Kumar, 2022).

Furthermore, the proposed model utilizes the Diffie-Hellman key exchange method and a privacy homomorphism cryptosystem to create a secure encrypted domain for sharing digital media. This domain employs fair and robust watermarking techniques to ensure the high imperceptibility and resilience of the watermarked images against attacks. The manuscript presented by the author Naik and Singh (2024) shows how a chaotic map can be implemented to generate pseudo-random numbers to perform encryption on multimedia. A hybrid image-key model customized for text cryptography (Khan et al., 2024). To enable genuine users (hospital employees) to access patients' highly sensitive medical data (images) securely over the cloud. This research study proposes a communal blind crypto-watermarking technique that protects the ownership and authenticity of patient medical information, while also ensuring security (Kaur et al., 2020). Several authors have also discussed the use of watermarking to improve image security and data encryption techniques in cloud environments, which are practical and efficient (Ge et al., 2024; Alomoush et al., 2023; Gupta et al., 2022a; Boujerfaoui et al., 2023).

2.2. A review of privacy-enhancing methods in machine learning

The author introduces a novel scheme that can safeguard datasets in the cloud, as well as those of a cloud provider, using a public key encryption with a double decryption algorithm (DD-PKE) approach. To enhance the accuracy and efficiency of encrypted data in the presence of noise, we utilize an ML (Machine Learning) algorithm on this noisy data, in conjunction with ϵ -differential privacy (Ping et al., 2018). A machine learning model designed by the authors is known as MLPAM (Machine Learning and Probabilistic Analysis-based Model), used for safeguarding information through privacy-preserving data storage, analysis, secure data sharing, and recognizing guilt units against data leakage (Gupta et al., 2021). The authors designed a model for handling confidential

genomic data, known as Geno-PPML. It combines completely homomorphic encryption based on TFHE (Fully Homomorphic Encryption over the Torus) for model inference with secure multiparty compute techniques based on the recently proposed Manticore framework for model training (Carpov et al., 2022). ML-Crypto is a tool implemented by the author to guess the class, also known as a distinguisher correctly. The author presented an approach using ML to generate a classifier that can able to solve distinguishing problems (Brunetta and Picazo-Sanchez, 2022). The researcher presented a model that classifies data into different categories, such as sensitive and non-sensitive parts of data, then inserts noise in sensitive data, and then classification is performed with the help of k -anonymization, differential privacy, and ML algorithms (Singh and Gupta, 2022). For image data security, privacy, and computing efficiency, the author Jia et al. (2023) showcases a novel approach for classifying image features that synchronize multilevel homomorphic encryption with image data partitioning. CNN (Convolution Neural Network) is implemented for training and testing the method. The Federated Learning (FL) paradigm is an Artificial Intelligence (AI) method introduced by researchers for solving data warehouse privacy problems. In this approach, FL (Federated Learning) protects data by utilizing secure multiparty computation (MPC), differential privacy, and hardware solutions to apply distributed multiparty ML systems and statistical models across datasets. Through the FL (Federated Learning) paradigm, ownership verification is performed by embedding watermarks into the Federated Learning (FL) model (Yang et al., 2023). One of the models for data security and privacy is presented by the author through a research study that employs a multi-modal approach, coupling deep learning, classification, and clustering (El-Kassabi et al., 2023). Through the survey research study, the author (Qayyum et al., 2020) tried to present the various methods to ensure safeguarding and privacy-preserving ML techniques in the domain of healthcare. In this research study, a new symmetric image encryption technique with both diffusion and confusion components is presented. It utilizes a multi-layered design of chaotic maps (Güvenoglu, 2024). In the research study Kumar and Revathi (2024), researchers coupled the Deep Neural Networks (DNNs) principle with a discrete *memristor* (a circuit unit with memory that shows the interconnection between charge and magnetic flux), a logistic chaotic map to improve the security of the data. The author suggested several security precautions, including the use of a Support Vector Machine (SVM) and the image encryption technique (Shafique et al., 2021). The method used by the author is a hybrid random forest and logistic regression (HRFLR) model to solve the identification problem (Yuan et al., 2024). This article presents a comparative and systematic study, and in-depth analysis of leading techniques for secure sharing and protecting the data in the cloud environment (Gupta et al., 2022b). In this research work, the application of deep learning (DL) models in a known plaintext setting was examined by (Chong and Salam, 2021). The models' objective was to use DL approaches to predict a cipher's secret key (Chong and Salam, 2021). Homographic Encryption and neural network-based classification models were proposed by researchers to preserve privacy in a cloud environment (Akram et al., 2024). An in-depth analysis of the security model has presented a detailed description of PPDL Privacy-Preserving Deep Learning (PPDL) and addresses the issue of using PPDL for ML as a Service (MLaaS) (Tanuwidjaja et al., 2020). To ensure a secure communication system in the cloud environment, the authors propose a new security design that uses Artificial Neural Networks (ANN) and encryption. The Matrix Operation-based Randomization and Encipherment (MORE) method, which is based on Fully Homomorphic Encryption (FHE), for training neural networks using encrypted input (Sana et al., 2021). An explanation of one LMS Learning Management Systems (LMS) model that might offer its customers the best possible security, a user-friendly interface, and speedy access has been attempted (Chatterjee et al., 2023). The author presented a forensic technique using ML to detect the post-processing of digital videos which is exposed to manipulation. These approaches will help to make fair

decisions and strengthen justice in administration (Sandoval Orozco et al., 2020). In addition to the aforementioned research study, several researchers have presented the effective use of ML algorithms and cryptography in conjunction to enhance image security and data encryption in a cloud environment (Fkirin et al., 2022; Al-Ghuwairi et al., 2023; Garima and Quraishi, 2022; Pulido-Gaytan et al., 2021; Terziyan et al., 2023). A mathematical model for distributed denial-of-service attacks is proposed in the study. The ML algorithms, such as Logistic Regression and Naive Bayes, are used to detect attacks and normal scenarios (Kumari and Mrunalini, 2022). The DDoS attack on an SDN (Software Defined Networking) environment and develop a method to identify the DDoS attack using Deep Learning (DL). The proposed method converts the captured raw network traffic to image data and classifies the malicious data from normal data (Clinton et al., 2024). These classifiers can be used with different feature engineering and feature selection strategies to create ML models that effectively handle certain security issues and enhance overall cybersecurity posture (Kareem and Jasim, 2021; Pathak et al., 2024).

3. Problem identification

Problem identification defines the modules that are associated with the model, along with their specified role, in addition to various types of vulnerabilities that may occur in the protocol, explains issues, and highlights the design objectives.

3.1. System architecture related issues

The system architecture consists of four entities: Data Steward (DS^{id}), Cloud Sphere (CS), End User (EU^{id}), and Unauthorized Party (UnP^{id}). The description of System Architecture Related Issues (SAR_i) and entities is described below:

SAR_{i1} : Data Steward 'DS^{id}' (It is a component of the architecture that uploads data and requests services from Cloud Sphere (CS). DS^{id} encrypts data before moving it to the cloud by using encryption techniques. DS^{id} may not see their data, but DS^{id} can see other Steward information. Therefore, DS^{id} is considered to be a trustworthy component. It can be represented by Eq. (1).

$$DS^{id} = \{DS^1, DS^2, DS^3, \dots, DS^n\} \quad (1)$$

SAR_{i2} : Cloud Sphere 'CS^{id}' (It is an important component of the architecture that accumulates various encrypted data from DS^{id} and facilitates storing of data, performs computation over data, and shares data. After receiving the ciphertext from DS^{id} , the cloud sphere performs specific computations. The cloud sphere applies ML techniques and trains the data after getting information. The CS^{id} consists of four clouds, namely Public, Private, Protected, and Hybrid. Here, Cloud One and Cloud Two contain data in fragmented forms and service agencies. Cloud three is used for computation and encryption. Cloud four uses ML-based classifiers. Further, it can be represented by Eq. (2) and Eq. (3).)

$$CS^{id} = \{CS^1, CS^2, CS^3, \dots, CS^n\} \quad (2)$$

$$\text{Types of Cloud Sphere} \{PuCUPrCUPcCUHyC\} \quad (3)$$

where $PuC = \text{PublicCloud}$, $PrC = \text{PrivateCloud}$, $PcC = \text{ProtectedCloud}$, $HyC = \text{HybridCloud}$

SAR_{i3} : End User 'EU^{id}' (The encrypted data is available to the end user from the cloud sphere, which can be decrypted using a key by the end user, and therefore it is possible to read the data. Further, it can be represented by Eq. (4).)

$$EU^{id} = \{EU^1, EU^2, EU^3, \dots, EU^n\} \quad (4)$$

SAR_{i4} : Unauthorized Party 'UnP^{id}' (It can access important information and can grab a set of data from an authorized user. Therefore, the unauthorized party is considered to be a doubtful entity. It can be represented by Eq. (5).)

$$UnP^{id} = \{UnP^1, UnP^2, UnP^3, \dots, UnP^n\} \quad (5)$$

SAR_{i5} : Data 'D' (Data in the system can be presented in several formats, such as textual data, image data, audio data, video data, real-time data, structured data, unstructured data, graphical data, etc. Handling various formats of data is challenging. It can be represented by Eq. (6).)

$$D^{id} = \{D^1, D^2, D^3, \dots, D^n\} \quad (6)$$

SAR_{i6} : Encrypted Data 'ED^{id}' (The data generated by the data steward is encrypted through the encrypted techniques applied to the data. It can be represented through Eq. (7).)

$$ED^{id} = \{ED^1, ED^2, ED^3, \dots, ED^n\} \quad (7)$$

SAR_{i7} : Non-Fault User 'NF^{id}' (The non-fault user is not a data leaker, and does not disclose the data but uses it appropriately. It can be represented by Eq. (8).)

$$NF^{id} = \{NF^1, NF^2, NF^3, \dots, NF^n\} \quad (8)$$

SAR_{i8} : Fault User 'F^{id}' (The fault user leaks data. It can be represented by Eq. (9).)

$$F^{id} = \{F^1, F^2, F^3, \dots, F^n\} \quad (9)$$

3.2. Threat analysis related issues

The Hybrid Machine Learning Cryptography Predictive Probabilistic (HyMLCPP) model is an authorized entity that can use or access data acquired by the DS^{id} . Below mentioned are the Threat Analysis Related Issues (TAR_i) that may occur in the HyMLCPP model:

- TAR_{i1} :** UnP^{id} can distort data and may be misused by others.
- TAR_{i2} :** UnP^{id} can perform data manipulation. The classifier captures the data transmitted by the service provider.
- TAR_{i3} :** UnP^{id} can convince the end user to obtain data.
- TAR_{i4} :** UnP^{id} can control the data transmission among different entities, such as DS^{id} , CS^{id} , and other classifiers.
- TAR_{i5} :** UnP^{id} can directly grab the data from DS^{id} .
- TAR_{i6} :** UnP^{id} manipulates data through vulnerabilities using CS^{id} .
- TAR_{i7} :** UnP^{id} may leak the data or broadcast it to others who can make a profit by selling it to different organizations or institutions.

3.3. System design objectives and research queries

There are data steward processes that manage data that is distributed or shared amongst various end users. During this process, the data steward comes across multiple issues and challenges that need to be eliminated. The description of System Design Objectives and Research Queries (SD_{ORQ}) are described below:

- SD_{ORQ1} :** How to safeguard data D^i from Data Steward (DS^{id}) who can expose data to some other entity within the cloud domain?
- SD_{ORQ2} :** How to mitigate the owner's data that is being shared in the form of ciphertext?
- SD_{ORQ3} :** Encrypted data ($ED^1, ED^2, ED^3, ED^4, \dots, ED^n$) is shared with the public and private key. So, CS^{id} and UnP^{id} can acquire data ($D^1, D^2, D^3, \dots, D^n$)

The following are the Design Goals (DGs) of HyMLCPP based on the mentioned problem identification and threat analysis model:

DG_1 : To enable the data steward to share data with appropriate security and privacy.

DG_2 :To share the data steward without unauthorized access and differentiate between fault users and non-fault users.

DG_3 : To perform calculations over data (D^{id}) for improving data preservation effectively.

DG_4 :To maintain the confidentiality of data (D^{id}) during its sharing and distribution with the authorized person.

4. Proposed model

4.1. Deep analysis of the proposed HyMLCPP model

Table 1 presents an in-depth analysis of the problem and threats associated with the existing system. It also presents the HyMLCPP model as a solution to the issues.

4.2. Proposed HyMLCPP model and system design

The HyMLCPP combines ML techniques and cryptography methods to encrypt data, minimize computational time, and enable the fair distribution of the data. The proposed model allows the fault user categorization $\{F^1, F^2, F^3, \dots, F^n\}$ and non-fault users, $\{NF^1, NF^2, NF^3, \dots, NF^n\}$, who are leaking or exposing data in a cloud environment. In the proposed model, let us consider that the data steward (DS^1) owns the data (D^1), where each data is considered to be an individual unit. Fig. 1 shows various components that interact with the proposed model.

The proposed HyMLCPP model and system design are divided into different key phases (Phase I to Phase IV):

Phase I: Cryptographic Techniques for Data Security.

Phase II: Classification of Data.

Phase III: Data Allocation.

Phase IV: Identifying Faulty and Non-Faulty Users.

Fig. 1 represents the proposed architecture of the HyMLCPP model. This figure shows a basic building block of the proposed model. It shows the various components of the model that communicate and interact directly with the proposed model. The HyMLCPP model enables safe and secure data transmission, data sharing, data distribution, and data safeguarding. It predicts vulnerabilities in data and enables the detection of both guilty and non-guilty users through a classifier. The proposed model facilitates full-fledged data security in a cloud environment by implementing ML and cryptography probabilistic approaches. The proposed model has three essential layers where *Layer 1* illustrates the components that directly interact with the model, *Layer 2* receives input and provides a response to the entity, and *Layer 3* defines the outcome provided by the model. The proposed model ensures safe and secure data accessibility, preventing data leakage by unauthorized users. The Fig. 1 shows the different functions of the proposed model in the cloud-IoT environment. The layers are interconnected to make the system stronger and increased reliability. This structure permits the proposed model to manage data, ensure security, and get used to changes effectively. The design also wires smooth operations across the cloud-IoT systems.

Fig. 1 provides an answer to the issues stated in Table 1. It brings deep drives into the proposed model by implementing ML techniques and cryptography methods for finding resolutions to the issues (Issue 1 to Issue 8) highlighted in Table 1.

Table 1

Identified problems in cloud security and solutions using the HyMLCPP model.

Issue No.	Problems Formulation	Issue Categorization	Description	HyMLCPP Solution
Issue 1	Vulnerabilities in Data Transmission and Storage	TAR_{t_4}	Sensitive data transmitted or stored in the cloud is at risk of unauthorized access and tampering due to weak encryption.	The HyMLCPP model combines cryptography with ML to predict vulnerabilities and enhance encryption in a real-time environment.
Issue 2	Inefficient Detection of Anomalous Activities	TAR_{t_1}	Static methods for intrusion detection often fail to detect advanced or new threats, such as zero-day attacks.	The model improves anomaly detection by analyzing user behavior, system logs, and network traffic with machine learning.
Issue 3	High Computational Costs of Security Mechanisms	SAR_{t_1} SAR_{t_6}	Traditional cryptography methods can be resource-intensive, causing performance issues in cloud services.	HyMLCPP optimizes cryptographic processes using ML to apply encryption based on the predicted sensitivity of data selectively.
Issue 4	Lack of Real-Time Threat Detection and Response	TAR_{t_5} TAR_{t_6}	Many systems are unable to predict and counter threats before they cause damage.	The predictive probability model enables early detection and proactive response to threats, thereby reducing the likelihood of breaches.
Issue 5	Privacy Concerns from Data Aggregation	$SDOR_{q_1}$ $SDOR_{q_2}$ TAR_{t_7}	Aggregated data used in ML models can unintentionally leak sensitive information.	Cryptographic techniques ensure that sensitive data is protected during ML processes, preserving privacy without sacrificing performance.
Issue 6	Balancing Security with User Experience	$SDOR_{q_3}$	Increased security measures can negatively impact legitimate users, complicating access and user experience.	HyMLCPP leverages ML to balance security needs with user behavior, allowing seamless access for legitimate users while blocking attackers.
Issue 7	Scalability Issues in Multi-Cloud Environments	SAR_{t_2}	Many security solutions fail to scale effectively across different cloud providers,	HyMLCPP provides adaptable security mechanisms that scale

(continued on next page)

Table 1 (continued)

Issue No.	Problems Formulation	Issue Categorization	Description	HyMLCPP Solution
			leading to security gaps.	across multiple cloud environments, offering unified security protection.
Issue 8	Categorize fault users and non-fault users	SAR_{I_f} SAR_{I_n}	It is essential to identify who is responsible for data leakage. Many of the existing system fails to identify the actual data leaker in the system.	HyMLCPP provides mechanisms that enable to differentiate between fault and non-fault users.

4.3. Work flow of the proposed HyMLCPP model

To improve privacy and security in cloud-IoT environments, the HyMLCPP model architecture integrates ML algorithms and cryptographic techniques faultlessly. It works by regularly monitoring data flow, observing irregularities, forecasting possible security threats, and dynamically modifying encryption settings as per the type and sensitivity of the data. This technique provides a balance between security and system performance, and ensures real-time threat identification and effective resource management. The HyMLCPP model employs

necessary steps such as data pre-treatment, threat prediction, cryptographic optimization, and continuous monitoring for establishing a safe and scalable cloud computing environment.

4.4. HyMLCPP system flow and architecture

Fig. 2 describes the architectural design and operational task flow of the proposed HyMLCPP model. It explains the process of data generated by data stewards from local or remote locations. The generated data is encrypted and preserved on the cloud by the user. The proposed model operates on the cloud. It permits the appropriate data distribution by safeguarding the security of the data without any data leakage. The proposed model is capable to detect any vulnerability in the data. The proposed model is also efficient in detecting modified images or distorted images. Further, this model can differentiate between non-guilt users and guilt users. The system also assures that sensitive information remains secure during complicated classification processes, and it implements privacy-preserving methods at various stages of communication.

Furthermore, the HyMLCPP model incorporates advanced machine learning techniques with cryptographic approaches to support a multi-layered security system. Its versatile architecture enables it to be resilient to numerous cloud environments and configurable for massive datasets. The proficiency to perform real-time monitoring enhances its reliability in dynamic data distribution scenarios. Altogether, the model provides trust, resilience, and efficiency in a cloud environment.

System Flow (SF): The SF encompasses data collection, data pre-processing, a Classifier, and a privacy-preserving approach. The

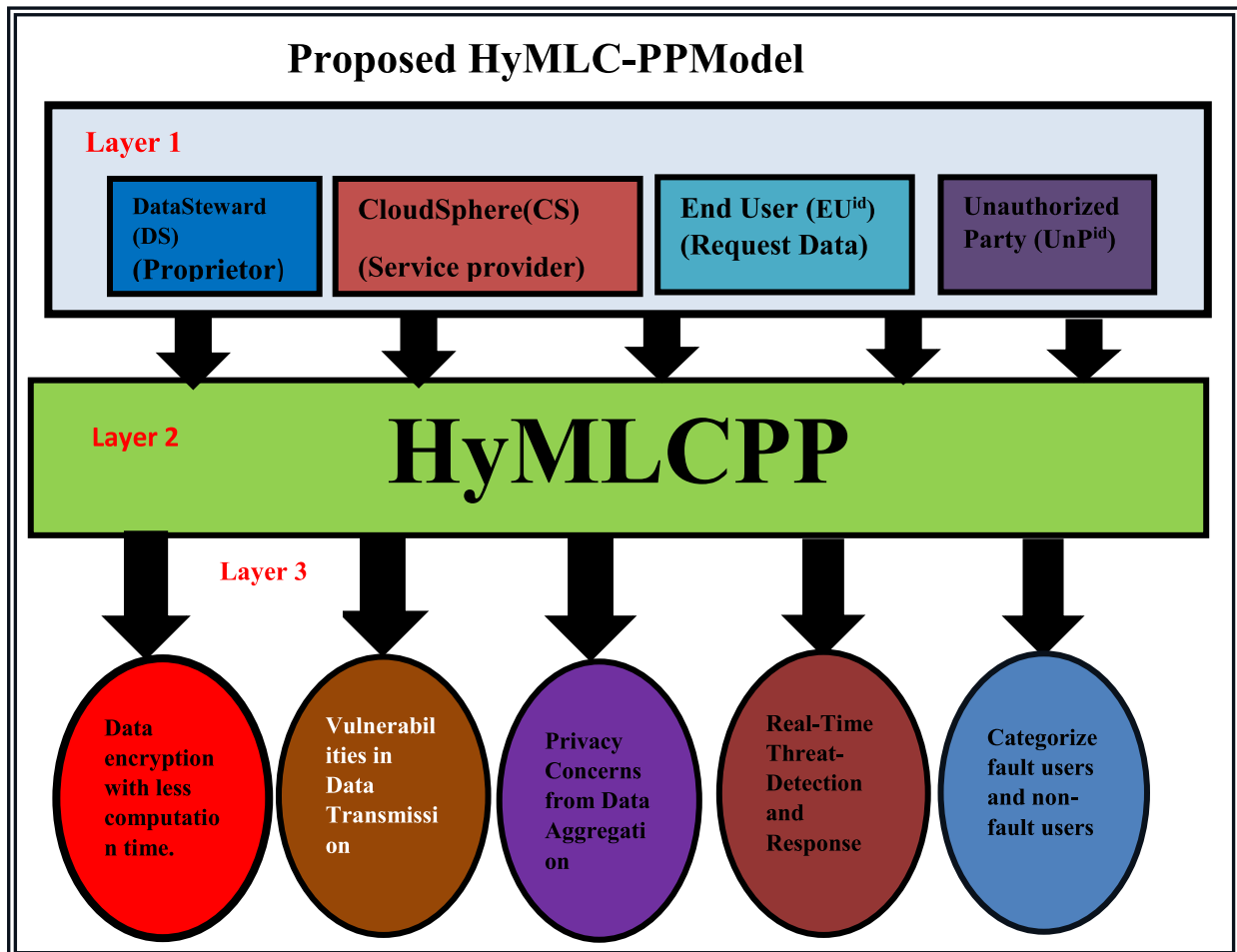


Fig. 1. Proposed HyMLCPP model.

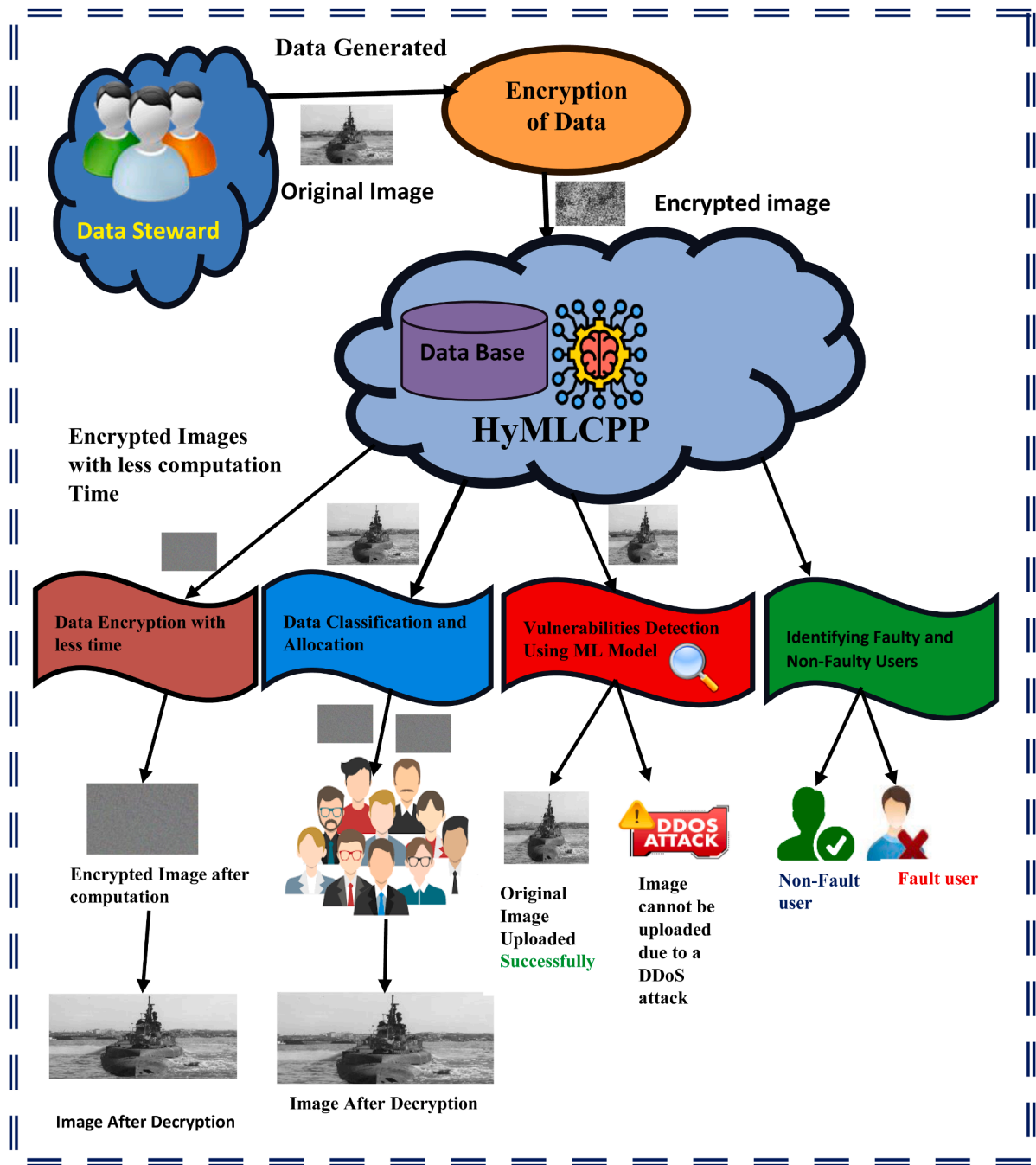


Fig. 2. Proposed model system flow and architecture.

following are the essential system flow phases (SFP_s):

SFP_1 : *Data Collection*: The process is initiated by accumulating data from various sources, such as IoT devices, cloud storage, and can include a mix of confidential and non-confidential information.

SFP_2 : *Pre-processing of Data*: Pre-processing is performed on the accumulated data before classification. Getting the data prepared for the ML models encompasses cleaning, normalizing, extracting features, and encoding. Here, private-preserving approaches, such as encryption or data anonymization, can be used to protect sensitive data.

SFP_3 : *HyMLPP Classifier*: It is a hybrid classification model that integrates different ML techniques. The system implements it as support vector machines (SVM), decision trees (DT), and neural

networks (NN). The layer-based model is pre-trained to classify the data precisely.

SFP_4 : *Mechanism to Preserve Privacy*: Sensitive data is protected from exposure by applying privacy-preserving techniques, including safe multiparty computation, homomorphic encryption, and differential privacy.

5. Cryptographic techniques used in proposed HyMLCPP model

To provide security to the digital images, the following standard approaches (SA) are used to minimize the computational intricacy and alter the digital image pixels in the proposed work:

SA₁: The ML Classifier Support Vector Machine (SVM).

SA₂: Chaotic Map-Based Modified Pulse-Coupled Spiking Neuron Circuit.

SA₃: Bit Plane Decomposition.

Before transmission, the data generated at the user end undergoes encryption to ensure confidentiality. An AES-256 symmetric encryption scheme is employed, with cryptographic keys being securely generated and managed at the user/system level. The encrypted information is subsequently transferred to the cloud via a secure TLS/SSL communication channel and retained in its encrypted form within the storage environment. During the retrieval phase, only authenticated users in possession of the correct decryption key can reconstruct the original data. This security framework strengthens data confidentiality and safeguards against unauthorized disclosure, even in scenarios where the cloud infrastructure is compromised.

5.1. Machine learning classifier support vector machine (SVM)

The primary goal of SVM is to maximize the margin between data points of distinct classes while determining the optimal decision boundary (hyperplane) that divides them. The proposed HyMLC-PP model classifies the digital image into two categories, i.e., a portion of a plaintext digital image is intended to be encrypted or not. To classify, data points can be separated using Eq. (10).

$$Z = V * M + a \quad (10)$$

In Eq. (10), 'Z' is the hyperplane, whereas 'V' represents the hyperplane's angle in 'D' dimensional space. 'M' represents input, and the threshold is represented by 'a'. The Threshold value indicates the hyperplane's separation from its starting point. Eqs. (11) and (12) represent the standard hyperplane for both negative and positive values.

$$Z - = V * M + a = -1 \quad (11)$$

$$Z + = V * M + a = +1 \quad (12)$$

A negative value indicates that the plaintext portion does not require encryption by the encryption algorithm. A positive value represents that the plaintext portion is mandatory to be encrypted by the encryption algorithm.

The SVM is the point of data used in Eqs. (11) and (12).

The objective of SVM is to reduce the factor. The support vectors primarily determine the position and direction of the hyperplane. The distance from the closest data point of either class (support vectors) to the hyperplane is known as the margin. A higher margin usually results in better generalization on unknown data, which is what the SVM algorithm seeks to maximize.

$$\text{Margin gap} = \frac{2}{\|V\|} \quad (13)$$

Fig. 3 illustrates the support vector organization, where line 'M' represents the margin that indicates the distance between the two classes. The line 'M' is the hyperplane that divides various courses. The best hyperplane to optimize the margin between classes is found using SVM. To calculate this decision boundary, it utilizes the support vectors. The Line 'N' that is available below the margin represents the specific section of the digital image that was prohibited from encryption. Line 'P' is a positive value, indicating a section of digital photos that requires encryption. This classification procedure guarantees that sensitive areas of the image are selectively safeguarded while non-critical portions remain accessible. By maximizing the margin, the SVM augments the accuracy of classification and minimizes the risk of misclassification. The disparity between encrypted and non-encrypted sections also enhances computational efficiency by neglecting unnecessary overhead. Overall, this approach furnishes a balanced trade-off between performance and security in digital images.

In the proposed HyMLC-PP model, the Support Vector Machine (SVM) is deployed to classify digital image sections into two broad categories: sections of plaintext that are recommended to be encrypted and those that are not, based on learned feature representations. As described in Eqs. (10)–(13), the proposed model depends on a linear decision boundary, with $Z = V * M + a$ defining the hyperplane in D-dimensional space.

The following are the SVM implementations that are used for kernel and hyperparameter configuration:

- **Kernel Type (linear):**

- This selection was steady with the theoretical formulation in Eq. (10), which presumes a linear segregation in the input feature space.
- The experimental observation proved that the data were **linearly separable**, and the linear kernel generated robust and interpretable outcomes without spare kernel transformation overhead.

- **Hyperparameter Settings:**

- $C = 1.0$: Regularization parameter controlling the trade-off between maximizing the margin and minimizing classification errors. This default setting provided a good balance between model complexity and generalization.
- $\text{Tol} = 1e-4$: Tolerance for stopping criterion to ensure convergence without overfitting.
- $\text{max_iter} = 1000$: Sufficient number of iterations to reach optimal separation in all tested samples.
- $\text{kernel} = \text{'linear'}$

- **Justification:**

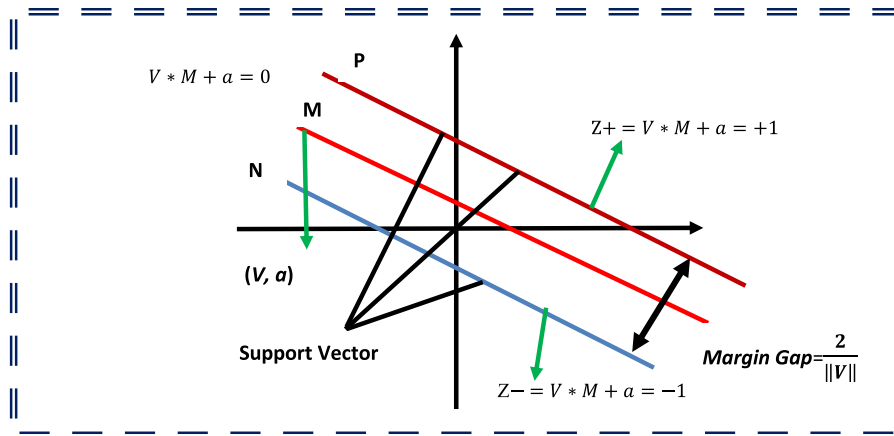


Fig. 3. SVM framework of proposed work.

- We conducted experiments using multiple kernel types (linear, RBF, and polynomial). We found that the **linear kernel consistently outperformed the others** in terms of classification accuracy and execution efficiency on our dataset.
- No non-linear patterns were observed in the data distribution that would necessitate non-linear kernel mapping.
- **Cross-Validation:**
 - A 5-fold cross-validation approach was employed to validate the selection of C and ensure that the linear kernel generalized well across unseen samples.

The selection of these parameters aligns to maximize the margin (as per Eq. (13): Margin gap = $2 / \|V\|$), which is central to the SVM's ability to generalize effectively.

The authors tried to solve *Issue 3* (the High Computational Costs of Security Mechanisms) by optimizing cryptographic processes using ML after selectively applying encryption-based sensitivity of data. Further, the SVM model classifies the digital image into two categories, where category 1 includes the portion of the image that will be encrypted, and category 2 consists of the portion of the image that will not be encrypted. The portion of the data or image containing more information will be encrypted. Therefore, it reduces the encryption time.

5.2. Implementation of chaotic map-based modified pulse-coupled spiking neuron circuit in HyMLCPP model

A system in which chaotic maps are used to alter or control the behavior of pulse-coupled spiking neurons is known as the Chaotic Map-Based Modified Pulse-Coupled Spiking Neuron Circuit (MPCSNC). The two main components of this idea are *chaotic maps*, which provide complexity and unpredictable factors to the behavior of the system, and *spiking neurons*, which imitate the activity of actual neurons. The complex and unpredictable system behavior is produced by combining pulse-coupled neural dynamics with chaos. It is advantageous for secure communication, computational modeling, and performing brain-like computation. MPCSNC is implemented in the HyMLCPP model due to the large key space and critical initial condition. It can prevent brute force attacks.

The chosen chaotic map of Fig. 3 is described by Eqs. (14)–(18).

$$f(i+1) = g(f(i)) = \begin{cases} g1(f(i)) \text{ mod } X & \text{for odd } i \\ g2(f(i)) \text{ mod } X & \text{for even } i \end{cases} \quad (14)$$

Where 'X' is a minimal fractional denominator.

$$g1(f(i)) = \left(\frac{-L_{12} \sin(2\pi s y(i)) + L_{11} \left(f_n(i) - \frac{1}{4} \right)}{k} \right) + f(i) \quad (15)$$

Where, $g1(f(i)) \ 0 \leq f_n(i) < \frac{1}{2}$

$$g1(f(i)) = \left(\frac{-L_{12} \sin(2\pi s y(i)) - L_{11} \left(f_n(i) - \frac{1}{4} \right)}{k} \right) + f(i) \quad (16)$$

Where, $g1(f(i)) \ \frac{1}{2} \leq f_n(i) < 1$

$$g2(f(i)) = \left(\frac{-L_{22} \sin(2\pi s y(i)) + L_{21} \left(f_n(i) - \frac{1}{4} \right)}{k} \right) + f(i) \quad (17)$$

Where, $g2(f(i)) \ 0 \leq f_n(i) < \frac{1}{2}$

$$g2(f(i)) = \left(\frac{-L_{22} \sin(2\pi s y(i)) - L_{21} \left(f_n(i) - \frac{1}{4} \right)}{k} \right) + f(i) \quad (18)$$

Where $\frac{1}{2} \leq f_n(i) < 1$

5.2.1. Experimental configuration for chaotic maps

The experimental configuration outlines the preparation and utilization of chaotic maps in this study. It involves specifying the initial values, control parameters, and iteration steps that generate the chaotic sequences. These settings are carefully chosen to highlight the sensitivity of the system to starting conditions and to ensure that the generated dynamics exhibit the required level of randomness and unpredictability. Such a configuration provides a consistent framework for evaluating performance and enables reliable comparison with existing methods.

The following are the Chaotic Maps parameters that were implemented in the proposed model:

Parameter 1: Initial Conditions (Initial Conditions are those that act as the primordial values for the chaotic map. The chaotic map, due to its sensitive nature, even a minute alteration in specified conditions results in completely discrete trajectories.)

Parameter 2: Control Parameter (To govern the transformative behaviour, appropriate governing parameters are designated within specified ranges.)

Parameter 3: Random Seed (The random seed is implemented by a chaotic map as one of the parameters. It ensures the reproducibility of experiments while maintaining randomness in the sequences.)

Parameter 4: Iteration (Here, the system has to pass several iteration steps, where the latest value is derived from the preceding one. This leads to the sequence that showcases high unpredictability and non-linearity.)

The system then undergoes multiple iteration steps, where each new value is derived from the previous one, leading to a sequence that demonstrates high unpredictability and non-linearity. This design guarantees stability in assessment while preserving the intrinsic randomness of chaotic systems.

Table 2 summarizes the parameters and initial conditions used to implement the MPCSNC chaotic map in the HyMLCPP framework. The coupling constants (L_{11} , L_{12} , L_{21} , L_{22}) regulate nonlinear interactions in the update equations, while k and s control normalization and sinusoidal scaling. The values of $f(0)$ and $y(0)$ serve as the initial states, with even small variations producing different chaotic trajectories. A fixed random seed (2025) ensures reproducibility of results during simulations.

Fig. 4 illustrates how the MPCSNC chaotic map reacts to small changes in its initial conditions. Three trajectories are plotted, each beginning with slightly different starting values: $\Delta f(0) = 10^{-3}$, 10^{-4} , and 10^{-5} . The vertical axis, shown on a logarithmic scale, represents the difference between paired trajectories, while the horizontal axis

Table 2

Chaotic map parameters and initial conditions.

Parameter / Condition	Value	Description
L_{11}	1.20	Coupling constant in Eqs. (15), (16)
L_{12}	0.90	Coupling constant in Eqs. (15), (16)
L_{21}	1.10	Coupling constant in Eqs. (17), (18)
L_{22}	0.80	Coupling constant in Eqs. (17), (18)
k	2.00	Normalization factor
s	0.50	Scaling factor in the sine function
X	1.00	Minimal fractional denominator
$f(0)$	0.630	Initial map state
$y(0)$	0.470	Initial neuron state
Random Seed	2025	Seed used for initialization reproducibility

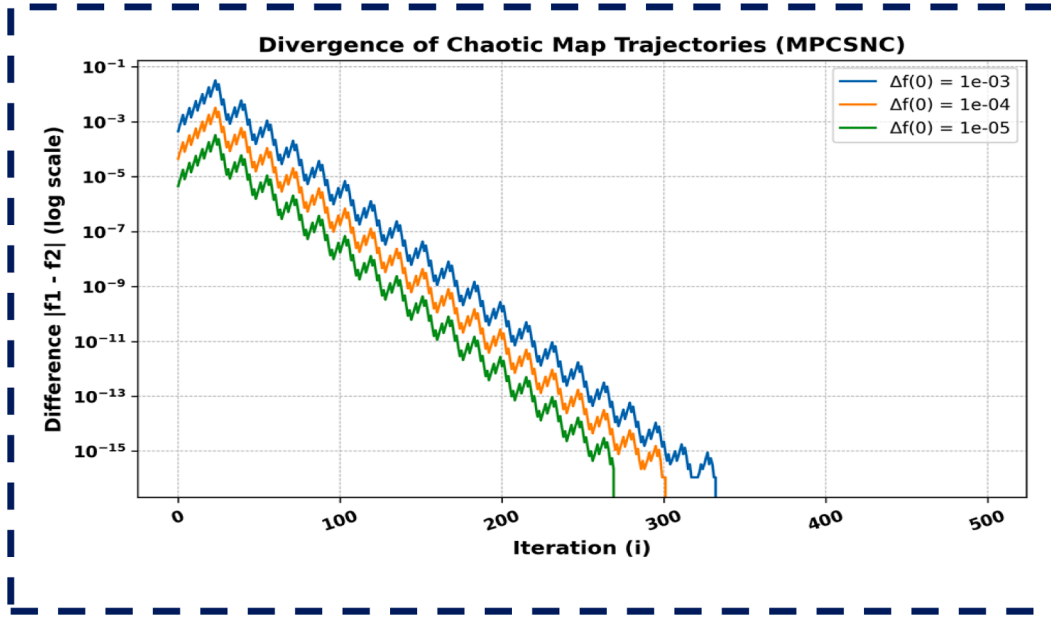


Fig. 4. . Sensitivity of MPCSN chaotic map to initial conditions.

indicates the number of iterations. The three colored lines correspond to different perturbation levels: the *blue line* represents $\Delta f(0) = 10^{-3}$, the *orange line* represents $\Delta f(0) = 10^{-4}$, and the *green line* represents $\Delta f(0) = 10^{-5}$. As the iterations progress, even tiny variations in the initial state grows rapidly, leading to noticeable divergence between trajectories. This demonstrates the sensitive dependence on initial conditions, which is a key property of chaotic systems. Such behavior increases unpredictability and ensures that the system cannot be easily reconstructed or predicted, making it highly useful for secure communication and cryptographic applications.

Chaotic maps offer strong potential in secure communication, but they demand precise synchronization between sender and receiver. Their extreme sensitivity to initial conditions and parameters means even minor mismatches can cause desynchronization and data loss. In practice, factors like channel noise, parameter drift, and environmental interference make stable synchronization difficult, posing a significant challenge for real-world use. To mitigate the synchronization challenge, HyMLCPP employs *Secure Key Initialization (SKI) techniques*, which are implemented in the proposed model. At the start of communication, chaotic map parameters such as initial states and control variables can be derived from a machine learning-assisted hash of the session key (e. g., generated through Diffie–Hellman or ECC). This removes the need for continuous synchronization during transmission.

5.2.2. Hybrid chaotic–ML framework for secure detection of distorted images in the cloud

The proposed HyMLCPP model is capable of detecting modified or distorted images due to its dual-layer validation. In the first layer,

chaotic maps generate highly sensitive hash values, where even minimal alterations in the image result in significant changes in the chaotic output. In the second layer, the ML module extracts deep feature embeddings that capture the structural and semantic content of the image. Any modification or distortion—such as tampering, noise injection, or geometric transformation—leads to a mismatch between the expected chaotic ML fingerprint and the received one. This dual validation mechanism enables HyMLCPP to not only secure the transmission but also provide reliable tamper and distortion detection.

Fig. 5 illustrates the detection of the distorted image in the cloud platform by the proposed model. The digital images are sent as the input to the model. The chaotic map generates a hash value, indicating that even small changes in the digital image can result in significant modifications to the output of the chaotic map. In the next stage, a machine learning model is deployed for feature extraction from the image, which extracts the vital features from the image. Both the chaotic map and the ML model are used to authenticate the correct or distorted image.

5.3. Advanced bit-plane extraction for image enhancement and encryption

Bit-plane (BP) extraction is a process in digital image processing that divides an image into multiple binary layers, called bit planes. In an 8-bit grayscale image, each pixel is composed of 8 bits. This method separates these bits into different planes, where each plane corresponds to a specific bit level for all pixels. The lower-order planes typically capture minor details, while the higher-order planes contain the most important visual information. This technique is commonly applied in areas like

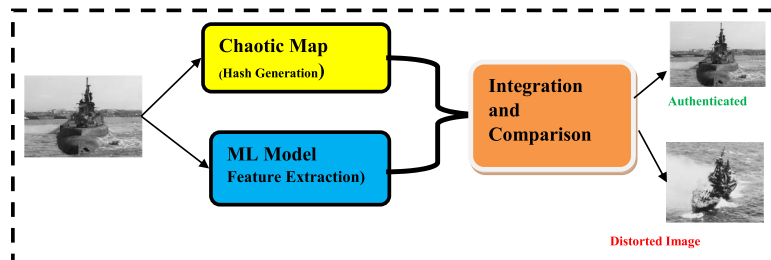


Fig. 5. Distorted image detection proposed model in cloud platforms.

image compression, steganography, and image enhancement. Further, it makes it easier to analyze and manipulate image data. In the proposed model, an 8-bit grayscale digital image is implemented. Fig. 4 shows the plain digital images and their related 8-bit plane ((7th)BP, (6th)BP,... (0th)BP). In the BP decomposition, the amount of information present in the several Bit Planes is calculated using Eq. (19).

$$(\text{BitPlane})_n \frac{2^{n-1}}{\sum_{i=0}^7 2^{i-1}} \quad (19)$$

It is understood from Fig. 6 that higher information is present in the 7th Bit Plane, and minimal information is present in the 0th Bit Plane. In the present work, only those bit planes are taken into consideration for encryption which provides adequate information in the plain digital image. Therefore, it reduces the computation time and data encryption time. Hence, Issue 3 (High Computational Costs of Security Mechanisms) of Table 1 is resolved.

This selective encryption approach ensures that critical data is secured without applying unnecessary operations on less informative regions. As a result, system efficiency is improved while maintaining the desired level of protection. The method also prevents resource wastage, making it suitable for large-scale cloud environments where processing power is a concern. Moreover, this strategy achieves a balance between strong security and faster performance, which is essential for real-time applications.

Fig. 7 represents the amount of information present in each plain text image. Here, Fig. 7(A) illustrates a column chart. From this chart, it is clear that the bit plane (7th) contains the maximum amount of information. The Bit Planes available from 7th BP to 4th BP include 85% to 89% of the information present in the digital image taken as input. So, it is required to encrypt the 7th to the 4th Bit-Planes. It is insignificant to encrypt the remaining bit planes because their encryption may consume more processing time. Fig. 7(B) is a table that shows pixel information density percentage. Fig. 7(B) shows that the maximum amount of information resides in the 7th block, which is approximately 73%. Therefore, encrypting the blocks containing a higher amount of information would reduce the computation time in the cloud sphere.

Fig. 8 is the algorithm that shows the step-by-step implementation of the encrypted image. HyMLCPP model can reduce the computation time and offer encryption of the original digital image by implementing the bit plane. Therefore, it reduces the computation time. The most informative block of the image is the 4th, 5th, 6th, and 7th blocks, and the same is encrypted during the process of data transmission. Hence, Issue 3

(High Computational Costs of Security Mechanisms), of Table 1 is fixed. By focusing only on the most significant blocks, the model avoids unnecessary encryption of less relevant data, leading to faster execution. This selective process also ensures that the security of critical information is maintained without compromising overall efficiency. The algorithm's design makes it scalable and suitable for handling large datasets in cloud environments. Furthermore, this approach provides an effective balance between performance optimization and strong data confidentiality.

5.4. Preserving privacy in the HyMLCPP model

The proposed model enables the data steward to keep the data safe and secure while preserving privacy in the cloud sphere. Since the data of the steward could be sensitive and confidential. Therefore, it is the responsibility of the cloud service provider to keep sensitive data protected from any leakage and damage. To achieve this milestone, the HyMLCPP model implements a unique two-way interface to secure sensitive data.

5.4.1. Preserving privacy using machine learning

Preserving privacy using the ML model includes standard approaches and methods to guarantee confidential data utilized in training and testing is not revealed. To achieve this, noise is introduced to the data, model parameters, or outcomes. Differential Privacy is a field of cryptography that is used to share information publicly about a data bank. It hides the contribution of individuals by maintaining the overall outcome while inserting random noise into the data. The ML model implements differential privacy for achieving privacy in the cloud sphere.

A concise step-by-step implementation of preserving privacy using differential privacy in the ML model is described below:

- i. The initial objective is to train the ML model 'M' to accomplish the task $f : x \rightarrow y$.

The task is achieved by preserving privacy without disclosing the identity of the individual in the data set $(D^1, D^2, D^3, \dots, D^n)$.

- i. The model is trained by using differential privacy for reducing the loss function $\mathbb{L}(M, D^n)$. Here, $\mathbb{L}(M(x^i), y^i)$ represents the loss between the model's prediction $M(x^i)$ for the input and the true label y^i .

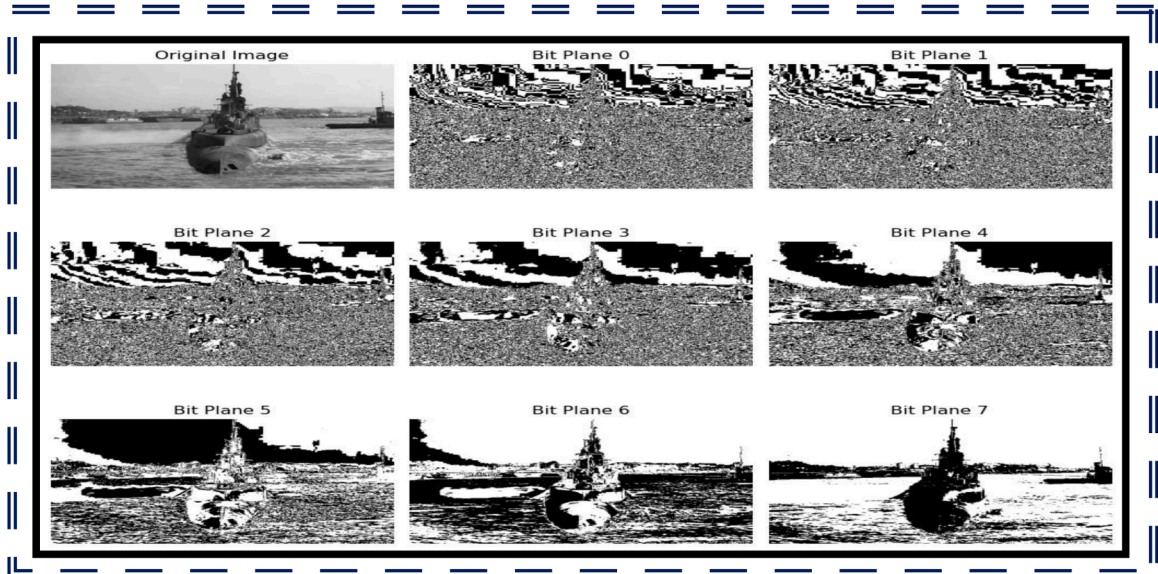


Fig. 6. Binary image decomposition.

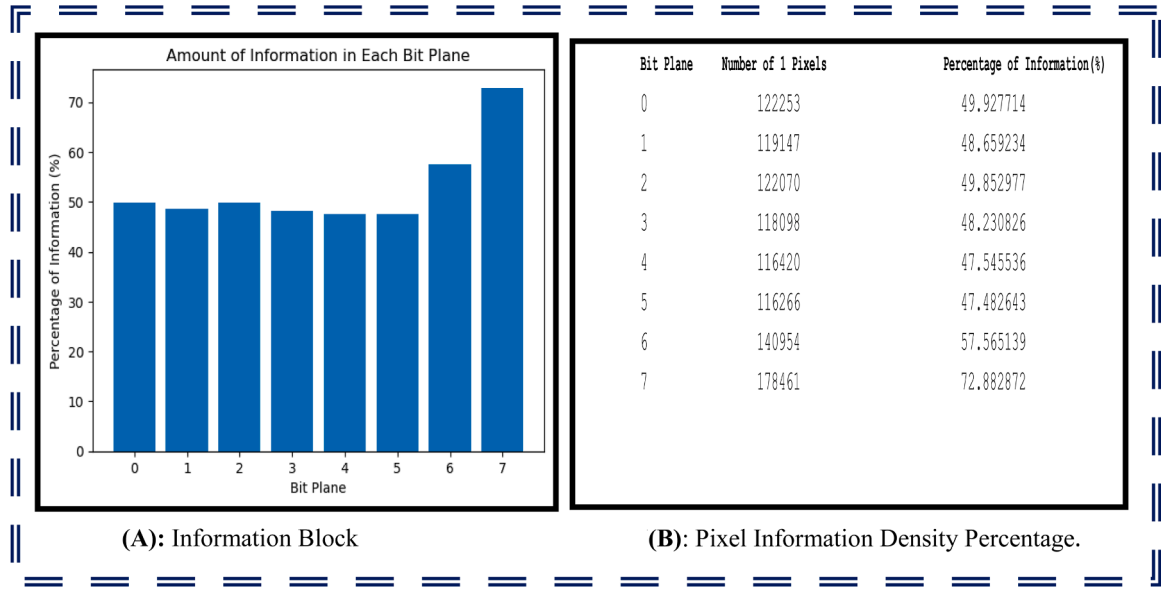


Fig. 7. . Information analysis of bit planes in digital images.

Input: Digital Image:

Output: An encrypted digital image with reduced computational time.

Algorithm 1: Encryption of Digital Image using Bit Plane Decomposition in HyMLCPP Model

Step 1: Include the library files cv2,numpy, and matplotlib.pyplot

Step 2: Input the Digital Image.

Step 3: Upload an image of 8-bit grayscale.

Step 4: Bit-Plane performs extraction by right-shifting the image bit and implementing the binary masking features.

```
def bit_plane_extraction(digitalimg, bit):
    return (img>> bit) & 1
```

Step 5: Print the encrypted Image.

Step 6: Finally, output the bit plane images on the output screen.

```
for i in range(8):
    bp = bit_plane_extraction(digitalimage, i) * 255
plt.subplot(3, 3, i + 2)
Step 7: End.
```

Fig. 8. Bit plane decomposition algorithm.

The standard symbol ' $\mathbb{L}$ ' is used to represent the loss or objective function that has to be reduced.

- i. Arrange the data set and divide it into two stages, where *stage 1* consists of the training data set and *stage 2* is used for testing the data. Further, it can be represented by Eq. (20).

$$D = \{D^{train} \cup D^{test}\} \quad (20)$$

- ii. Define the Privacy Parameter as follows:
- iii. Delta ' δ ': Probability of the privacy mechanism.
- iv. Epsilon ' ϵ ': Regulates privacy intensity; greater privacy is achieved at lower levels of ϵ .
- v. *Trimmed Data(TD)*: It ensures that each gradient update is bounded by preventing individual data points from

having a significant influence on the model. It can be represented by Eq. (21)

$$TD = \{TD^1, TD^2, TD^3 \dots TD^n\} \quad (21)$$

Continue5vi. Training the model with the help of differentially private stochastic gradient descent (DP-SGD), where every specimen of data calculates the gradient loss with respect to the model's parameters. It can be represented by Eq. (22).

$$G^i = \mathcal{G}^\theta \mathbb{L}(M(x^i, y^i)) \quad (22)$$

Whereas, $(x^i, y^i) \subset \{D^1, D^2, D^3, D^4, D^5\}$

Here, G^i shows the gradient loss function concerning the argument ' θ '. It indicates how many parameters of the model shall be updated to

reduce the loss. The function of loss ' $\mathcal{L}$ ' calculates the difference between the expected and actual labels. Standard loss functions include mean squared error (for regression tasks) and cross-entropy loss (for classification tasks).

Fig. 9 illustrates the approach employed by the HyMLCPP model to maintain privacy in the cloud environment. The system makes use of both differential privacy and ML approaches. Sensitive information is gathered from the data steward, and sample data is chosen for training. The method employs additional differential privacy techniques during inference to guarantee that predictions stay private beyond training. The gradient is computed after trimming the data, and the average value is applied by introducing noise to the data mean. In addition, the parameter is updated, and the gradient's value is applied to the model. This ensures that individual user contributions cannot be reverse-engineered, thereby minimizing the risk of data leakage. The combination of noise addition and gradient updates enhances model robustness while maintaining strong privacy guarantees. Furthermore, this approach supports compliance with data protection regulations, making it suitable for real-world cloud applications. Overall, the integration of ML with differential privacy establishes a secure and efficient framework for handling sensitive cloud data.

Differential privacy and ML techniques are used to address *Issue 5* listed in Table 1. The distribution of data among the different users is particularly crucial when it is transmitted over the cloud. This information can be used for a variety of studies, research projects, or pattern analysis. It is essential to preserve original data while making these allocations. When distribution occurs, the right procedure should be followed. This is done by adding noise to the data average. One possible form of this type of noise is a Gaussian distribution. It can be represented by Eq. (23).

$$\text{Gaussian noise} : q(D) + N(0, \sigma^2) \quad (23)$$

The complement procedure is demonstrated in Fig. 9, which provides an overview of how privacy can be accomplished and how data can be protected during distribution over the cloud. Here, *Mean (0)* ensures that the noise has zero bias value, and *Variance (σ^2)* is used to control the magnitude of the noise.

Privacy Parameter Settings:

- **Epsilon (ϵ):** $\epsilon = 3.0$
 - This value provides a **moderate privacy guarantee**, balancing privacy and model performance. A smaller value (e.g., <1) would result in stronger privacy but significantly degrade accuracy.
- **Delta (δ):** $\delta = 1e-5$
 - This was chosen based on standard recommendations where δ is set to be smaller than $1/n$, with n being the size of the dataset. This ensures a very low probability that any individual's data significantly affects the model output.
- These values were selected following common practice in privacy-preserving ML research, where ϵ values in the range of 1–10 and δ values in the range of $1e-5$ to $1e-8$ are considered acceptable, depending on the sensitivity of the data.
- In our case, we prioritized a balance between privacy protection and model utility, ensuring that the added noise (Gaussian mechanism, Eq. (23)) did not severely compromise classification performance.
- With these settings, the model maintained a classification accuracy of 94.3%, demonstrating the effectiveness of the approach under privacy constraints.

The Gaussian noise added during training (as shown in Eq. (23)) was used:

- Mean = 0
- Variance (σ^2) = 1.0 (tuned during model experimentation to control noise magnitude)

These parameters were calibrated to achieve $\epsilon = 3.0$ and $\delta = 1e-5$ based on the privacy accounting method outlined in the Moments Accountant technique.

5.5. Vulnerability detection in the HyMLCPP model

Vulnerability is a defect or weakness within a system, network, or application that malicious actors can exploit to gain unauthorized

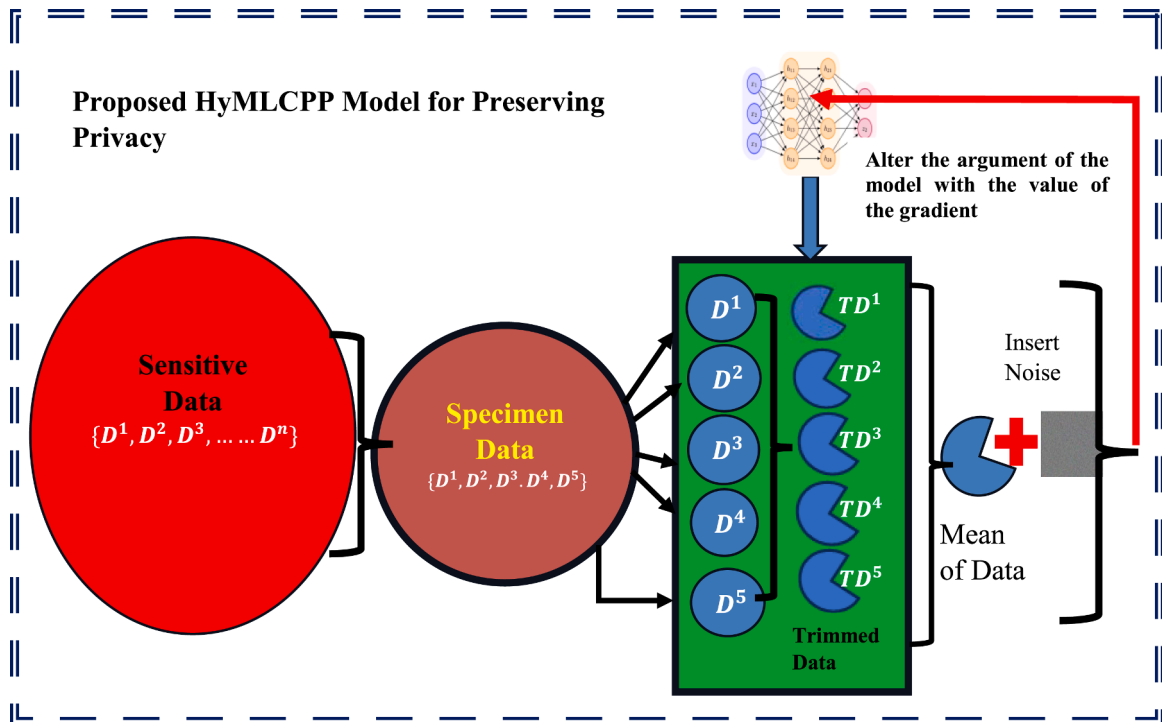


Fig. 9. Preserving privacy using an ML model.

access, manipulate data, or disrupt operations. The complexity and scale of interconnected Cloud and IoT technologies give rise to vulnerabilities in data security and privacy. Cloud computing may lead to data breaches due to configuration errors, weak access management, or insecure APIs. Additionally, IoT devices present further risks, such as limited security protocols, poor encryption, vulnerability to device misuse, and many more. Sensitive data stored or transmitted in the cloud is vulnerable to unauthorized access and may be altered when encryption is insufficient. Various types of vulnerability detection methods and standard detection tools used in cloud-IoT systems are described in Table 3.

Table 3 highlights the various vulnerability categories and the accompanying vulnerability detection techniques. The vulnerabilities are categorized according to their type (software, network, or system-based), and particular frameworks or tools for detection are listed. To properly address and mitigate different security issues, Table 3 can be used as a reference to select the right technologies.

The detection of a DDoS attack means finding whether a DDoS attack has occurred or not. The identification of a DDoS attack means tracing the specific attack that is responsible for data modification or data leakage in a cloud IoT environment. The HyMLCPP model implements a threshold-based approach for detecting DDoS attacks and a Random forest-based approach for identifying specific DDoS attacks. One of the vulnerabilities that cloud and IoT systems face when they are integrated. They struggle to establish the connection during the data transmission; as a result, data are either unsent within a particular time frame or sent slowly. One of the root causes of this is a Distributed Denial of Service (DDoS) attack. The DDoS attack aims to overwhelm a target's network, server, or application by sending an excessive quantity of traffic from numerous sources at once. DDoS attacks are often deployed to interrupt services, inflict financial losses, harm an organization's reputation, or act as a distraction for additional malicious actions. The DDoS assaults can be challenging to counter because of their scattered nature. Many

Table 3
Various vulnerability and detection tools to detect vulnerability.

S. No.	Vulnerability Detection Methods	Description	Common Tools
1.	Static Analysis	Analyzes source code or binaries without execution to identify vulnerabilities based on coding standards. Suitable for early detection during the development phase.	SonarQube, Checkmarx, Veracode
2.	Dynamic Analysis	Tests applications during execution, simulating attacks to uncover runtime vulnerabilities. Essential for identifying vulnerabilities in running applications.	OWASP ZAP, Burp Suite, Acunetix
3.	Interactive Application Security Testing (IAST)	Combines static and dynamic analysis using agents to provide real-time feedback on vulnerabilities. Provides a holistic view by combining static and dynamic approaches	Contrast Security, Seeker by Synopsys
4.	Network Vulnerability Scanning	Scans networked systems for known vulnerabilities using a database of common exploits and weaknesses. Crucial for identifying network-level vulnerabilities.	Nessus, Qualys, OpenVAS
5.	Web Application Scanning	Specializes in identifying vulnerabilities in web applications, such as SQL injection and XSS attacks. Focuses specifically on web application security.	Nikto, Arachni, Netsparker

researchers have proposed techniques to safeguard from DDoS attacks to some extent. It can be identified, but due to the scattered nature of DDoS attacks, these approaches fail in some situations. This may be because the researchers neglect some of the steps that are important in identifying the attack. Therefore, a robust technique is required to identify these DDoS attacks in the Cloud-IoT Environment. The solution is provided by the authors in the current research study through the HyMLCPP model, which utilizes machine learning techniques to identify DDoS attacks. The proposed HyMLCPP model is implemented using a two-way attack detection method in a cloud-IoT environment.

Method 1: Threshold Approaches to Detect DDoS Attacks.

Method 2: Random Forest-Based ML Approaches to Detect DDoS Attacks.

Table 4 illustrates a comparison of the Random Forest and Threshold Method for the HyMLCPP model-based DDoS attack detection. Essential parameters are assessed, including complexity, computing efficiency, performance, interpretation, and use case. Although the Threshold Method may provide faster detection with less processing demands, the comparison shows how Random Forest performs better in terms of accuracy and robustness.

5.5.1. Threshold method to detect DDoS attack (Method 1)

The threshold method is the traditional method to identify the attack. This method sets a limit i.e., a threshold at a particular point. Setting predetermined criteria for specific metrics, like packet rates, connection requests, or bandwidth utilization, and keeping an eye on network traffic are all part of it. The DDoS attack may be suspected when these metrics show unusual activity that is above the set thresholds. This threshold approach is effective because DDoS attacks often result in sudden and significant deviations from normal network behavior. Key characteristics of threshold-based approaches for detecting DDoS attacks include *Simplicity, Real-Time Detection, Minimal Computational Overhead, and Dependency on Threshold*

Fig. 10 is the flowchart diagram that shows the process involved in vulnerability detection of DDoS attacks in a cloud environment. The process begins by collecting traffic data at regular intervals. After collecting data, the threshold value is fixed for the input traffic. Thereafter, the traffic analysis is done. If the traffic is found to exceed the threshold value, then "DDoS Attack is Identified"; otherwise, "No DDoS Attack". This mechanism enables early detection of abnormal traffic patterns, which helps in preventing service disruption. By continuously monitoring network behavior, the system can adapt to dynamic attack strategies. The use of threshold-based detection also ensures quick decision-making with minimal computational overhead. Moreover, the flowchart highlights a simple yet effective defense strategy that can be integrated with advanced models for real-time cloud protection.

Fig. 11 describes the proposed HyMLCPP model's algorithm for

Table 4
Comparative study of threshold method and random forest for DDoS attack detection used in HyMLCPP model.

Aspect	Threshold-Based Methods	Random Forest
Overview	Makes decisions by setting a cut-off.	Ensemble model combining multiple decision trees.
Level of Complexity	Low	High
Performance	Effective for simpler patterns	Excellent for capturing complex relationships.
Ease of Interpretation	High	Moderate
Handling Imbalanced Data	Requires careful tuning of the threshold	Manages well, especially with class weighting or sampling.
Typical Use Cases	Suitable for basic classification tasks	Ideal for complex classification and regression problems.

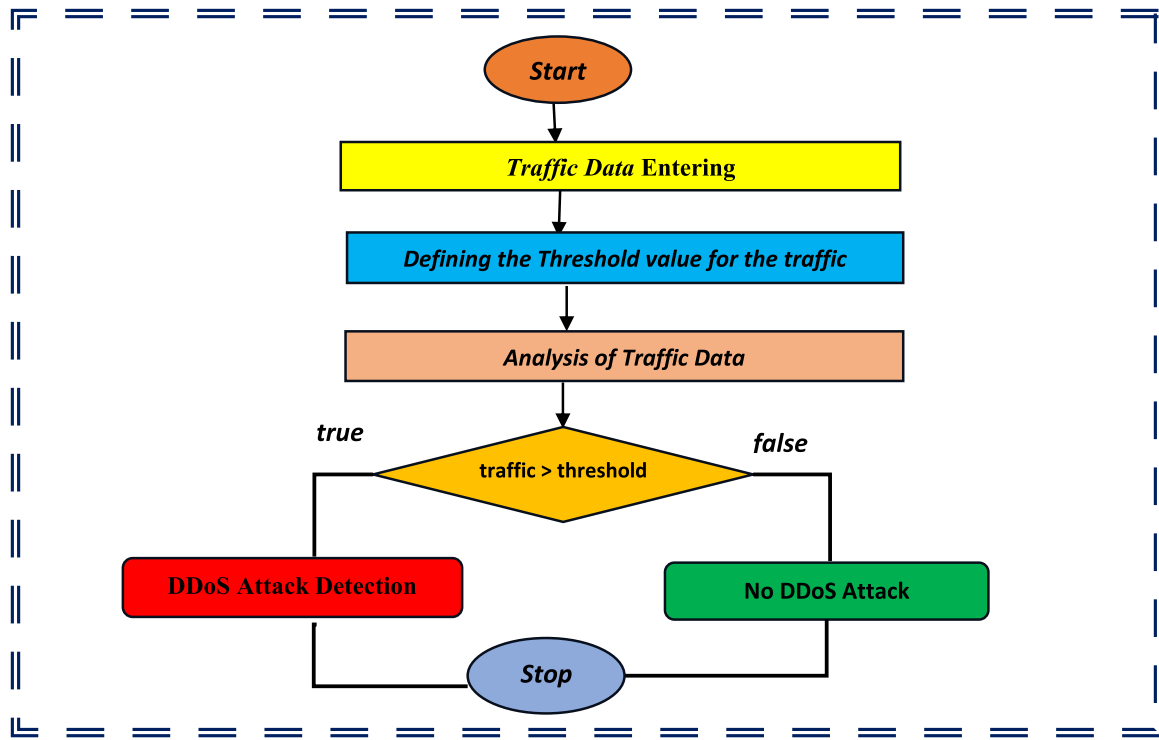


Fig. 10. Flow diagram of DDoS detection in traffic data.

Input: Traffic Data

Output: Classification of Normal traffic and DDoS Attack.

Algorithm 2: Algorithm to detect Vulnerability Detection (DDoS) attacks in the HyMLCPP model using the Threshold Technique.

Step 1: Generate data from network traffic by creating a simulation.

Step 1.1: Replicate typical traffic statistics.

Step 1.2: Categorize the traffic of DDoS attacks.

Step 2: Integrate assault traffic with regular traffic.

Step 3: Establish a threshold for detecting DDoS (Example: *threshold* =150).

Step 4: Determine whether traffic crosses the threshold.

Step 4.1: **if** (*traffic*>*threshold*)

print ("DDoS Attack Detection in Traffic Data");

else

print ("No DDoS attack detected.");

Step 5: Display the outcome.

Step 6: END.

Fig. 11. HyMLCPP model's vulnerability detection algorithm for DDoS attacks through the Threshold technique.

detecting DDoS attacks using the Threshold technique. The proposed algorithm works by establishing preset criteria for network traffic indicators, including bandwidth use, packet rate, and flow duration. Abnormalities are detected when these metrics exceed the thresholds, and it may be a signal of potential DDoS activity. The approach is effective and lightweight, which makes it appropriate for real-time detection. However, to reduce false positives and false negatives, it significantly depends on precise threshold setting. The algorithm is integrated with adaptive learning methods that automatically adjust thresholds based on evolving traffic patterns to ensure reliability. This allows the system to remain resilient against both traditional and emerging attack strategies. Additionally, its low computational demand makes it highly suitable for cloud environments where scalability is essential. Overall, the method provides a practical balance between detection accuracy and

efficiency.

Fig. 12 represents one of the vulnerabilities that occur in a cloud-IoT environment when data transmission takes place. The Python-based technique is implemented in the proposed HyMLCPP model for DDoS attack detection that looks for irregularities through traffic monitoring. This technique looks for potential assaults by comparing traffic patterns with predetermined thresholds. This vulnerability is a DDoS attack. The graph of Fig. 11 is generated from the incoming traffic data. The X-axis denotes Time, and the Y-axis denotes Traffic Volume at the particular time frame.

The traffic volumes on the left side of the graph in Fig. 12 represent regular traffic. It is comparatively steady and far below the threshold, which indicates regular cloud network behavior. The threshold is fixed at 150 traffic volume after conducting a regression analysis. The Attack

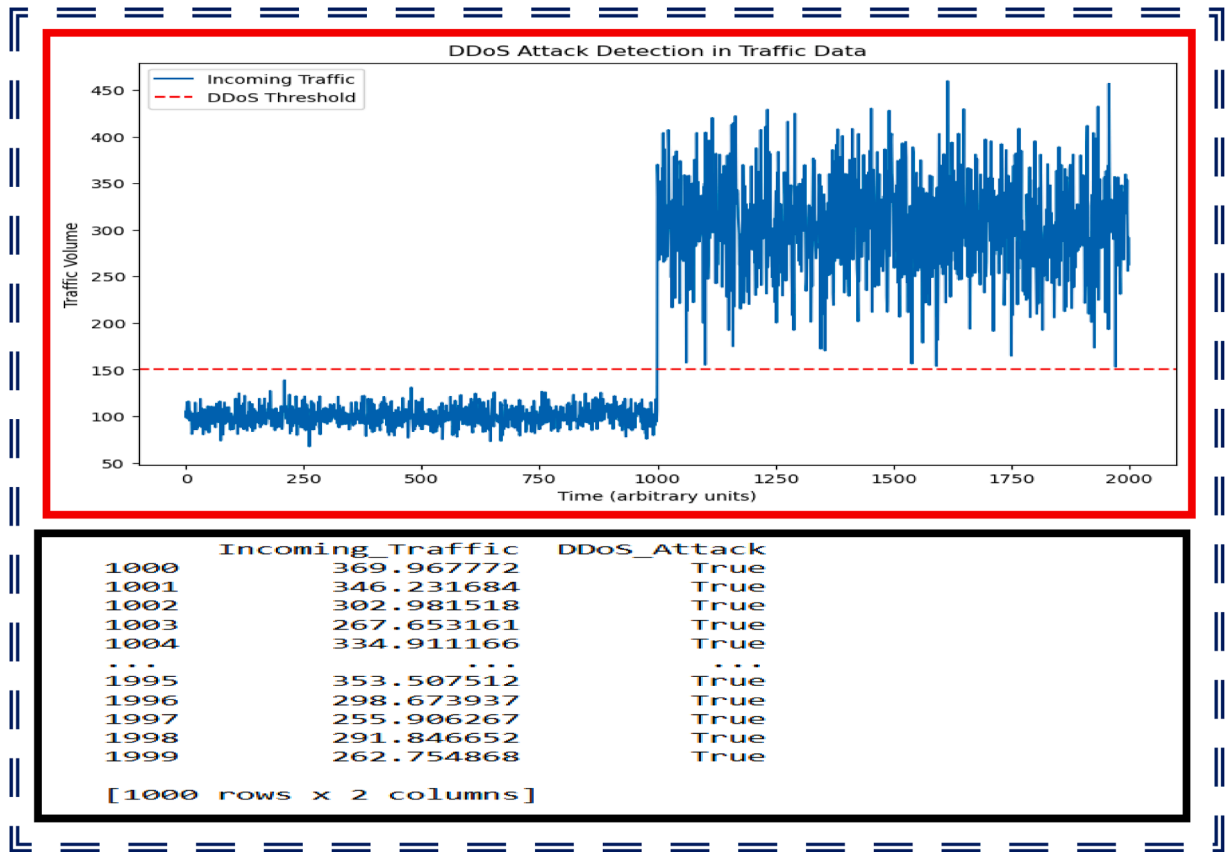


Fig. 12. Vulnerability detection: DDoS attack detection in traffic data by simulating network traffic data.

Traffic (Graph on the right) indicates the sudden increase in traffic signal. It means that a DDoS attack is about to start. The system flags the traffic as an assault because it surpasses the pre-determined threshold. If the traffic is beyond the stated value, it signifies that there is a DDoS attack. Therefore, the proposed HyMLCPP model can identify the potential time of a DDoS attack by rapidly analyzing when traffic crosses the threshold.

5.5.2. HyMLCPP model's algorithm for detecting DDoS attacks using the Random Forest algorithm (Method 2)

The proposed HyMLCPP is a sophisticated framework that combines ML and cryptography to improve the detection and defense against DDoS attacks in the current networks of cloud-IoT systems. The HyMLCPP's Random Forest component processes many variables, including packet speeds, flow durations, and byte counts, to find anomalies in traffic data. In machine learning, the Random Forest approach is a popular ensemble algorithm for identifying DDoS attacks because of its versatility, precision, and resilience. It builds several decision trees during training and combines their results for categorizing network traffic as benign or malicious. Because of these features, Random Forest is a dependable and efficient technique used for identifying DDoS attacks in a dynamic and high-volume network setting.

Key characteristics of Random Forest approaches for detecting DDoS attacks are as follows:

- Multimodal Learning:** Random Forest increases classification accuracy and noise resistance by combining the predictions of several decision trees.
- Feature Importance:** It can rearrange the characteristics according to their significance when making choices. Hence, it aids in locating crucial DDoS attack indications (such as packet rate and flow duration).

- Tolerance to Missing Data:** It uses substitute splits or average outputs across trees to deal with missing values.
- Probabilistic Results:** It enables adaptable decision-making based on degrees of confidence by providing forecast probabilities.
- Handles High-Dimensional Data:** It works well with big datasets that include a lot of features.

The crucial step involved in detecting Vulnerabilities of DDoS attacks in the HyMLCPP model is described in Table 5.

Table 5 describes the actions that must be followed to identify DDoS assaults after applying the HyMLCPP model. It covers every step, including feature selection, data pre-processing, model training, and evaluation. The proposed HyMLCPP model is integrated with various components and functions to examine traffic patterns. The precise detection of possible DDoS threats is also highlighted in Table 5.

Fig. 13 portrays DDoS detection by the proposed model, which helps in detecting whether an attack has occurred or not. This process starts with data collection from various data repositories. The data is then processed, and the required features are extracted from the data set. After the extraction of features, the data is sent to the Random Forest classification model. It is an ML technique used for the classification of multiple classes. The proposed model implements Random Forest to distinguish between normal and attack traffic. Create warnings automatically using the proposed HyMLCPP model when unusual traffic levels are above pre-determined thresholds (150). This warning could be a sign of a DDoS attack. The integration of feature extraction with Random Forest improves the accuracy of detection by capturing both linear and non-linear traffic behaviors. The model also minimizes the chances of misclassification by leveraging ensemble learning. Furthermore, the automated alert system enhances real-time responsiveness, allowing for quick countermeasures against ongoing threats. Overall, the proposed approach ensures both efficiency and robustness in

Table 5
Steps involved in detecting DDoS attack and HyMLCPP model application.

S. No	Steps	Key Step	Implementation	HyMLCPP Model Application
1.	Step 1	Data Collection	Gather data from IoT devices and cloud systems, focusing on logs, network traffic, and API requests.	Utilize data from network devices to monitor traffic patterns and identify anomalies in access logs. Example: Data set from Kaggle. dataset_sdn.csv
2.	Step 2	Machine Learning for Anomaly Detection	Apply ML algorithms to analyze the gathered data for potential vulnerabilities.	Implement algorithms like <i>Random Forest</i> to distinguish between normal and attack traffic.
3.	Step 3	Securing Communications and Integrity Verification	Ensure encryption of communications between IoT devices and cloud infrastructure. Use digital signatures to validate the integrity of data and configurations. It may help to prevent tampering.	Establish secure channels using TLS/SSL protocols for communication between devices and servers.
4.	Step 4	Risk Analysis with Probabilistic Models	Use probabilistic models to assess the likelihood of a DDoS attack occurrence. It is based on historical and real-time data.	Implement Bayesian networks to estimate the probability of a DDoS attack based on observed traffic anomalies and device behaviors.
5.	Step 5	Continuous Monitoring, Alerts, and Model Updates	Set up a real-time monitoring system to track system performance and identify signs of a DDoS attack.	Utilize the HyMLCPP model to automatically trigger alerts when abnormal traffic volumes exceed defined thresholds of 150, indicating a potential DDoS attack.

managing cloud-based DDoS vulnerabilities.

Fig. 14 is an algorithm representation of the proposed vulnerability detection algorithm for DDoS attacks using the Random Forest approach. The Random Forest approach yields the best outcome in categorizing whether an event is an “Attack” or “No Attack”. The process starts with uploading the dataset, followed by feature selection in the dataset. Next, it divides the dataset into training and testing. Further, it implements a random forest approach to classify the traffic. The algorithm works by constructing multiple decision trees, each trained on different subsets of the data, to avoid overfitting and enhance prediction accuracy. The outputs from all trees are combined using majority voting to decide the final classification. This ensemble method improves robustness, as even if some trees misclassify, the overall result remains reliable. Additionally, the algorithm efficiently handles large-scale traffic data, making it suitable for real-time cloud security applications. Random Forest was specifically chosen because of its ability to process high-dimensional features, scalability in distributed environments, and superior accuracy compared to single classifier methods.

The following is a detailed description of the experimental setup:

- **DDoS Simulation:** The proposed model was tested on standard datasets such as the *CICIDS 2017 Dataset*, *Kaggle Dataset*, *AIND DDoS Dataset*, and *CTU-13 Dataset* for DDoS attack detection. The proposed

model implements *Random Forest* to distinguish between normal and attack traffic.

- **Duration of Scenarios:** Each scenario ran for 5 min with 1-minute idle gaps, repeated five times per profile to ensure sufficient traffic volume for analysis.
- **Number of Samples:** Benign samples: 63,561, Attack samples: 40,784, Total: 104,345

In Fig. 15, the confusion matrix for the dataset tested under various training and testing conditions is shown. The confusion matrix is an ML performance metric for categorization issues. To assess a model's performance, it contrasts the expected and actual classifications. Fig. 15(A) represents the visualization of data with 15% of Training and 85% of Testing. The top-left represents True Positive (TP), which means correctly predicted positive values, and its value is 53,720. The top-right represents False Negative (FN), which means missed positives, and it is 22. The bottom-left represents False Positive (FP), which means incorrectly predicted positives, and its value is 52. The bottom-right represents True Negative (TN), which correctly predicted negatives, and its value is 34,470. It performs well at identifying both positive and negative classes, with very few misclassifications. The proposed model is highly reliable, as it yields high values for all metrics. The accuracy of the proposed model for Fig. 15(A) is 99.92%. It means that almost all the cases were correctly classified. Fig. 15(B) represents the interpretation of data with 30% of Training and 70% of Testing. It indicates an excellent balance between identifying positives and negatives. False negatives (missed positives) and false positives (incorrectly classified negatives) are almost negligible. The confusion matrix of Fig. 15(B) shows that the model handles class imbalance very well. The high number of TP and TN versus the negligible number of FP and FN suggests strong performance in both classes. In Fig. 15(B), the value of TP is 44,158. It means that the majority of positive cases were correctly classified. Further, the value of TN is 28,527. It means that most of the negative cases were also correctly classified. Furthermore, the values of FN = 1, and FP = 2 mean that the model made only three total misclassifications across both classes. It confirms its high reliability.

Fig. 15(C) represents the interpretation of data with 40% of training and 60% of Testing. Here, the value of TP is 37,900. It means that, in these cases, the proposed model correctly predicts the positive class. In Fig. 15(c), the value of FN is 4. These are the cases where the model incorrectly predicted the negative class for actual positives. In the same case, the value of FP is 3. Therefore, in 3 cases, the model incorrectly predicted the positive class for actual negatives. Here, the value of TN is 24,397. It means the model correctly predicted the negative class in these cases. The calculated Accuracy for Fig. 15(C) is 99.99% which shows the overall percentage of correct predictions. In Fig. 15(D), the authors used Training data =50%, and Testing data =50%. It can be visualized through the confusion matrix. Here, the value of TP is 31,652. It means the positive cases were correctly predicted as positive in 31,652 cases.

In Fig. 15(D), the calculated value of FN is 4. It indicates the situations where positive cases are incorrectly predicted as negative. Further, the value of FP = 0 suggests that no negative case is incorrectly predicted as positive. Hence, the model perfectly identifies all negatives, avoiding unnecessary flags or disruptions. The value of TN = 20,264 indicates the negative cases that were correctly predicted as negative. The model's performance is nearly flawless, and its deployment would provide reliable and consistent results for the given classification tasks. The analysis of the confusion matrix and performance metrics demonstrates that the model exhibits exceptional classification performance.

The heatmap of Fig. 16 (Fig. 16 (A) and Fig. 16(B)) presents correlations among various features in the dataset. It helps understand the relationships and their potential impact on the target variable (label).

In case 1 of Fig. 16(A), if Training is 15%, and Testing is 85%, then the Red color is close to 1. It indicates that the features with a strong positive correlation move in the same direction. Example: If *pktscount* increases,

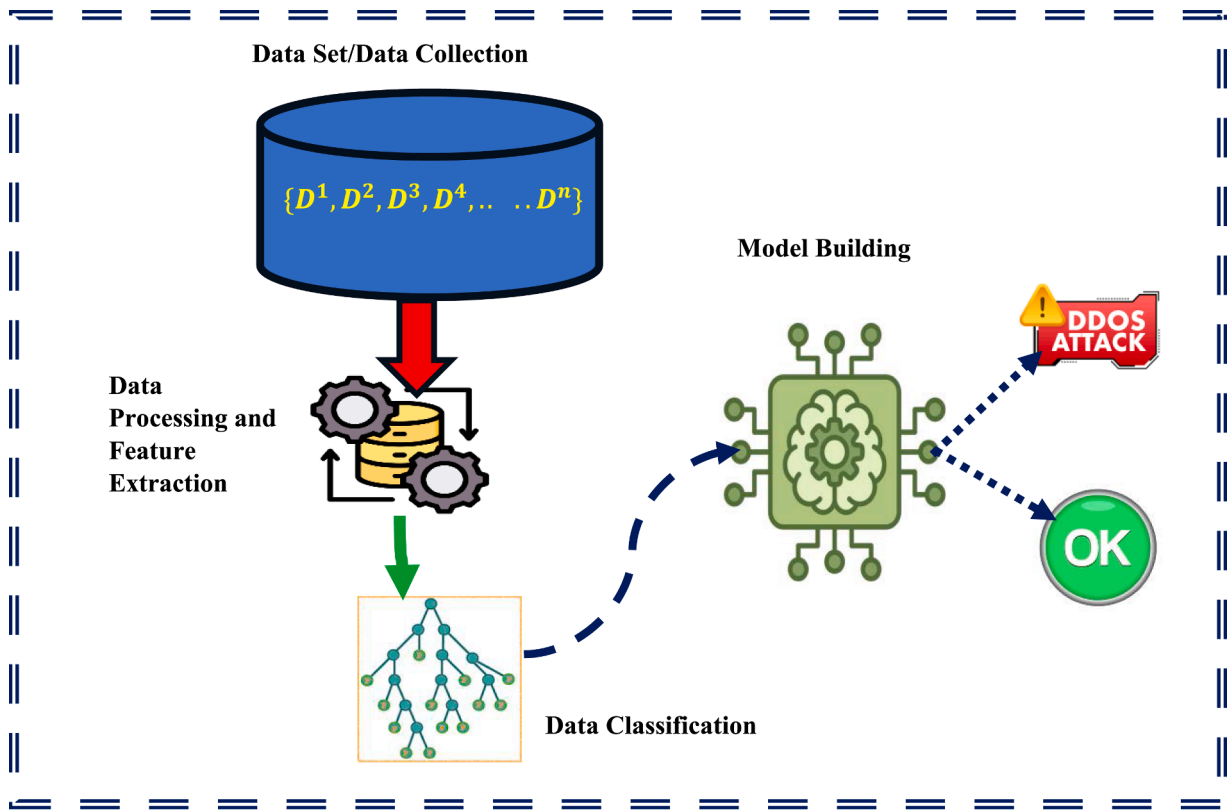


Fig. 13. DDoS detection framework architecture.

Input: Data Set

Output: Identification of DDoS attacks correctly.

Algorithm 3: DDoS Attack Vulnerability Identification in Traffic Data using Random Forest.

Step 1: Upload Data set (C:/Users/MAYANK/ProjectVehical/archive (3)/dataset_sdn.csv)

Step 2: Divide the dataset into labels ('Y') and features ('X').

Step 3: Create training and testing sets from the dataset.

Step 4: Set up the Random Forest classification system.

Step 5: Train the model.

Step 6: Predict the test set.

Step 7: Examine the model.

Step 8: END

. 14. Vulnerability detection algorithm for DDoS attacks through the Random Forest algorithm in HyMLCPP.

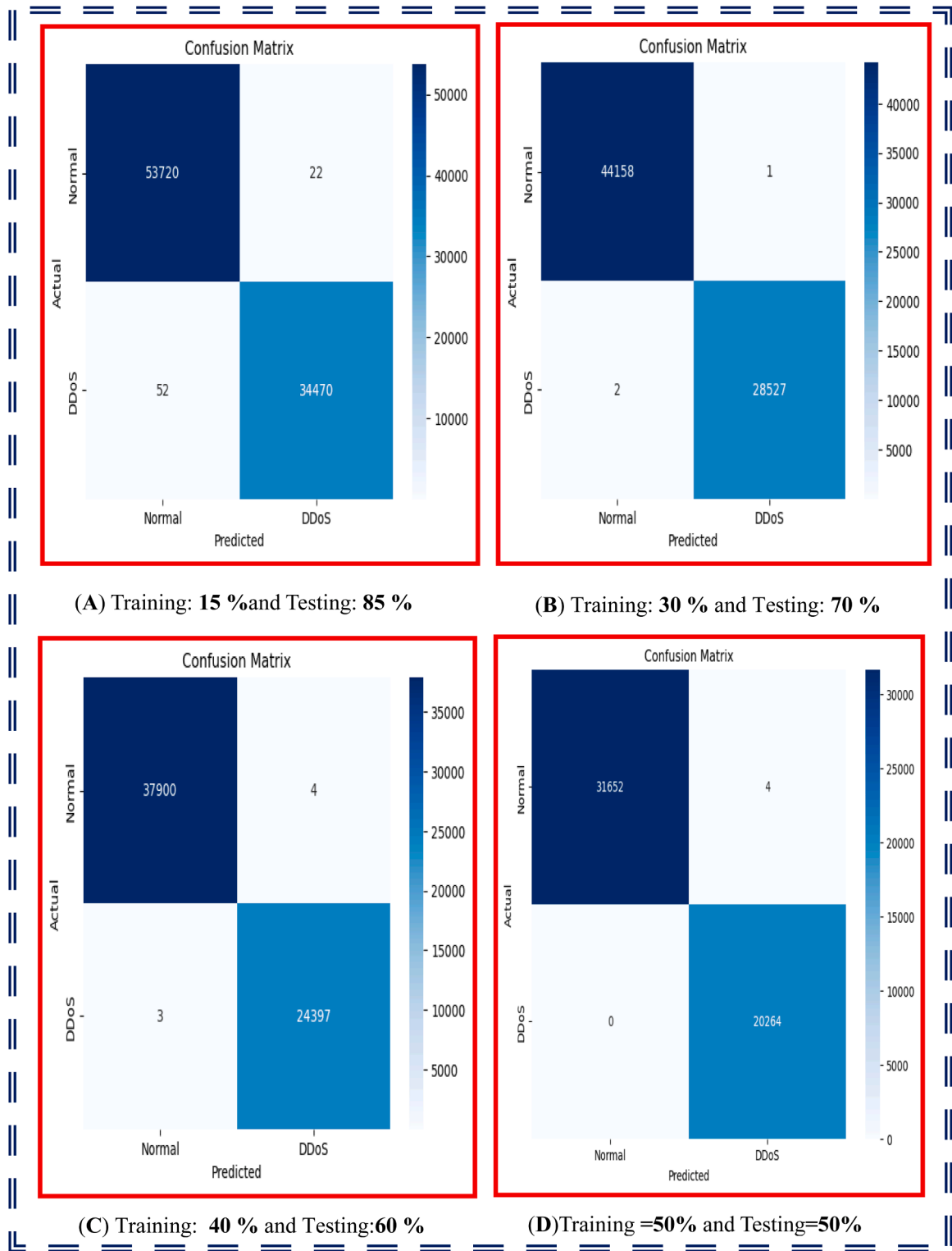
then the byte count also increases. The attributes *account*, and *byte count* (value 0.68) indicate a strong dependency between the number of packets, and the data volume (bytes transmitted). Further, the Blue color, which is close to -1, indicates that a strong negative correlation moves in opposite directions. Example: If *port_no* increases, then the probability of the label being positive decreases. The *port_no* and label (-0.40) indicate that specific port numbers are inversely related to the target (label). It may indicate the fault patterns in specific ports. The white/Neutral color (almost 0) indicates that the features have a weak or no linear relationship. Example: *dur_nsec* has virtually no relationship with the label. Features such as *switch*, *rx_bytes*, and *dur_nsec* show minimal correlation with labels, suggesting that they may contribute a little to the classification task.

In Case 2 of Fig. 16(A), the Training is 30%, and Testing is 70%. The heatmap of Fig. 16(A) displays the correlation between various properties with their values ranging from -1 to 1. The strong positive

correlations are indicated by red hues when two traits rise together. Blue colors indicate the weak or negative correlations. The strong relationship between *account* and *byte count*, for instance, is seen by their correlation value of 0.68. Likewise, strong correlations exist between *byteperflow* and *pktperflow*(0.81), and *tx_kbps* and *tot_kbps*(0.76). It indicates that the information is overlapping.

In Case 1 of Fig. 16(B), the Training is 40%, and the Testing is 60%. The heatmap of Fig. 16(B) displays the correlation between the various attributes with values ranging from -1 to 1. A high negative correlation is shown by a number near -1 (blue), whereas a strong positive correlation is indicated by a value near 1 (red), implying two traits increase together.

Here, a little or no association is indicated by a correlation value near 0 (white). For example, the correlation between *pktcount* and *bytecount* is 0.68, indicating a strong positive relationship between them. Likewise, there is a substantial strong correlation between *byteperflow* and

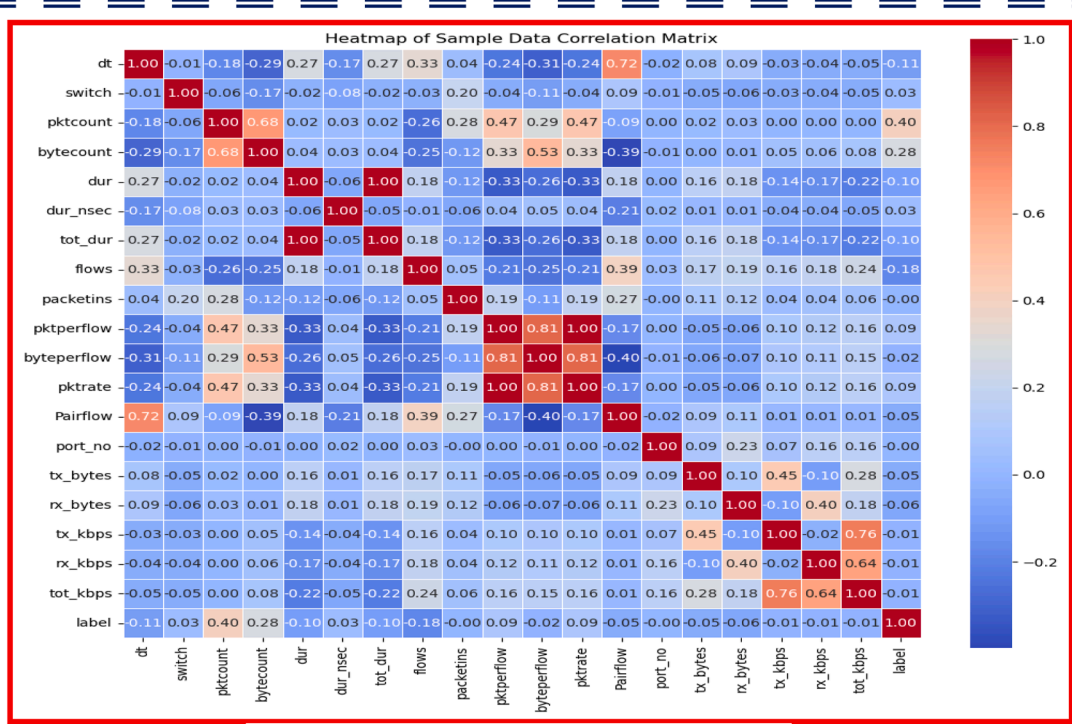
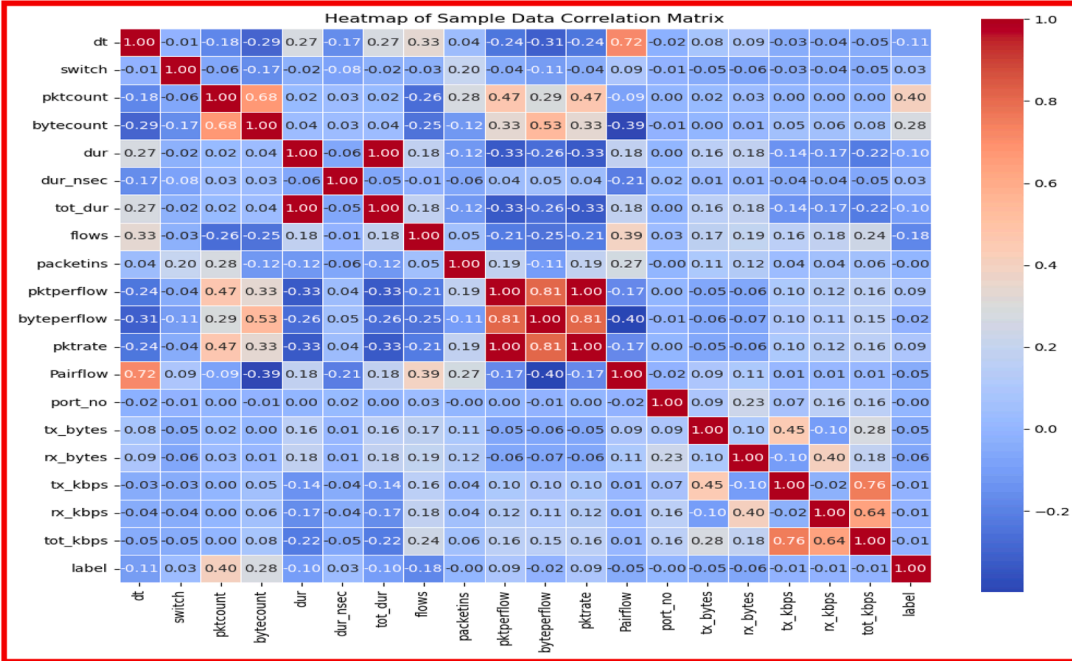


15. Confusion matrix of DDoS attack detection on dataset.

pktpflow (0.81), and *tx_kbps* and *tot_kbps* (0.76). Similar results were also obtained in case 2 of Fig. 16(B).

In Fig. 17(A), the true positive (TP) rate is displayed on the graph for each training phase. The TP rate initially peaks at about 60.85%, but at 30 training steps, it drops to about 60.75%, suggesting a reduction in performance. After that, it recovers a lot, reaching about 60.95% by 50 training steps, indicating that the model improves its performance and correctly classifies positives in later training phases. In Fig. 17(B), the

true negative (TN) rate for each training step is depicted in the graph. The TN rate first rises before reaching its maximum at about 30 training steps (~39.25%), suggesting better performance in accuracy while identifying negatives. In Fig. 17(C), the graph shows a significant drop in the false positive (FP) rate as training steps increase, with the rate reducing from 0.05% (at 15 training steps) to near 0% (at 30 training steps). Beyond 30 training steps, the FP rate stabilizes, indicating that the model has effectively learned to minimize misclassifications. This

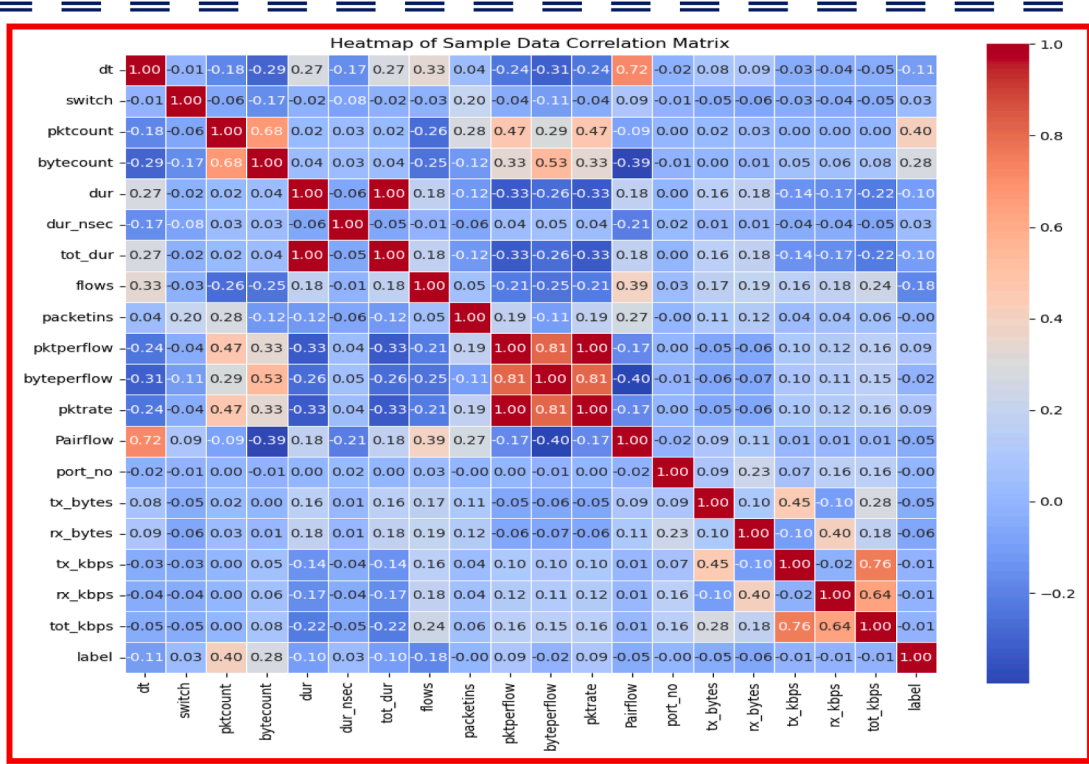
**Case 1: Training: 15 % Testing: 85 %****Case 2: Training: 30 % Testing: 70 %****16(A). Heat map of DDoS attack detection.**

experiment suggests that 30 steps could be an optimal point for balancing performance and computational efficiency. The near-zero FP rate demonstrates the model's high reliability in detecting anomalies accurately.

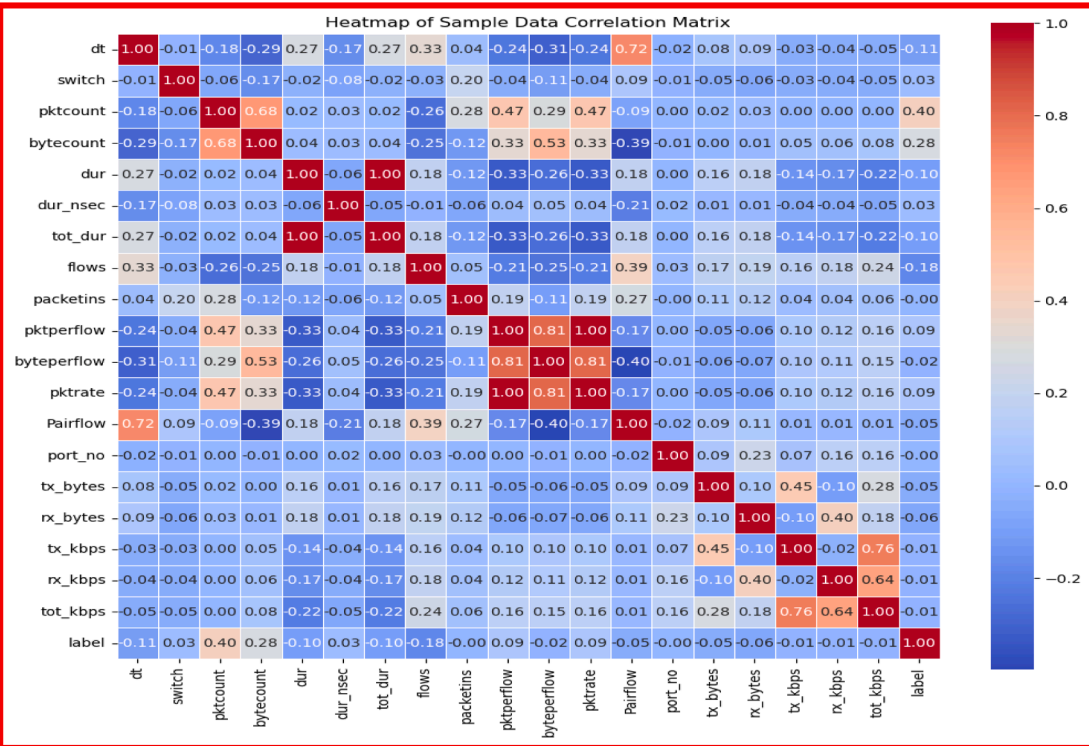
In Fig. 17(D), the graph demonstrates a notable improvement in anomaly identification during this phase, with the false negative (FN) rate dropping from 0.025% (at 15 training steps) to almost 0% (at 30 training steps). The FN rate gradually rises after 30 steps and levels off at

0.005%. The results suggest that 30 training steps may be the ideal number to strike a balance between computational efficiency and detection accuracy. The model is an example of successful learning.

In Fig. 17(E), the graph shows that the accuracy steadily increased from 99.92% (at 15 training steps) to almost 100% (at 30 steps), suggesting substantial learning improvement. The model appears to have hit its peak performance and may experience diminishing returns after 40 steps because the accuracy stabilizes after 30 steps. Therefore, the



Case 1: Training: 40 % Testing: 60 %



Case 2: Training Data=50% and Testing Data =50%

. 16(B). . Heat map of DDoS attack detection.

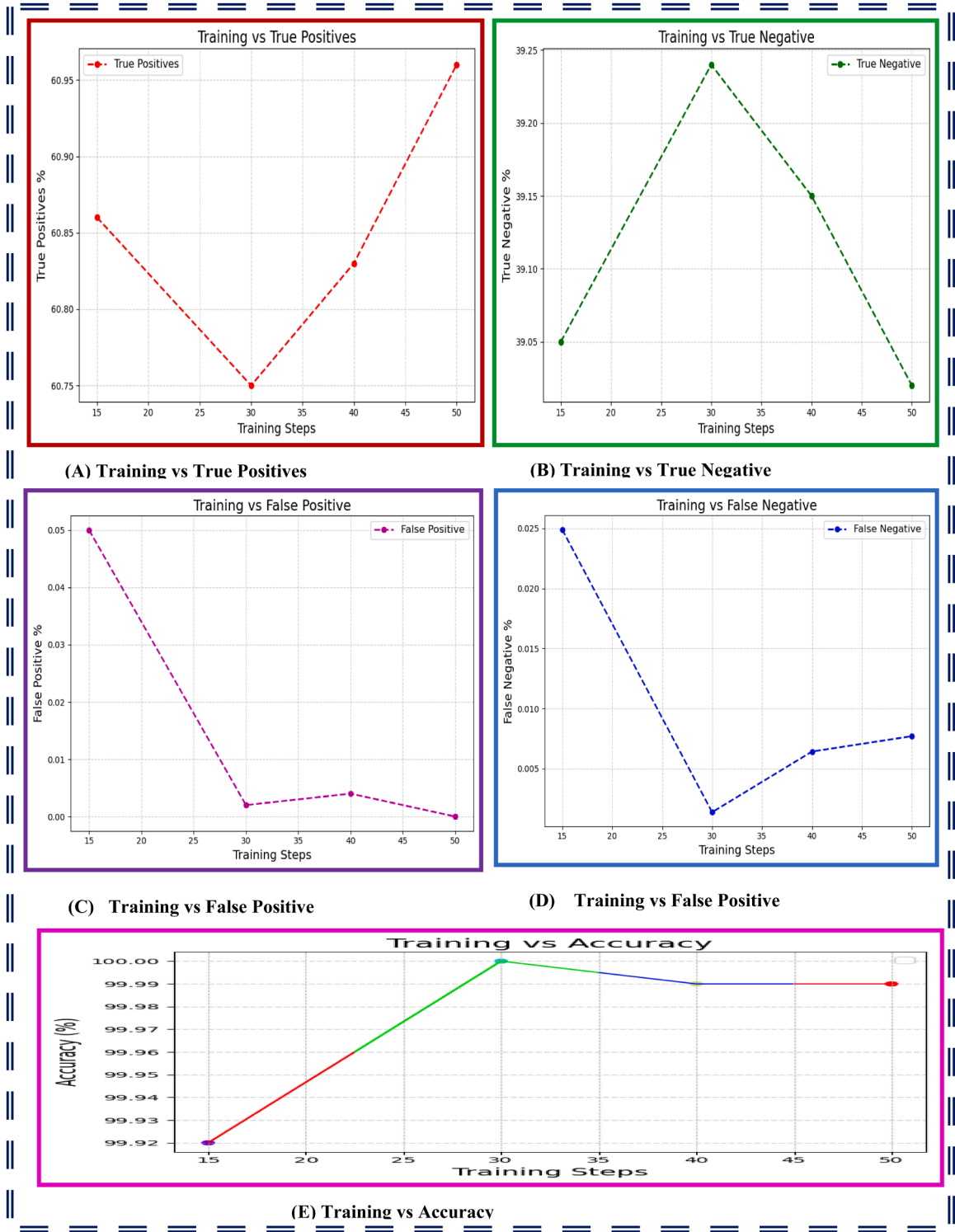


Fig. 17. Performance metrics of the training dataset: TP, TN, FP, FN, trends in detecting DDoS attack.

ideal number of training steps to balance accuracy and efficiency is thirty.

In Fig. 18, the classification report of proposed approach shows high overall accuracy of 99.92% (Training Data: 15% Testing Data: 85%) 100% (Training Data: 30% Testing Data: 70%), 99.99% (Training Data: 40% Testing Data: 60%), 99.99% (Training Data: 50% Testing Data: 50%) with excellent precision, recall, and F1-scores for both classes (0 (Normal), and 1 (DDoS Attack)). The macro and weighted average confirm that the performance of the proposed approach is balanced

across the datasets. Further, getting substantial support for each class makes the results reliable and robust. These results indicate that the model is highly effective in distinguishing between regular and attack traffic, even under different training and testing splits. The consistency of accuracy across varying data partitions highlights the stability and generalization capacity of the proposed method. Moreover, the substantial precision and recall values demonstrate that the model not only reduces false positives but also minimizes false negatives, which is critical in security-sensitive applications. Overall, this performance

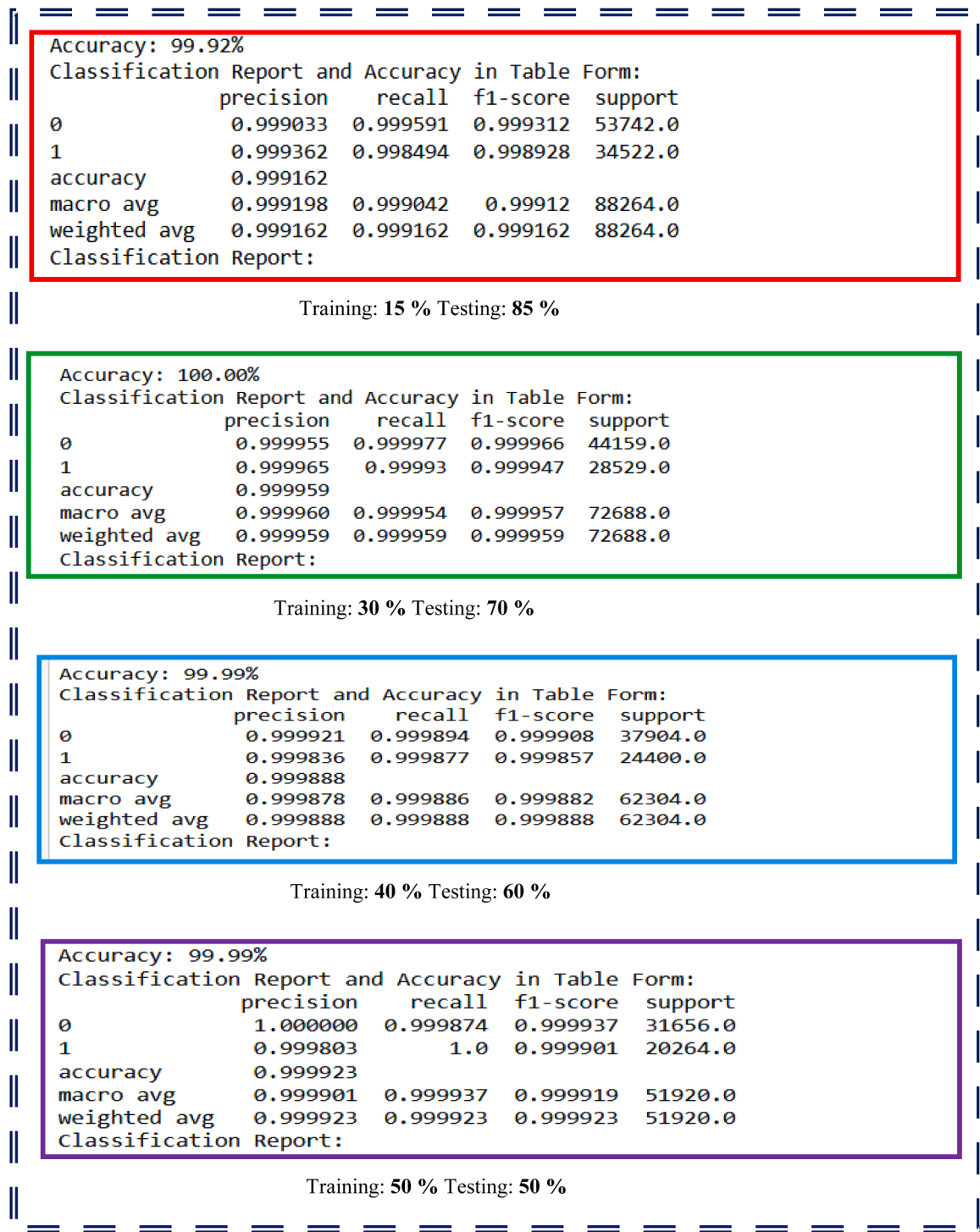


Fig. 18. Accuracy report of various training and testing datasets.

shows that the proposed approach is capable of delivering trustworthy real-time detection in cloud environments. Compared to traditional threshold-based detection methods, the proposed model achieves far superior accuracy and adaptability, making it a more reliable solution for evolving DDoS attack scenarios.

5.5.3. Cross validation and statistical analysis of the proposed model

Cross-validation performance analysis provides a robust mechanism to evaluate the generalizability of a model across different data partitions. At the same time, statistical significance testing ensures that observed improvements are not due to random variation. In this context,

the HyMLCPP model is compared with baseline approaches to validate its stability, reliability, and superiority through both distribution-based performance metrics and hypothesis testing.

i. Cross-Validation Performance Analysis of HyMLCPP Model:

To rigorously assess the predictive capability of the HyMLCPP model, cross-validation was conducted across multiple folds. This evaluation strategy reduces bias associated with a single train–test split and provides a more precise measure of model generalization. Analysis of accuracy, precision, recall, and F1 scores across folds further

illustrates the model's stability and consistent performance under diverse data partitions. The detailed outcomes of this cross-validation process, including fold-wise metrics and comparative performance against baseline models, are summarized in Table 6.

The results in Table 6 highlight that Bagged Trees achieved the best overall performance, reaching near-perfect accuracy and F1 scores. This can be attributed to the ensemble nature of bagging, which reduces variance and enhances robustness against overfitting. However, despite slightly lower accuracy, the HyMLCPP model demonstrates strong competitiveness with stable precision and recall values, while also offering the advantage of privacy-preserving mechanisms that traditional models do not provide. The Decision Tree model achieved reasonably high accuracy but was slightly less stable across folds, and Boosted Trees lagged due to their sensitivity to noisy patterns in the dataset. These findings indicate that while Bagged Trees lead in raw performance, HyMLCPP provides a balanced trade-off between accuracy, stability, and enhanced security features, making it suitable for deployment in real-world sensitive environments.

ii. Pre-fold cross-validation Analysis of the proposed model:

Per-fold cross-validation analysis offers a detailed view of how individual models perform across different partitions of the dataset. Instead of relying solely on average metrics, this approach highlights the stability, variability, and consistency of each classifier from one fold to another. Presenting the fold-wise results for HyMLCPP alongside baseline methods provides more unmistakable evidence of its reliability, demonstrating how the model maintains balanced accuracy, precision, recall, and F1 scores under repeated testing conditions.

The per-fold cross-validation results in Table 7 indicate that *Bagged Trees* achieved the best overall performance, with accuracy and F1 scores consistently close to 1.0, reflecting exceptional stability. The *Decision Tree* also performed strongly, with accuracy between 0.982 and 0.986 and F1 scores around 0.978–0.982, though with slightly higher variability. The proposed *HyMLCPP model* showed competitive and reliable performance, maintaining accuracy in the 0.980–0.982 range and F1 scores around 0.975, consistent with its average results. In contrast, *Boosted Trees* demonstrated weaker generalization, with notably lower accuracy (≈ 0.894) and F1 (≈ 0.868), highlighting their reduced stability across folds. While Bagged Trees attained the highest classification accuracy, the HyMLCPP model demonstrated competitive performance with consistent stability across validation folds. The balanced outcomes in terms of precision, recall, and F1 score emphasize its robustness and reliability. Consequently, HyMLCPP emerges as a suitable candidate for deployment in security-sensitive and mission-critical environments.

Fig. 19 represents the accuracy distribution across folds for different models. The boxplot shows that Bagged Trees achieve nearly perfect accuracy with minimal variance, highlighting their robustness and stability. In contrast, Boosted Trees exhibit lower accuracy and higher variability, while Decision Tree and HyMLCPP provide competitive but slightly lower performance compared to Bagged Trees.

Fig. 20 illustrates F1 score distribution across folds for different models. Bagged Trees consistently achieve high F1 scores with minimal variability, reflecting their balanced performance between precision and recall. Boosted Trees show comparatively lower F1 values, whereas Decision Tree and HyMLCPP provide competitive results.

Table 6

Cross-validation performance metrics of HyMLCPP and baseline models.

Model	Accuracy_Mean	Accuracy_Std	F1_Mean	F1_Std	Precision_Mean	Recall_Mean
Bagged Trees	0.999626	0.000114	0.999522	0.000146	0.999461	0.999583
Decision Tree	0.984388	0.001382	0.980135	0.001712	0.975145	0.98519
HyMLCPP	0.980794	0.000856	0.975532	0.001141	0.971508	0.9796
Boosted Trees	0.894887	0.001979	0.868312	0.002281	0.850774	0.886598

Table 7

Per-fold cross-validation results for HyMLCPP and baseline models.

Model	Fold	Accuracy	F1	Precision	Recall
Decision Tree	1	0.983708	0.979286	0.973238	0.985410
Decision Tree	2	0.982414	0.977672	0.970411	0.985044
Decision Tree	3	0.986008	0.982110	0.981629	0.982592
Decision Tree	4	0.984666	0.980533	0.973192	0.987986
Decision Tree	5	0.985145	0.981072	0.977253	0.984921
Bagged Trees	1	0.999473	0.999325	0.999632	0.999019
Bagged Trees	2	0.999712	0.999632	0.999510	0.999755
Bagged Trees	3	0.999617	0.999510	0.999265	0.999755
Bagged Trees	4	0.999760	0.999693	1.000000	0.999387
Bagged Trees	5	0.999569	0.999449	0.998898	1.000000
Boosted Trees	1	0.894916	0.868454	0.850147	0.887567
Boosted Trees	2	0.894149	0.867351	0.850047	0.885375
HyMLCPP	1	0.981032	0.976021	0.971222	0.980750
HyMLCPP	2	0.979814	0.974885	0.969874	0.979102
HyMLCPP	3	0.981254	0.975338	0.972001	0.978950
HyMLCPP	4	0.980112	0.975742	0.970853	0.979485
HyMLCPP	5	0.981758	0.975175	0.973591	0.980714

iii. Statistical Significance Analysis of HyMLCPP vs Baseline Models:

To strengthen the reliability of the evaluation, statistical significance testing was performed to compare the proposed HyMLCPP model against established baseline classifiers. These tests help determine whether the observed performance differences are due to genuine improvements rather than random variation. Both parametric (paired *t*-test) and non-parametric (Wilcoxon signed-rank test) approaches were applied, providing complementary insights into the robustness of the results. This analysis ensures that the comparative evaluation of HyMLCPP is both rigorous and statistically sound. The outcomes of the statistical significance analysis, including pairwise comparisons of F1 scores between HyMLCPP and baseline models, are presented in Table 8 to highlight the reliability and validity of the observed performance differences. These statistical results complement the cross-validation findings by confirming that the observed performance trends are not incidental but rather reflect consistent and reliable differences across models.

Table 8 highlights the *Statistical Significance Analysis of the proposed model*. This table presents the results of pairwise statistical significance tests comparing the F1 scores of the HyMLCPP model with baseline classifiers. Both paired *t*-tests and Wilcoxon signed-rank tests were conducted to assess whether the observed performance differences are statistically meaningful. The results indicate that Bagged Trees significantly outperform HyMLCPP, while the Decision Tree shows a smaller but still statistically meaningful advantage. In contrast, HyMLCPP achieves a clear and significant improvement over Boosted Trees, confirming its competitive reliability among the evaluated models.

Fig. 21 presents the mean accuracy with standard deviation for different models. Bagged Trees achieve the highest accuracy with minimal variability, indicating strong stability. Decision Tree and HyMLCPP also deliver competitive results. Decision Tree and HyMLCPP also provide strong results, maintaining high accuracy with relatively low variability, making them competitive alternatives. In contrast, Boosted Trees show a noticeable drop in mean accuracy and exhibit slightly higher uncertainty, suggesting weaker generalization compared to the other models.

Fig. 22 compares the mean F1 scores with standard deviation for

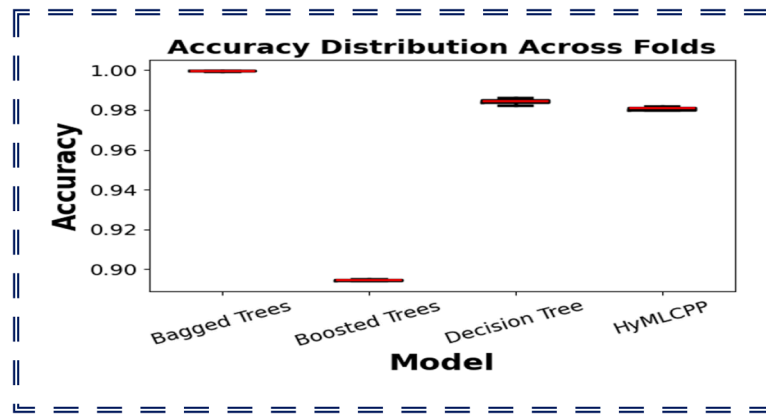


Fig. 19. Cross-fold accuracy distribution of models.

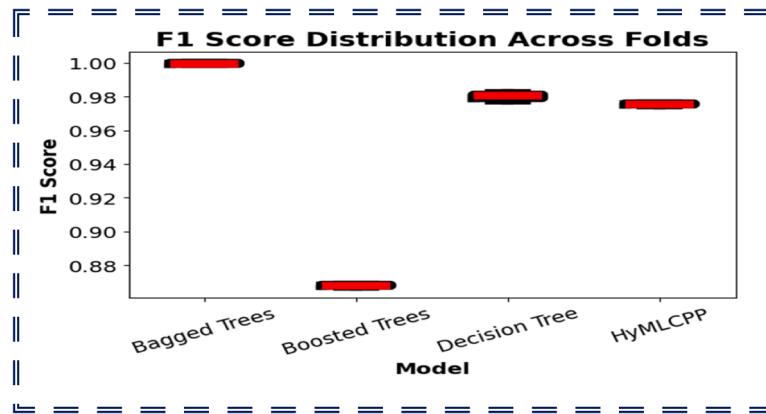


Fig. 20. Cross-fold F1 score distribution of models.

Table 8

. Statistical comparison of HyMLCPP with baseline models.

Comparison	HyMLCPP F1	Baseline F1	t-stat	p-value (t-test)	Wilcoxon Stat	Wilcoxon p-value
HyMLCPP vs Decision Tree	0.975532	0.980135	-5.259	0.00626	0	0.0625
HyMLCPP vs Bagged Trees	0.975532	0.999522	-43.475	1.67E-06	0	0.0625
HyMLCPP vs Boosted Trees	0.975532	0.868312	131.26	2.02E-08	0	0.0625

Bagged Trees, Decision Tree, HyMLCPP, and Boosted Trees, including statistical significance tests. Bagged Trees achieve the highest F1 score with minimal variability, confirming superior balance between precision and recall. Decision Tree and HyMLCPP provide competitive performance but remain statistically lower than Bagged Trees. Boosted Trees perform the weakest, with significantly lower F1 scores, indicating reduced classification effectiveness.

5.6. HyMLCPP model as predictive probability model for enhancing security and privacy in cloud computing

The proposed model not only enhances the security in the cloud environment, but it also tends to predict future events of an attack. By analyzing the traffic over a network, the model has the potential to provide predictions about future chances of occurrence of such attacks. Hence, it makes the proposed model extraordinary. The proposed HyMLCPP model implements the Bayesian Network, which is a Probabilistic graphical model to represent a set of variables. The conditional dependencies are represented by a DAG (Directed Acyclic Graph). It combines the theory of graphs and probability. The variables used in this model are as follows:

w : It represents High Network Traffic.

x : It represents a High Packet Rate.

y : It represents Periodic Connection.

z : It represents the occurrence of DDoS attacks.

Further, the following relationships can be established between variables 'w', 'x', and 'y' to obtain the outcome 'z' for finding out whether the attacks occurred or not:

$z : DDoSattackisimpactedbyw, x, y.$

In our Bayesian network, these variables depend on whether a DDoS attack (z) has occurred or not.

$z \rightarrow w; z \rightarrow x; z \rightarrow y$

Eq. (24) should be true for the occurrence of an attack.

$$w = true, x = true, \text{ and } y = true \quad (24)$$

To calculate the posterior probability (PP) of 'z', Eq. (25) will be applicable.

$$(z = true) \text{ i.e. } PP(z = true | w = true, x = true, y = true) \quad (25)$$

We can get the probability of the occurrence of an attack using Eq. (26).

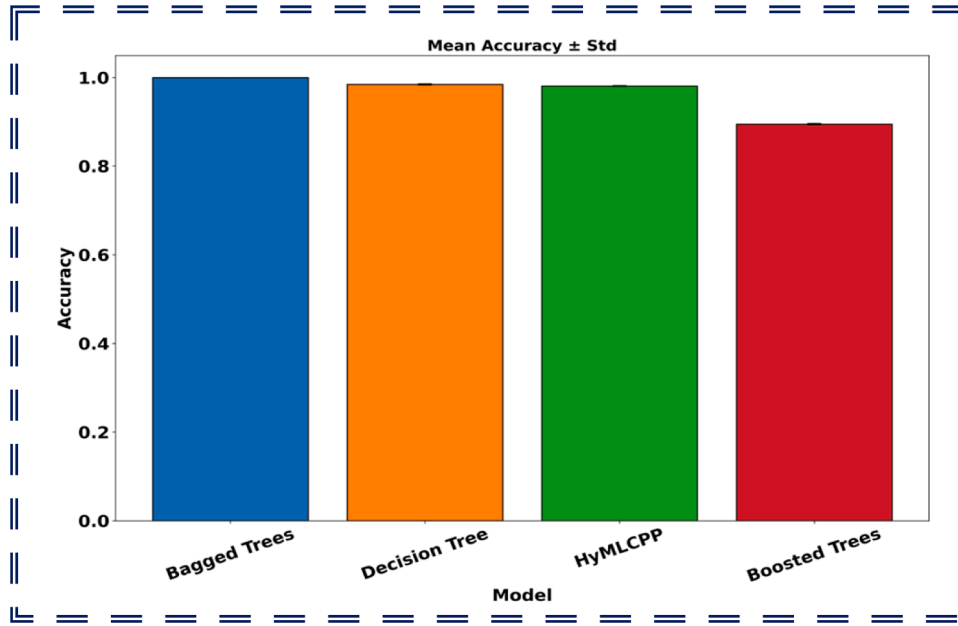


Fig. 21. Mean accuracy with standard deviation across models.

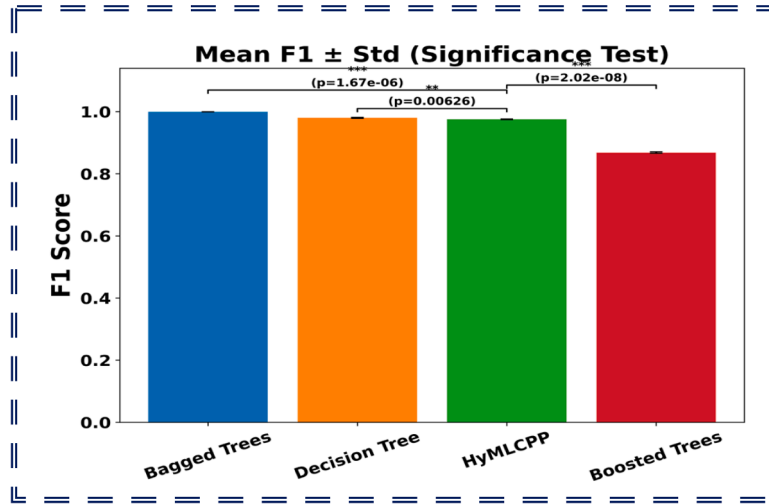


Fig. 22. Mean F1 score with standard deviation and significance testing across models.

$$Probability(z|w, x, y) = \frac{P(w, x, y|z)P(z)}{P(w, x, y)} \quad (26)$$

Where, $P(w, x, y|z) = P(w|z), P(x|z), P(y|z)$, and $P(w, x, y) = P(w, x, y|z = true)P(z = true) + P(w, x, y|z = false)P(z = false)$

Fig. 23 illustrates a probabilistic algorithm framework designed to predict DDoS attacks. Large-scale network traffic data can be analyzed using this framework. It uses statistical and ML approaches to find abnormalities related to DDoS activity. Further, the Bayesian network is used to analyze traffic patterns and to forecast the probability of harmful behaviour. The algorithm works by modeling conditional dependencies among different traffic features, which helps capture both direct and indirect relationships. By continuously updating probabilities as new data is observed, the framework adapts to evolving attack strategies. This probabilistic approach not only identifies potential attacks but also estimates the level of risk associated with them. As a result, it provides a more flexible and reliable detection mechanism compared to rigid rule-based systems. When integrated with the HyMLCPP model, this Bayesian framework strengthens the overall defense by combining

probabilistic prediction with privacy-preserving and computationally efficient methods, ensuring robust protection in cloud environments.

Fig. 24 shows that the proposed model implements a Bayesian Network to estimate the probability of a DDoS attack based on observed traffic anomalies and device behaviors. In Fig. 24(A), the probability that *pktcount* will take on a specific value to train the model is represented by each row in the form of CPD. The CPDs provide insight into the behavior of data set features. The $pktcount(0) = 0.494599$ means that approximately 49.46% of the samples have a *pktcount* of 0. Similarly, $pktcount(1) = 0.101553$ means that approximately 10.16% of the samples have a *pktcount* of 1. The higher probability implies that the value is more likely to exist in the dataset. As per the given information in Fig. 24 (B), the algorithm has a minor tendency to identify the event as a DDoS assault, as it indicates a slight edge towards label (1) (DDoS attack).

Thus, the authors resolved the issues of Table 1 in the form of the proposed system. Due to inadequate encryption, sensitive data sent or stored on the cloud is vulnerable to illegal access and manipulation. A proposed model can detect successful DDoS attacks that may occur due to the flow of numerous packets over the network. In addition, the

Input: Traffic Data
Output: Probability of DDoS attack given the evidence.
Algorithm 4: DDoS Attack Detection Using Bayesian Networks (probabilistic model)
 Step 1: Upload the data set(C:/Users/MAYANK/ProjectVehical/archive (3)/dataset_sdn.csv)
 Step 2: Preprocessing data (choosing key features)
 Step 3: Use SMOTE (Synthetic Minority Over-sampling Technique)to address class imbalance.
 Step 4: Describe the structure of a Bayesian network.
 Step 5: Fit inside the model
 Step 6: Execute inference
 inference = VariableElimination(model)
 query_result = inference.query(variables=['label'],
 evidence={'pktcount': 1, # Example values after discretization
 'bytecount': 2,
 'dur': 1,
 'tot_kbps': 2})
 Step 7: Display the Probability of a DDoS attack.
 Step 8: End.

Fig. 23. Probabilistic algorithm framework for predicting DDoS attacks.

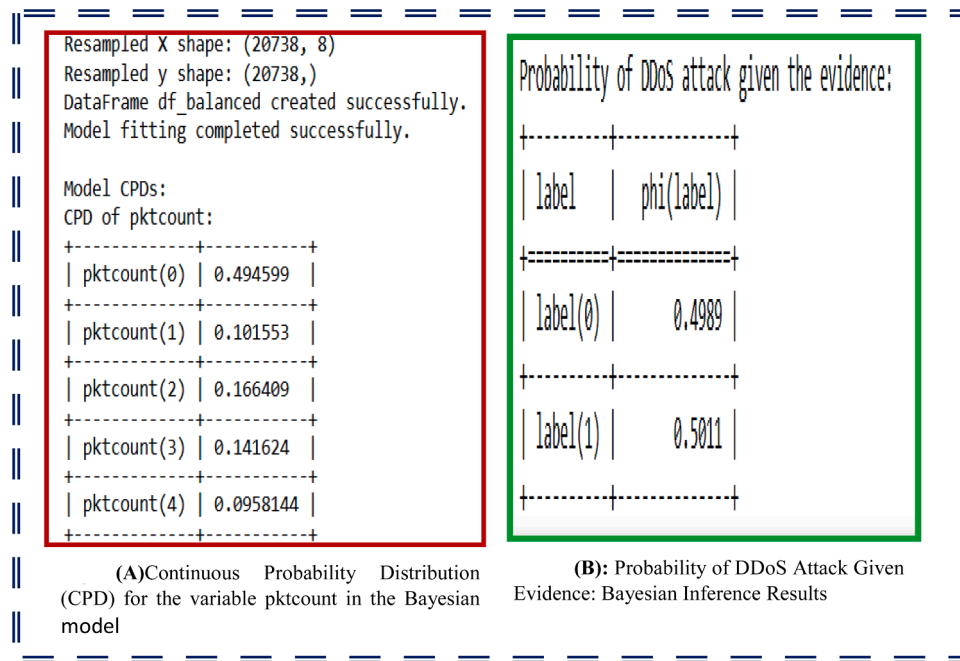


Fig. 24. Bayesian network-based probabilistic model for DDoS attack detection.

proposed model, HyMLCPP, can predict attacks, making it more robust and efficient than other existing systems. The proposed model produces impressive results with almost 100% accuracy. The proposed model is tested on the self-generated and standard data sets.

Table 9 illustrates a comparison between the conventional model and the proposed model. Parameters such as accuracy, adaptability, resource, effort, security, and scalability are examined to highlight the enhancement shown by the proposed method. The proposed model showcases higher accuracy and a superior rate of detection, particularly in complex scenarios, while lowering false positives. The proposed model can handle large datasets while simultaneously increasing the efficiency and productivity of the network system. The comparison presented in Table 9 demonstrates the superiority of the proposed HyMLCPP model in terms of achieving stable and scalable performance. Kaggle. Table 10 presents an in-depth analysis of various researchers' works in terms of multiple domains and parameters used to evaluate the performance of the model they developed. The authors tested the

performance of the proposed model on standard datasets such as the *CICIDS 2017 Dataset*, *Kaggle Dataset*, *AIND DDoS Dataset*, *Datasets of network traffic*, and *CTU-13 Dataset* for DDoS attack detection. It is evident from Table 10 that the proposed model has proved to be superior in terms of reliability and performance parameters. Hence, the proposed model HyMLCPP is considered to be robust and successful in providing security in a cloud computing environment.

6. Conclusions

This research study presents a unique HyMLCPP model for adequate data protection in a cloud computing environment. To provide a robust security system, every module is observed as a doubtful model, and an effective mechanism is created by looking at every viable risk that can be encountered during the process of data transmission and interconnection between various modules. The proposed HyMLCPP model presents a novel encryption method created for the cloud-IoT environment.

Table 9
Performance comparison of traditional model and proposed model.

Aspects	Tradition Method	Proposed Method
Image Encryption Approach	Image encryption involves applying classical cryptographic techniques such as Permutation-Based Methods and Substitution-Based Methods.	The proposed model utilizes advanced image encryption techniques by combining bit plane and chaotic map with ML techniques.
Privacy Preservation	Traditional cloud privacy and data preservation strategies combine process-based approaches, access control systems, and encryption techniques. Security is compromised with the adoption of such methods.	The model proposed with the help of advanced approaches, such as differential privacy and ML approaches, could provide better sharing and preserving data without compromising security. Therefore, protects sensitive information.
Attack Detection Approach	Sets predefined thresholds techniques, Blocklisting, Rate Limiting, Firewall Rules, and Access Control Lists (ACLs)—manual Traffic Inspection for network traffic metrics such as packet rate or bandwidth usage.	Utilizes supervised ML to classify normal and abnormal traffic patterns.
Predictive Approach	Traditionally, Signature-Based Detection (Antivirus), Anomaly-Based Detection, and Rule-Based Systems have been researched for predicting attacks. It produces average results.	The proposed model utilizes a Bayesian Network, serving as a bridge between traditional and modern techniques. The probabilistic proposed model uses a Bayesian Network, which is part of ML techniques. The model provides better results compared to the conventional method.
Complexity	Low; straightforward to implement.	High; requires data pre-processing, feature selection, and training.
Adaptability	Limited manual tuning is needed to adjust thresholds for varying network conditions.	High; automatically learns patterns from data and adapts to new attacks during retraining.
Accuracy	Moderate; effective for known patterns but may fail with sophisticated or evolving attacks.	High; robust in handling complex and dynamic attack patterns.
False Positive Rate	Relatively high; prone to triggering alarms for legitimate traffic near thresholds.	Low; reduces false positives due to better pattern recognition.
False Negative Rate	High; may overlook attacks that fall within threshold limits.	Low; capable of detecting subtle anomalies.
Scalability	Suitable for small-scale networks; struggles with large-scale environments.	Scalable; performs well in large and complex networks.
Performance in HyMLCPP	Moderate; offers baseline detection capabilities but struggles with advanced DDoS patterns.	Excellent; achieves high precision and recall for DDoS detection in HyMLCPP.
Resource Usage	Low; requires minimal computational resources.	High; it depends on the model's complexity and dataset size.
Implementation Effort	Minimal; easy to configure with simple rules.	High; requires expertise in machine learning.
Overall Performance	Average can be used for initial filtering.	Superior: provides comprehensive detection with fewer errors.
Security	Moderate.	High.

Integrating ML and Cryptography components like chaotic maps and Bit-Plane extraction, the proposed model provides unique security features for digital data. It intelligently reduces the computing time. In the world of digital images where different properties of the image exist, the

Table 10
Performance comparison of the proposed model with others in terms of features considered.

Criteria	Gupta et al. (2021)	Brunetta and Pizarro-Sanchez (2022)	Kumari and Mrunalini (2022)	Al-Ghuwairi et al. (2023)	Akram et al. (2024)	Divakar Allavarpu et al. (2024)	Kumar et al. (2023)	Shafique et al. (2021)	Clinton et al. (2024)	HyMLCPP (Proposed)
Features Used	Packet counts, flow duration, bandwidth	Byte rates, port usage, protocol features	Flow-based statistical features	Collaborative feature selection + time series	Image Encryption	Credit Risk Analysis (CRA) + HELR	LSTM	Image Encryption (Bit Plane, Chaotic map, SVM, CNN) Self-generated dataset	Network traffic	Traffic rate, packet size, entropy, flow features
Dataset	Glass, Iris, Wine, Balance Scale	Public dataset	CAIDA 2007	CIC-IDS2018	MNIST	German, Taiwan, Japan, and Australian datasets	CICDDoS2019	Test-bed simulated dataset	Deep Learning (DL)	Kaggle, CICIDS2017, AIND DDoS, CTU-13 Hybrid ML + Cpp
Algorithm / Model	Random Forest	NIST(DRBG)/ MLCrypto (Naive Bayes)	Logistic Regression + Naive Bayes	FS + Prediction Model	Neural Networks + Homomorphic Encryption	HELR (Homomorphic Logistic Regression)	LSTM (DL)	SVM		
Detection Rate (%)	87.0	92.3	97.8	91.2	90.5	80.5	96.9	96.2	99.5	99.8
False Positive Rate (%)	3.1	4.8	2.1	6.3	5.6	7.2	2.4	2.9	3.5	1.2
Accuracy (%)	97	93.1	99.2	92	91	50–70.8	98	97.4	99.7	99.9
Precision (%)	96	90.5	97.2	90.1	89.8	87.8	98	97.8	99.2	99.5
Recall (%)	43	92	98.5	91	90.3	88.2	97.1	98.9	98.4	99.9
F1-Score (%)	95	91.2	99.2	90.5	90	88	97	94	99.1	99.99
Outcome	Strong performance but slightly higher false positives	Effective for distinguishing problems with ML	Excellent for DDoS detection	Reliable for intrusion & network attack detection	Moderate accuracy, limited for complex attacks	Privacy-preserving credit risk analysis	Excellent anomaly detection	Strong for image encryption, reduces computation time	Enhances SDN security by blocking malicious traffic	Highly efficient, scalable, and very low false positives

encryption system of the proposed HyMLCPP model intelligently classifies blocks of digital image into three categories, i.e., low information block, moderate information block, and finally high information block. The ML techniques are used to ensure the appropriate encryption of selected blocks of image. The proposed model provides further better data distribution and sharing guidelines among the users while preserving privacy through the implementation of differential privacy and ML models.

An efficient and effective mechanism for detecting DDoS attacks in a cloud environment is proposed. The proposed model implements a Bayesian network to estimate the probability of a DDoS attack based on observed traffic anomalies and device behaviors. The proposed HyMLCPP model offers a dual-layered protection mechanism by fusing the security of cryptographic techniques with the resilience of ML. The proposed HyMLCPP model attained significant improvement over previous research on the standard datasets like *CICIDS 2017*, *Kaggle*, *AIND DDoS*, *CTU-13*, and *Datasets of network traffic* for DDoS attack detection. It significantly improves accuracy, recall value, f1-score, and precision scores in comparison to the existing methods.

The existing approaches have achieved promising results; their performance remains limited compared to the proposed HyMLCPP model. For example, Gupta et al. (2021) achieved 97% accuracy with a false positive rate of 3.1%, whereas HyMLCPP improves accuracy to 99.9% with only 1.2% false positives. Similarly, Brunetta and Picazo-Sanchez (2022) reported 93.1% accuracy and 90.5% precision, while our model achieves 99.9% accuracy and 99.5% precision. Even advanced deep learning models, such as that of Clinton et al. (2024), obtained 99.7% accuracy but with a higher false positive rate of 3.5%, compared to HyMLCPP's 1.2%. These direct comparisons clearly demonstrate that HyMLCPP offers a significant improvement over prior works, providing both higher detection accuracy and substantially lower false positive rates, while also ensuring secure and reliable data distribution in cloud environments.

The performance comparison of the proposed model on standard datasets with prior work shows that HyMLCPP is more efficient, secure, and superior.

However, the model has some *limitations* that are discussed below:

- %1 The model is tested on a standard dataset that produces fruitful results. Predictive models need to be retrained periodically to ensure that they remain accurate. However, this process can become complex when working with workflows that involve encrypted data.
- %1 Implementing secure storage, encrypted communication methods, and cryptographic operations can result in considerable expenses.
- %1 A breach in the cryptographic layer through weak keys or implementation errors can compromise the security of the entire system. Therefore, a strong key is required for encryption.

The proposed HyMLCPP sets the following grounds for future work:

- The security layer can be further increased in proposed models using federated learning, secure communication protocols, etc.
- The authors should analyze the models on live streaming data to see the deviations in the actual outcomes and the obtained outcomes.
- Artificial intelligence can be implemented in the model to enhance security and mitigate attacks in multiple environments, such as Big Data and IoT (Internet of Things).
- In the proposed approach, the encryption is performed on standard images, but it can also be applied to video and audio datasets to reduce computational time while integrating with cloud-IoT technologies.
- Future research can focus on detecting data leakage and distinguishing between guilty and non-guilty users in cloud data distribution. This distinction is crucial for maintaining fairness,

preventing false accusations, and strengthening the security framework.

Declaration

Funding Open access funding provided by NTNU Norwegian University of Science and Technology (incl St. Olavs Hospital - Trondheim University Hospital).

Availability of data and material

The data and material used in this research study are appropriately referred to and described in this research study.

Code availability

The source code/custom code/software application will be provided when required.

CRediT authorship contribution statement

Mayank Pathak: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Kamta Nath Mishra:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Satya Prakash Singh:** Writing – review & editing, Visualization, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Alok Mishra:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this research study.

References

- Akram, A., Khan, F., Tahir, S., Iqbal, A., Shah, S.A., Baz, A., 2024. Privacy-preserving inference for deep neural networks: optimizing homomorphic encryption for efficient and secure classification. *IEEE Access* 12, 15684–15695. <https://doi.org/10.1109/ACCESS.2024.3357145>.
- Alomoush, W., Khashan, O.A., Alrosan, A., et al., 2023. Digital image watermarking using discrete cosine transformation-based linear modulation. *J. Cloud Comput.* 12, 96. <https://doi.org/10.1186/s13677-023-00468-w>.
- Alshammari, A., El Hindi, K., 2024. Privacy-preserving deep learning framework based on restricted Boltzmann machines and instance reduction algorithms. *Appl. Sci.* 14, 1224. <https://doi.org/10.3390/app14031224>.
- Anil, A., Shukla, V.K., Mishra, V.P., 2020. Enhancing data security using digital watermarking. In: *Proceedings of the 2020 International Conference on Intelligent Engineering and Management (ICIEM)*. London, UK, pp. 364–369. <https://doi.org/10.1109/ICIEM48762.2020.9160090>.
- Al-Ghuwairi, A.R., Sharrab, Y., Al-Fraihat, D., et al., 2023. Intrusion detection in cloud computing based on time series anomalies utilizing machine learning. *J. Cloud Comput.* 12, 127. <https://doi.org/10.1186/s13677-023-00491-x>.
- Boujerfaoui, S., Riad, R., Douzi, H., Ros, F., Harba, R., 2023. Image watermarking between conventional and learning-based techniques: a literature review. *MDPI Electron.* 12, 74. <https://doi.org/10.3390/electronics12010074>.
- Brunetta, C., Picazo-Sanchez, P., 2022. Modelling cryptographic distinguishers using machine learning. *J. Cryptogr. Eng.* 12, 123–135. <https://doi.org/10.1007/s13389-021-00262-x>.
- Carpov, S., Gama, N., Georgieva, M., Jetchev, D., 2022. GenoPPML – a framework for genomic privacy-preserving machine learning. In: *Proceedings of the 2022 IEEE 15th International Conference on Cloud Computing (CLOUD)*. Barcelona, Spain, pp. 532–542. <https://doi.org/10.1109/CLOUD55607.2022.00076>.
- Chatterjee, P., Bose, R., Banerjee, S., et al., 2023. Enhancing data security of cloud-based LMS. *Wirel. Pers. Commun.* 130, 1123–1139. <https://doi.org/10.1007/s11277-023-10323-5>.
- Chenthara, S., Ahmed, K., Wang, H., Whittaker, F., 2019. Security and privacy-preserving challenges of e-Health solutions in cloud computing. *IEEE Access* 7, 74361–74382. <https://doi.org/10.1109/ACCESS.2019.2919982>.

- Chong, B.Y., Salam, I., 2021. Investigating deep learning approaches on the security analysis of cryptographic algorithms. *MDPJ. Cryptography* 5 (4), 30. <https://doi.org/10.3390/cryptography5040030>.
- Clinton, U.B., Hoque, N., Robindro Singh, K., 2024. Classification of DDoS attack traffic on SDN network environment using deep learning. *Cybersecurity* 7, 23. <https://doi.org/10.1186/s42400-024-00219-7>.
- Divakar Allavarpu, V.V.L., Naresh, V.S., Krishna Mohan, A., 2024. Privacy-preserving credit risk analysis based on homomorphic encryption aware logistic regression in the cloud. *Secur. Priv.* 7 (3), e372. <https://doi.org/10.1002/spy2.372>.
- El-Kassabi, H.T., Serhani, M.A., Masud, M.M., et al., 2023. Deep learning approach to security enforcement in cloud workflow orchestration. *J. Cloud Comput.* 12, 10. <https://doi.org/10.1186/s13677-022-00387-2>.
- Fkirin, A., Attiya, G., El-Sayed, A., et al., 2022. Copyright protection of deep neural network models using digital watermarking: a comparative study. *Multimed. Tools Appl.* 81, 15961–15975. <https://doi.org/10.1007/s11042-022-12566-z>.
- Fu, Z., Wang, F., Cheng, X., 2020. Secure steganography for hiding images via GAN. *J. Image Video Process.* 2020, 46. <https://doi.org/10.1186/s13640-020-00534-2>.
- Fu, Z., Xia, L., Sun, X., Liu, A.X., Xie, G., 2018. Semantic-aware searching over encrypted data for cloud computing. *IEEE Trans. Inf. Forensics Secur.* 13 (9), 2359–2371. <https://doi.org/10.1109/TIFS.2018.2819121>. Sept.
- Garima, Quraishi, S.J., 2022. Machine learning approach for cloud computing security. In: *Proceedings of the 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*. London, United Kingdom, pp. 158–163. <https://doi.org/10.1109/ICIEM54221.2022.9853056>.
- Ge, C., Liu, Z., Susilo, W., Fang, L., Wang, H., 2024. Attribute-based encryption with reliable outsourced decryption in cloud computing using smart contract. *IEEE Trans. Dependable Secure Comput.* 21 (2), 937–948. <https://doi.org/10.1109/TDSC.2023.3265932>. March–April.
- Gupta, I., et al., 2019b. Layer-based privacy and security architecture for cloud data sharing. *Semantic Scholar J. Commun. Softw. Syst.* <https://doi.org/10.24138/jcomss.v15i2.617>.
- Gupta, I., Gupta, R., Singh, A.K., Buyya, R., 2021. MLPAM: a machine learning and probabilistic analysis based model for preserving security and privacy in cloud environment. *IEEE Syst. J.* 15 (3), 4248–4259. <https://doi.org/10.1109/JSYST.2020.3035666>. Sept.
- Gupta, I., Singh, A.K., Lee, C.N., Buyya, R., 2022b. Secure data storage and sharing techniques for data protection in cloud environments: a systematic review, analysis, and future directions. *IEEE Access* 10, 71247–71277. <https://doi.org/10.1109/ACCESS.2022.3188110>.
- Gupta, I., Singh, N., Singh, A.K., 2019a. Layer-based privacy and security architecture for cloud data sharing. *J. Commun. Softw. Syst.* 15 (2), 173–185. <https://doi.org/10.24138/jcomss.v15i2.617> pages.
- Gupta, S., Saluja, K., Solanki, V.V., Wang, K., Singla, P., Shahid, M., 2022a. Efficient methods for digital image watermarking and information embedding. *Meas. Sens.* 24, 100520. <https://doi.org/10.1016/j.measen.2022.100520>.
- Guvencoglu, E., 2024. An image encryption algorithm based on multi-layered chaotic maps and its security analysis. *Connect. Sci.* 36 (1). <https://doi.org/10.1080/09540091.2024.2312108>.
- Jia, H., Cai, D., Yang, J., et al., 2023. Efficient and privacy-preserving image classification using homomorphic encryption and chunk-based convolutional neural network. *J. Cloud Comput.* 12, 175. <https://doi.org/10.1186/s13677-023-00537-0>.
- Kareem, M.I., Jasim, M.N., 2021. The current trends of DDoS detection in SDN environment. In: *Proceedings of the 2021 2nd Information Technology To Enhance E-Learning and Other Application (IT-ELA)*. Baghdad, Iraq, pp. 29–34. <https://doi.org/10.1109/IT-ELA52201.2021.9773744>.
- Kaur, G., Agarwal, R., Patidar, V., 2020. Crypto-watermarking of images for secure transmission over cloud. *J. Inf. Optim. Sci.* 41 (1), 205–216. <https://doi.org/10.1080/02522667.2020.1714185>.
- Kaur, G., Bharathiraja, N., Murugesan, S., Pradeepa, K., Sudhakar, G., Vinoth Kumar, K. M., 2023. A security model with efficient AES and Security performance trade-off analysis of cryptography systems with cloud computing. In: *Proceedings of the 2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. Erode, India, pp. 01–08. <https://doi.org/10.1109/ICECCT56650.2023.10179752>.
- Kelur, S., Ranjan Kumar, H.S., Raju, K., 2019. Selective area encryption using machine learning technique. In: *Proceedings of the 2019 Innovations in Power and Advanced Computing Technologies (i-PACT)*. Vellore, India, pp. 1–7. <https://doi.org/10.1109/i-PACT44901.2019.8959954>.
- Khan, Y., Ahmad, S., Asghar, M.Z., 2024. Customized model for text cryptography using hybrid image-key. *Signal Image Video Process.* 18 (Suppl 1), 965–978. <https://doi.org/10.1007/s11760-024-03208-y>.
- Kumar, A., 2022. A cloud-based buyer-seller watermarking protocol (CB-BSWP) using a semi-trusted third party for copy deterrence and privacy-preserving. *Multimed. Tools Appl.* 81, 21417–21448. <https://doi.org/10.1007/s11042-022-12550-7>.
- Kumar, B.S., Revathi, R., 2024. An efficient image encryption algorithm using a discrete memory-based logistic map with a deep neural network. *J. Eng. Appl. Sci.* 71, 41. <https://doi.org/10.1186/s44147-023-00349-8>.
- Kumar, D., Pateriya, R.K., Gupta, R.K., Dehalwar, V., Sharma, A., 2023. DDoS detection using machine learning. *Procedia Comput. Sci.* 218, 2420–2429. <https://doi.org/10.1016/j.procs.2023.01.217>.
- Kumar, P.R., Goel, S., 2023. Integrating machine learning algorithms with an advanced encryption scheme: enhancing data security and privacy. *Int. J. Intell. Syst. Appl. Eng.* 11 (4), 453–465. <https://ijisae.org/index.php/IJISAE/article/view/3543>.
- Kumari, K., Mrunalini, M., 2022. Detecting denial of service attacks using machine learning algorithms. *J. Big Data* 9, 56. <https://doi.org/10.1186/s40537-022-00616-0>.
- Lee, T., Moon, H.S., Jang, J., 2021. Data encryption method using CP-ABE with symmetric key algorithm in the blockchain network. In: *Proceedings of the 2021 International Conference on Information and Communication Technology Convergence (ICTC)*. Jeju Island, Korea, Republic of, pp. 1371–1373. <https://doi.org/10.1109/ICTC52510.2021.9620889>.
- Liu, L., Gao, M., Zhang, Y., et al., 2022. Application of machine learning in intelligent encryption for digital information of real-time image text under big data. *J. Wirel. Commun. Netw.* 2022, 21. <https://doi.org/10.1186/s13638-022-02111-9>.
- Meng, F., Cheng, L., Wang, M., 2021. Ciphertext-policy attribute-based encryption with a hidden sensitive policy from keyword search techniques in a smart city. *J. Wirel. Commun. Netw.* 2021, 20. <https://doi.org/10.1186/s13638-020-01875-2>.
- Mishra, A., Jabar, T.S., Alzoubi, Y.I., Mishra, K.N., 2023. Enhancing privacy-preserving mechanisms in cloud storage: a novel conceptual framework. *Concurr. Comput. Pract. Exp.* 35 (26), e7831. <https://doi.org/10.1002/cpe.7831>.
- Naik, R.B., Singh, U., 2024. A review on applications of chaotic maps in pseudo-random number generators and encryption. *Ann. Data Sci.* 11, 25–50. <https://doi.org/10.1007/s40745-021-00364-7>.
- Pathak, M., Mishra, K.N., Singh, S.P., 2024. Securing data and preserving privacy in cloud IoT-based technologies analysis of assessing threats and developing effective safeguard. *Artif. Intell. Rev.* 57, 269. <https://doi.org/10.1007/s10462-024-10908-x>.
- Ping, L., Tong, L., Heng, Y., Jin, L., Chen, X., Yang, X., 2018. Privacy-preserving machine learning with multiple data providers. *Future Gener. Comput. Syst.* 87, 341–350. <https://doi.org/10.1016/j.future.2018.04.076>.
- Priya, M.J., Yamuna, G., 2022. Privacy-preserving data security model for cloud computing technology. In: *Proceedings of the 2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*. Villupuram, India, pp. 1–5. <https://doi.org/10.1109/ICSTSN53084.2022.9761350>.
- Pulido-Gaytan, B., Tchernykh, A., MCortés-Mendoza, J., et al., 2021. Privacy-preserving neural networks with homomorphic encryption: challenges and opportunities. *Peer-to-Peer Netw. Appl.* 14, 1666–1691. <https://doi.org/10.1007/s12083-021-01076-8>.
- Qayyum, A., Qadir, J., Bilal, M., Al-Fuqaha, A.I., 2020. Secure and robust machine learning for healthcare: a survey. *IEEE Rev. Biomed. Eng.* 14, 156–180. <https://doi.org/10.1109/RBME.2020.3013489>.
- Qayyum, A., Qadir, J., Bilal, M., Al-Fuqaha, A., 2021. Secure and robust machine learning for healthcare: a survey. *IEEE Rev. Biomed. Eng.* 14, 156–180. <https://doi.org/10.1109/RBME.2020.3013489>.
- Sana, M.U., Li, Z., Javaid, F., Liaqat, H.B., Ali, M.U., 2021. Enhanced security in cloud computing using neural network and encryption. *IEEE Access* 9, 145785–145799. <https://doi.org/10.1109/ACCESS.2021.3122938>.
- Sandoval Orozco, L., Huamán, C.Q., Povedano Álvarez, D., García Villalba, L.J., 2020. A machine learning forensics technique to detect post-processing in digital videos. *Future Gener. Comput. Syst.* 111, 199–212. <https://doi.org/10.1016/j.future.2020.04.041>.
- Shafique, A., Ahmed, J., Boulila, W., Ghandorh, H., Ahmad, J., Rehman, M.U., 2021. Detecting the security level of various cryptosystems using machine learning models. *IEEE Access* 9, 9383–9393. <https://doi.org/10.1109/ACCESS.2020.3046528>.
- Shafique, A., Mehmood, A., Alawida, M., et al., 2024. A fusion of machine learning and cryptography for fast data encryption through the encoding of high and moderate plaintext information blocks. *Multimed. Tools Appl.* <https://doi.org/10.1007/s11042-024-18959-6>.
- Shaikh, Z.A., et al., 2024. A new trend in cryptographic information security for industry 5.0: a systematic review. *IEEE Access* 12, 7156–7169. <https://doi.org/10.1109/ACCESS.2024.3351485>.
- Sharma, D.K., Singh, N.C., Noola, D.A., Doss, A.N., Sivakumar, J., 2022. A review on various cryptographic techniques & algorithms. *Mater. Today Proc.* 51 (Part 1), 104–109. <https://doi.org/10.1016/j.matpr.2021.04.583>. Pages.
- Singh, A.K., Gupta, R., 2022. A privacy-preserving model based on a differential approach for sensitive data in a cloud environment. *Multimed. Tools Appl.* 81, 33127–33150. <https://doi.org/10.1007/s11042-021-11751-w>.
- Tanuwidjaja, H.C., Choi, R., Baek, S., Kim, K., 2020. Privacy-preserving deep learning on machine learning as a service—a comprehensive survey. *IEEE Access* 8, 167425–167447. <https://doi.org/10.1109/ACCESS.2020.3023084>.
- Terziyan, V., Malyk, D., Golovianko, M., Branytskyi, V., 2023. Encryption and generation of images for privacy-preserving machine learning in smart manufacturing. *Procedia Comput. Sci.* 217, 91–101. <https://doi.org/10.1016/j.procs.2022.12.205>.
- Wang, S., Zhou, J., Liu, J.K., Yu, J., Chen, J., Xie, W., 2016. An efficient file hierarchy attribute-based encryption scheme in cloud computing. *IEEE Trans. Inf. Forensics Secur.* 11 (6), 1265–1277. <https://doi.org/10.1109/TIFS.2016.2523941>. June.
- Yang, Q., Huang, A., Fan, L., et al., 2023. Federated learning with privacy-preserving and model IP-right-protection. *Mach. Intell. Res.* 20, 19–37. <https://doi.org/10.1007/s11633-022-1343-2>.
- Yuan, K., Huang, Y., Du, Z., et al., 2024. A multi-layer composite identification scheme of cryptographic algorithm based on hybrid random forest and logistic regression model. *Complex Intell. Syst.* 10, 1131–1147. <https://doi.org/10.1007/s40747-023-01212-2>.



Mr. Mayank Pathak is from the Jasidih district of Deoghar, Jharkhand, India. He earned a Master's in Computer Applications (MCA) from Birla Institute of Technology Mesra. He has completed his 10th board from St. Francis School, Jasidih, and 12th from Dev Sangha National School, Deogarh. He earned his Bachelor's degree in Computer Applications from Sidhu Kanu Murmu University, Dumka. He has more than 5 years of teaching experience in the field of computer science. Currently, he is pursuing his Ph.D. from Birla Institute of Technology, Mesra. His research interest includes Cloud Computing, Cloud Security, IoT (Internet of Things), Image processing, Big Data Security, Machine Learning, and Artificial Intelligence.



Dr. Satya Prakash Singh is presently working as an Assistant Professor (Selection Grade) in the Department of Computer Science & Engineering, Birla Institute of Technology, Mesra, Ranchi, India. He has 24 years of teaching experience. He received his Ph. D. degree in Computer Science and Engineering in 2012. He has received an M. Tech. Degree in Computer Science & Engineering in 1999. He received an M. Sc Physics (Electronics) degree in 1997. He has supervised many research scholars of MCA, M. Tech., and Ph.D. degrees. He has published more than 65 research studies in Scopus-indexed journals, IEEE, Springer, and Elsevier conferences, and many other national and international journals and conferences. He has been an editorial and reviewer in the Journal of Green Computing (Index in Scopus). His areas of interest include artificial intelligence, Data Mining, and Image Processing. Parallel and Distributed Computing, IoT, and Software Engineering.



Dr. Kamta Nath Mishra earned his Bachelor of Science (B.Sc., Maths) degree from the University of Gorakhpur, India, in 1992, followed by a Master of Computer Application (MCA, 3 Years) degree from Madan Mohan Malviya Engineering College (currently MMMUT), Gorakhpur, Uttar Pradesh, India, in 1996. In 2003, he completed his Master of Technology (M.Tech.) (Software Systems) degree from Birla Institute of Technology and Science (BITS) Pilani, India, and obtained his Ph.D. (Engg.) from the CSE department of B.I.T. Mesra, India, in May 2015. Further, he obtained his Post Doctoral Research from NTNU, Norway. With over twenty-five years of teaching and research experience, Dr. Mishra currently holds a senior faculty position (Associate Professor Level) at the Department of CSE, B.I.T.

Mesra, Ranchi, India, since August 2009. Previously, he served as a faculty member in the Department of Computer Science at the University of Sebha, a public university in Libya, from October 2006 to July 2009. From July 2004 to September 2006, he was a senior lecturer at B.I.T. Mesra (Noida Campus). Before that, Dr. Mishra worked as a senior project engineer (September 2003 to June 2004) and project engineer (September 2000 to August 2003) at the Centre for Development of Advanced Computing (CDAC) in Noida, Uttar Pradesh, which falls under the Ministry of Communication & IT, Government of India. Before joining CDAC, he served as a lecturer in the CSE department at Krishna Institute of Engineering & Technology (KIET), Ghaziabad, India, from July 1998 to August 2000. Dr. Mishra has authored three books, contributed to twenty book chapters (published by Springer, IGI Global, and Elsevier), and published sixty research studies in reputable international journals and conferences. His research interests encompass Biometric Systems, Image Processing, algorithm analysis, Cloud Computing, Smart Society Computations, and Distributed Cloud Computing. He holds professional memberships in the IEEE Biometrics Society (USA) and the ACM (USA).



Prof. Alok Mishra (Senior Member, IEEE) is a Professor in Data Management & Software Engineering at Norwegian University of Science and Technology (NTNU), Norway. His research interests include software engineering, artificial intelligence, and cybersecurity. He is actively involved in editing special issues of reputed journals in his areas of research interest. In teaching, he has received the Excellence in Online Education Award by U21Global Singapore, while in research, he has been awarded by the Scientific and Research Council of Turkey and the Board of Management of the University for outstanding publications in science and social science citation-indexed (Thomson Reuters) journals. He was a recipient of many scholarships, international awards, and research projects. He is an Editorial Board Member of reputed journals, including Computer Standards & Interfaces (Elsevier), ICT Express, Software Impacts, and Data Technologies and Applications.