Svedberg Open
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Hybrid Intrusion Detection System For Securing Information In The Cloud Environment

Ashima Jain[1], Ashima Narang[2], Manju[3*]

[1,2]Department of Computer Science and Engineering, Amity University, Gurugram, Haryana, India

[3]Department of Computer Science and Engineering, PES University, Bangalore, India

Corresponding author: ashimajain046@gmail.com[1], Ashimanarang04@gmail.com[2], manju.nunia@gmail.com[3]

Abstract

Cloud environments face escalating security risks due to their distributed and multi-tenant nature, rendering traditional intrusion detection systems insufficient. Cloud setting become attractive targets for various cyber-attacks besides these advantages, thus security is a big concern for both cloud consumers and providers. Traditional security standards may not work for dynamic, multi-tenant, distributed cloud architecture, necessitating more advanced and powerful intrusion detection solutions. Thus, we are proposing a Hybrid Intrusion Detection Model (HIDM) that integrates rule-based filtering, machine learning, and deep learning to enhance cloud network protection. The framework employs a three-layer detection strategy and ensemble weighted voting to minimize false positives while detecting both known and zero-day attacks. Evaluations are done using five benchmark datasets which includes NSL-KDD, CICIDS2018, UNSW-NB15, CIC-DDoS2019, and CIC Bell DNS EXF2021which shows that HIDM achieves 97.74% accuracy, 97.99% F1-score, and a 1.14% false positive rate, outperforming single-model approaches. There is various comparative analysis of accuracy are also given in the simulation section. Unlike prior works, HIDM uniquely combines adaptive learning with real-time deployment capabilities, demonstrating a novel, scalable, and explainable approach to intelligent intrusion detection in the cloud.

Keywords: Cloud Computing, Intrusion Detection System, Security, Machine Learning, Deep Learning

1. Introduction

The way that programs and data are managed and delivered has been revolutionized as a result of cloud computing, which has enabled businesses to swiftly expand their operations. In the opposite direction, the utilization of cloud technology has also resulted in the creation of new potential for cyber-attacks. As a result of their dynamic and multi-tenant nature, cloud infrastructures are especially vulnerable to forms of cyber-attacks data breaches, privilege escalation attacks [1]. Intrusion detection systems, sometimes known as IDS, are crucial technologies because they enable cloud infrastructure monitoring and protection. Although traditional intrusion detection systems may be rendered ineffective in complex cloud environments, hybrid detection models, which mix signature-based and anomaly-based approaches, demonstrate a promising way ahead. So, building a hybrid intrusion detection system that enhances cloud security with smart, real-time intrusion detection characteristics is the main focus of this study. At present, most of the IDS systems are heavily based on the pre-defined attack signatures and sometimes they are trapped in finding false positives during the process of anomaly detection. Authors in [2] suggested that there should be an intelligent, adaptable security system which can consider the changing threats and react accordingly [2]. This gap can be significantly reduced if one can apply either Signature-based or anomaly-based techniques [3-4]. One of the considerable integration technical challenges is to monitor encrypted conversations without compromising on the privacy of the recipient. Recent cyber attacks need to address with respect to the cloud infrastrucre and resource usages.

Study was motivated by the necessity of enhancing intrusion detection through the utilization of a hybrid architecture that not only identifies known threats through the utilization of signature-based techniques, but also adapts to novel threats through the utilization of anomaly detection that is driven by machine learning. The major motivation behind this proposed work is to improve upon the false positives parameter so that it can be significantly reduced during the ma process of real-time detection [5].

This dynamic and distributed nature reveals weaknesses that entice thieves to cloud systems. Conventional IDS sometimes lack the flexibility and scalability needed to manage cloud concerns. Recent research have studied how hybrid models, deep learning, and machine learning can improve IDS performance despite concerns about detection accuracy, false alarm rates, and real-time response. More advanced and prevalent attacks like data breaches, and advanced persistent threats are driving need for robust, adaptive, smart cloud detection systems. Traditional signature or anomaly-based IDS methods have drawbacks when used alone. This work combines both methodologies and adds smart technologies like deep learning and optimization algorithms to create a hybrid intrusion detection system with high accuracy, low false positives, and real-time threat detection in a scalable cloud environment. Cloud security is complicated by resource variety, multi-tenancy, and dispersed control. Intrusion detection systems must be fast enough to adapt to cloud architecture changes while monitoring enormous traffic volumes and minimizing disruption. Current IDS systems struggle with high false positive rates, poor responses, and zero-day attack detection, especially with encrypted or obfuscated real-time communication.By developing and deploying a hybrid intrusion detection framework, this study hopes to overcome the shortcomings of current cloud-based IDS.

The remaining sections of the paper are organized as follows: Section 1 introduces cloud computing, the importance of intrusion detection, and the motivation for this study. Section 2 provides a detailed review of related literature, highlighting strengths and limitations of existing methods. Section 3 states the research problem and identifies the gaps motivating this work. Section 4 presents the proposed methodology, including the hybrid architecture, datasets, algorithms, and evaluation design. Section 5 discusses experimental results, performance analysis, and comparison with existing approaches. Finally, Section 6 concludes the paper and outlines potential directions for future research.

2. Literature Review

Intrusion detection systems (IDS) have grown significant as they tackle the complex and evolving security concerns in fog settings, Internet of Things (IoT), and cloud computing. Combining several detection systems including signature-based, anomaly-based, and Artificial Intelligence (AI)-based approaches has shown greater accuracy and adaptability. Modi et.al.[1] offered a model called an H-NIDS for cloud environments employing anomaly and signature-based system, hence enhancing threat detection. Zbakh et al. did a comparative study of IDS architectures using multi-criteria decision-making to identify optimal designs suitable for different cloud scenarios [2]. A neural network inspired cooperative strategy along with Snort is addressed by Chiba et al. [3] This work improved few cloud based security concerns which in turn achieved fair detection accuracy. Next, authors introduced another IDS system which is based on hybrid methodology for evaluating the model in a cloud setting where there achieved better detection rates and decreased false positives [5]. Further, swarm optimizing based detecting techniques is addressed by Venkatraman et al [6]. This methodology is essentially building an adaptable hybrid system which is mainly suitable for multimedia based IoT applications in real-time. After this improvement, De Souza et al. [7] proposed fog-based IDS which are again hybrid in nature. This involved with lightweight parts for latency-sensitive applications. Working on cloud security, authors in [8] designed a robust and precise CNN-RNN hybrid deep learning system for better detection. Sharon et al. [9] applied a hybrid Deep Learning (DL) technique combining Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) to identify complex patterns of cloud intrusion. Singh et al. [10] created edge-based IDS in mobile edge situations utilizing hybrid techniques to balance detection efficiency and computation. Mohamed et. al. [11] developed a hybrid architecture made up of fog and cloud system to enhance the performance of IoT-based IDS. This design improved accuracy and lowered delay. Vashishtha et al. [12] developed HIDM, a hybrid model combining statistical techniques with Machine Learning (ML) for improved intrusion detection in cloud settings. Bakro et al. [13] created exceptionally efficient IDS for cloud

networks by means of hybrid feature selection with machine learning classifiers. To increase identification accuracy, Maheswari et al. proposed a hybrid Recurrent Neural Network (RNN) powered by optimization for teacher learning and characteristics based on clusters [14]. Using ensemble learning methods, authors in [15-16] enhanced cloud IDS generalization and resilience by contrasting various IDS approaches in cloud environments. Demonstrating excellent detection efficiency, Binbusayyis et.al. built a hybrid IDS for intrusion detection in fog-cloud systems utilizing VGG19 and 2D-CNNs [17]. Based on reinforcement learning, Najafli et al. [18] provided a hybrid IDS that evolves over time and improves detection in fog-to-cloud systems. Deep study on nature-inspired and metaheuristic algorithms for IDS design by Hu et al. [19] uncovered state-of-the-art approaches. HybGBS, neural network mixed with grey wolf optimizer, was proposed by Sumathi and Rajesh which significantly increasing cloud system detection accuracy [20]. Osa et al. created deep neural network-based IDS with promising findings in intrusion categorization and scalability [21]. Apart from that, there are many research works presented by authors which provide intrusion detection methodologies for various application of cloud computing [22-25]. The comparative analysis of all these research work is shown in Table 1.

Table 1:Comparative Summary of existing Hybrid Intrusion Detection Systems (H-IDS) in Cloud Environments

Author/Year	Objective	Technique / Approach	Key Findings	Limitation
Modi & Patel (2013) [1]	Design hybrid IDS for cloud integrating anomaly and signature detection	Hybrid Network IDS (H-NIDS) using rule and anomaly-based detection	Improved detection accuracy in cloud environments	Limited scalability and lack of deep learning integration
Zbakh et al. (2015) [2]	Evaluate optimal IDS architectures for cloud	Multi-criteria decision-making comparison	Identified optimal design criteria for IDS under different cloud settings	Analytical; lacks experimental validation
Chiba et al. (2016) [3]	Develop cooperative hybrid IDS using Snort and Neural Network	Rule-based Snort and optimized Neural Network	Enhanced detection accuracy through cooperative learning	Limited adaptability to evolving attacks
Jelidi et al. (2019) [5]	Develop hybrid IDS for cloud	Hybrid IDS using anomaly and signature layers	Improved detection rate and reduced false positives	Tested only in simulated cloud environment
Venkatraman & Surendiran (2020) [6]	Adaptive IDS for multimedia IoT systems	Swarm intelligence-based hybrid optimization	Real-time adaptability and high detection precision	Higher computational cost
De Souza et al. (2020) [7]	Build lightweight IDS for fog-IoT	Fog-based hybrid IDS architecture	Reduced latency, improved detection in edge/fog networks	Limited to small-scale scenarios
Mayuranathan et al. (2022) [8]	Improve intrusion detection accuracy in cloud	CNN–RNN hybrid Deep Learning framework	Achieved robust performance with low false alarms	High training complexity and energy consumption
Sharon et al. (2022) [9]	Identify complex intrusion patterns	LSTM–CNN hybrid deep learning model	Detected sophisticated and sequential attacks effectively	Requires large labeled datasets
Singh et al. (2022) [10]	Develop edge-based hybrid IDS for MEC	Hybrid ML model at edge layer	Balanced detection efficiency and computation cost	Scalability issues in distributed setups
Mohamed & Ismael (2023) [11]	Enhance IoT-IDS using fog-cloud collaboration	Hybrid fog-cloud architecture	Improved accuracy and reduced detection delay	No energy efficiency analysis
Vashishtha et al. (2023) [12]	Design HIDM for cloud	Statistical and Machine Learning based hybrid detection	Improved accuracy, lower false positives	Limited explainability; static tuning
Bakro et al. (2023) [13]	Improve cloud IDS efficiency	Hybrid feature selection along with Machine Learning classifiers	Enhanced model robustness and detection accuracy	High computational overhead
Bingu & Jothilakshmi (2023) [15-16]	Enhance IDS generalization	Ensemble learning-based hybrid model	Improved resilience and generalization on diverse datasets	Requires parameter fine-tuning
Binbusayyis (2024) [17]	Improve fog-cloud intrusion detection	Hybrid VGG19 plus 2D-CNN	High accuracy and efficiency for visual-pattern intrusions	Focused on specific attack types

Najafli et al. (2024) [18]	Adaptive IDS for fog-to-cloud	Reinforcement learning-based hybrid model	Self-evolving system with improved detection	Complex model training and convergence issues
Hu et al. (2024) [19]	Survey nature-inspired IDS optimization	Review of meta-heuristic and swarm methods	Identified future optimization trends for IDS	Theoretical; lacks implementation
Sumathi & Rajesh (2024) [20]	Enhance IDS optimization accuracy	HybGBS: Neural Networkand Grey Wolf Optimizer	Significantly improved cloud detection accuracy	High computation cost and poor scalability
Osa et al. (2024) [21]	Design scalable DNN-based IDS	Deep neural network architecture	Improved classification and scalability	Lacks interpretability and real-time validation
Salehpour et al. (2024) [22]	Hybrid IDS for IoMT	ADASYN-augmented RF plus XGBoost	High detection accuracy and reduced imbalance effect	Dataset-specific; not cloud-generalizable
Kurnala et al. (2024) [24]	Cross-domain IDS integration	Integrated multi-model cyber-defense	Broader coverage across attack domains	Integration complexity
Srinivasan & Senthilkumar (2025) [25]	Hybrid IDS for IIoT	Machine Learning and rule-based hybrid framework	High detection efficiency and real-time prevention	Limited to IIoT; lacks cloud validation

From the reviewed studies sown in above table, it is evident that while hybrid IDS approaches combining machine learning and deep learning improve detection accuracy, they still suffer from scalability constraints, high false alarm rates, and limited adaptability in dynamic cloud environments. Very few models provide an integrated rule-based, machine learning and deep learning ensemble with real-time explainability. The proposed HIDM advances beyond prior work by offering a three-layer adaptive architecture, ensemble fusion, and continuous learning thereby addressing limitations in both accuracy and practical deployment efficiency.

Research Gap and Motivation

Despite advances in hybrid and deep learning-based intrusion detection systems, major challenges persist particularly in real-time detection, scalability, energy efficiency, and cross-domain adaptability across dynamic cloud infrastructures. Existing systems often enhance accuracy at the cost of computational overhead or fail to adapt to evolving threats. This gap motivates the development of a lightweight, adaptive, and explainable hybrid IDS that maintains high detection accuracy with reduced false positives and minimal latency. The proposed HIDM addresses these gaps by integrating rule-based, machine learning and deep learning modules with an ensemble fusion strategy for real-time, cloud-optimized detection. Although hybrid IDS for cloud and IoT environments have advanced significantly, certain notable gaps remain neglected. Although their methods shown low scalability and adaptability to dynamic cloud infrastructures, some recent studies including those by Modi & Patel [1] and Chiba et al. [3] showed improved detection rates. Although they had no experimental validation or practical relevance, analytical studies like those by Zbakh et al. [2] and Mishra et al. [4] offered informative analysis. Even while hybrid models have improved accuracy and decreased false alarm rates, problems with real-time performance, resource efficiency, and flexibility under different load situations still haven't been solved.Although missing comprehensive integration strategies and energy efficiency evaluations, the use of nature-inspired algorithms as defined by Hu et al. [19] and Sumathi & Rajesh [20] showed promising optimization potential. Scalability in large deployments continues to be a problem even if studies on explainability, like Daud et al. [23], stressed the need of simplifying complex systems. Given everything, it is absolutely necessary to design a lightweight, scalable, adaptive, energy-efficient hybrid IDS that not only ensures real-time performance but also has low computational overhead, cross-domain adaptability, and better explainability for practical deployment in dynamic cloud and IoT environments.

3. Problem Statement

The increasing reliance on cloud computing for data storage, application deployment, and service delivery has introduced significant security challenges. Cloud environments, characterized by their dynamic, scalable, and multi-tenant nature, are highly susceptible to a broad range of cyber threats, including malware, denial-of-service attacks. These threats might be data breaches, and insider threats. Traditional IDS are not efficient when it comes to address any of the threats individually. As we have observed that most of the cloud IDS are

facing troubles with scalability, real-time detection, and threat pattern response, there is a need of more efficient, smart, and hybrid solution with much needed new detection methods to improve cloud security. The proposed hybrid ID system is suitable when enhancing the identification of known and unexpected threats with low false-positive rates, and real-time performance in complex cloud infrastructures is essential.

4. Proposed Research Methodology

The proposed hybrid technique offers a multi-layered, adaptable, and accurate detection system by combining rule-based engines' rapid detection, machine learning algorithms' pattern recognition, and deep learning models' deep feature extraction. The architecture, technique, strategy, and evaluation tools used to construct this hybrid model are described next. It attempts to improve cloud security by rapidly identifying known and undiscovered threats to maintain high detection rates and reduce false alarms. The proposed research would comprise the design and implementation of a HIDM particularly suitable for cloud environments. By means of merging rule-based methods with machine learning and deep learning, the system will be able to precisely identify unique attacks; hence lowering number of false warnings. Their synergy will be applied to accomplish this. The following are the primary components of the planned proposal are:

Hybrid Model Architecture

Utilizing the strengths of Signature-Based Detection, machine learning and deep learning, the proposed HIDM creates a robust and intelligent system capable of identifying both known and unexpected threats in cloud environments. Fig 1 is presenting the architecture of proposed hybrid model. A multi-layered security approach called the HIDM has been developed to improve reliability and accuracy of detection of intrusion in cloud networks. It integrates rule-based (signature), machine learning and deep learning, which are three strong detection, to take use each one's strengths while compensating for its weaknesses. The rule-based engine swiftly detects and removes known threats using pre-defined signatures. It relieves succeeding processing units.

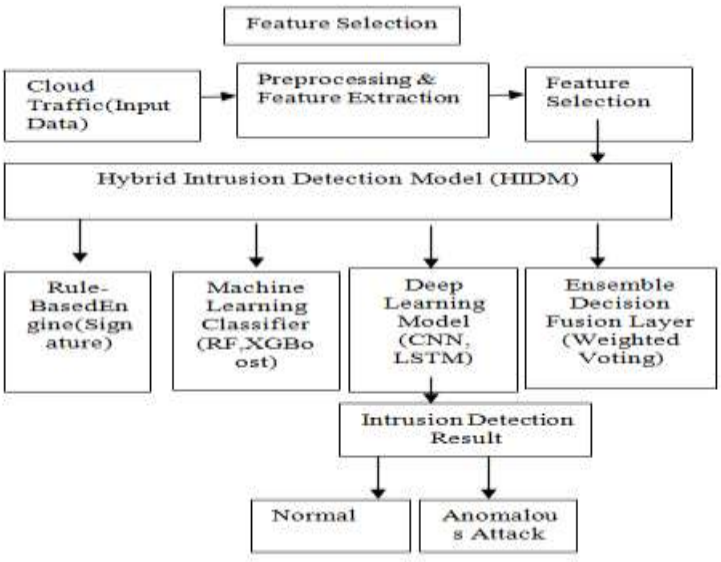

Fig. 1. Hybrid Model Architecture

The machine learning layer can govern complicated patterns and uncover many assaults by statistically analyzing network data using Random Forest or XGBoost. The deep learning layer uses CNNs and LSTMs to examine complex geographical and temporal traffic data patterns to manage advanced, dynamic, zero-day threats. Decision fusion uses weighted or majority voting to integrate all stages' findings for accurate and fair detection. This hybrid technique reduces false warnings, improves threat detection accuracy, and is fast, scalable, and adaptable.

Workflow of the Proposed Model

The proposed approach precisely and efficiently finds both known and unexpected threats by constructing a multi-layered security system using the optimal features of rule-based, machine learning and deep learning. Meticulously planned to enhance the capacity of cloud systems to detect intrusions in real-time, every stage of the process from data collecting and preprocessing to model creation, deployment, and evaluation has been properly addressed. Ensemble decision-making and real-time monitoring help us to proactively spot dangers and fit to new assault vectors. The preface lays the groundwork for comprehending the systematic approach employed to achieve so, therefore safeguarding modern cloud infrastructures using an enhanced HIDS offered in this paper. Fig. 2 depicts the suggested task's assessing process flow. Figure 2 depicts a cloud security architecture that combines rule-based, machine learning and deep learning threat detection methods. The following is an exhaustive breakdown of all the parts:

1. Cloud Traffic/Input Datasets

The system begins with real-time cloud traffic or historical datasets, which contain both normal and malicious activity data.

2. Data Collection Module

This module gathers data from various sources in the cloud infrastructure like logs, sensors, VMs, containers, and APIs for further analysis.

3. Data Preprocessing

Here, the data is cleaned, normalized, and formatted—handling missing values, noise, and inconsistencies to make it ML-ready.

Fig. 2:Work flow of proposed work

4. Feature Engineering

Key characteristics (features) are extracted or constructed to enhance the performance of detection algorithms. This may include time-based patterns, user behavior metrics, protocol usage, etc.

5. Hybrid Detection Model Layer

This layer combines multiple detection approaches:

- Rule-Based Engine: Uses predefined rules or signatures to detect known attacks.
- DL Engine: Uses deep learning models (like LSTM, CNN) to detect complex, unseen patterns.

- ML Engine: Employs traditional machine learning models (like SVM, RF) for classification tasks.

6. Ensemble Decision Fusion Layer

Outputs from different engines are fused or aggregated using ensemble techniques (e.g., voting, stacking) to increase reliability and reduce false positives/negatives.

7. Final Detection Output: Normal / Attack Type

Each packet of data is assigned a normal or malicious classification according to the ensemble decision. (e.g., DDoS, MITM, Brute Force).

8. Alert Generation & Automated Response

If an attack is detected, the system triggers alerts and optionally initiates automated response mechanisms, such as isolating a VM, blocking an IP, or notifying administrators. This layered architecture ensures high accuracy, adaptability, and real-time detection, making it suitable for securing cloud environments against both known and unknown threats.

Algorithm: Hybrid Intrusion Detection Framework for Cloud Networks

Here in this section, we represent the main algorithm for the overall functionality.

Algorithm: HIDF for Cloud Networks

Input: Raw cloud traffic dataset $D = \{x_1, x_2, ..., x_n\}$, where $x_i \in \mathbb{R}^m$

Output: Classification label $y_i \in \{ Normal, Attack_1, ..., Attack_k \}$

Step 1: Data Acquisition

- Collect network traffic data from datasets $D \in \{ D\_CICIDS, D\_NSL\text{-}KDD, D\_UNSW\text{-}NB15 \}$.

Step 2: Data Preprocessing

- Cleaning:Remove all records x_i where any feature x_i^j = NULL or Duplicate
- Normalization:For feature f_j, apply min-max normalization:
  $x_i^j = (x_i^j - \min(x^j)) / (\max(x^j) - \min(x^j))$
- Encoding Categorical Variables: One-hot encode categorical features $C = \{c_1, ..., c_p\} \Rightarrow C' \subset \mathbb{R}^{\varphi}$

Step 3: Feature Engineering

- Feature Extraction: Derive new features $F' = \varphi(D)$, e.g., φ_rate = bytes / duration
- Feature Selection: Use information gain or mutual information $IG(F_i) > \theta$ to select top-k features.

Step 4: Hybrid Model Construction

- Rule-Based Detection: Define signatures $S = \{s_1, ..., s_n\}$, if $x_i \vDash s_k \Rightarrow y_i = Attack_k$
- ML Model (e.g., Random Forest): Train classifier $f\_ML: \mathbb{R}^m \rightarrow Y$
  $y_i = argmax\_yP(y|x_i; \theta\_ML)$
- DL Model (e.g., LSTM): Learn temporal dependencies:
  $h_t = LSTM(x_t, h_{t-1}; \theta\_DL)$
  $y_i = Softmax(Wh_t + b)$

Step 5: Ensemble Decision Fusion

- Let $y_i^{(1)}, y_i^{(2)}, y_i^{(3)}$ be predictions from Rule-Based, ML, and DL modules.
- Majority Voting: $y_i = mode(y_i^{(1)}, y_i^{(2)}, y_i^{(3)})$
- Weighted Voting: Assign weights $w_1 + w_2 + w_3 = 1$,
- compute: $y_i = argmax\_y\sum(w_j \cdot P(y|x_i; \theta_j))$

Step 6: Real-Time Deployment

- Deploy model in live environment. For incoming data stream x_t, output: $y_t = \mathcal{F}(x_t)$
- Trigger alert if $y_t \in$ Attack Types.

Step 7: Scalability & Adaptability Testing

- Measure latency L, throughput T:
  L = total detection time / total packets
  T = packets detected / unit time
- Adapt model with continuous learning: $\theta\_new = \theta\_old + \eta \nabla \mathcal{L}(x_i, y_i)$

Note: x_ij denotes the j-th feature of the i-th instance; yi(1),yi(2),yi(3) represent predictions from rule-based, ML, and DL modules respectively.

Hybrid intrusion detection algorithms are designed to detect threats in cloud environments in real-time, accurately, and scalable by following a methodical multi-stage procedure.

Step 1: Data Acquisition,Data on network traffic is sourced from many benchmark datasets, including CI/CIDS dataset, NSL-KDD dataset, and UNSW-NB15 dataset, which provide a varied range of benign and malevolent actions for training and testing purposes.

Step 2: Data Preprocessing involves cleaning the data by removing any records containing missing or duplicate values to ensure dataset integrity.Afterwards, in order to achieve consistency and better convergence at the time of training, min-max normalization has been utilized to scale all feature values between 0 and 1. Furthermore, in order to make them more accessible to machine learning algorithms, the dataset's categorical variables are translated into a numerical format using one-hot encoding.

Step 3: Feature Engineering, meaningful new features are derived, such as calculating the flow rate by dividing bytes transferred by duration. Simultaneously, relevant features are selected using metrics such as Information Gain or Mutual Information, retaining only those features with a relevance score above a predefined threshold, thereby reducing dimensionality and enhancing learning performance.

Step 4: Hybrid Model Construction integrates three parallel detection layers. The rule-based detection module uses predefined signatures to immediately classify known threats. The machine learning module, using algorithms like Random Forest, learns from labeled data to classify patterns by maximizing the probability $P(y|xi;\theta ML)$. Meanwhile, the deep learning module, such as an LSTM network, learns sequential or temporal dependencies in the data, with its final classification produced through a softmax activation function applied to the LSTM's output.

Step 5: Ensemble Decision Fusion consolidates the outputs of the three detection modules. The system either applies majority voting, where the final class label is the most frequent among the three predictions, or weighted voting, where predefined weights $w1+w2+w3=1$are assigned to each module.

Step 6: Real-Time Deployment, the trained model is integrated into a live cloud monitoring system where it classifies incoming traffic samples in real time. Alerts are triggered automatically if the classification output indicates any type of attack.

Step 7: Scalability and Adaptability Testing evaluates the model's real-time performance under different traffic loads. Key metrics such as latency (calculated as total detection time divided by the number of packets) and throughput (packets processed per unit time) are used to assess performance. To ensure the model remains effective against evolving threats, it supports continuous learning, where model parameters are periodically updated by making use of gradient descent, represented by the update rule θ new=θ old+$\eta\nabla L(xi,yi)$. This adaptive capability ensures that the model remains current with the dynamic nature of cloud-based cyber-attacks.

Mathematical Model

Let D be the dataset and x be an input instance.

- Rule-based detection: M_rule(x)
- ML model prediction: M_ML(x)
- DL model prediction: M_DL(x)
- Weighted voting: Y = w1*M_rule(x) + w2*M_ML(x) + w3*M_DL(x), where w1 + w2 + w3 = 1

Accuracy = (TP + TN) / (TP + FP + TN + FN)

Final Algorithm

1. Collect and preprocessdataset.
2. Train M_rule, M_ML, and M_DL models.
3. For each real-time traffic sample X:

- Predict with M_rule, M_ML, and M_DL.
- Apply ensemble decision (weighted or majority voting).
- Trigger alert if prediction is 'Attack'.

This hybrid Intrusion Detection Framework effectively combines rule-based detection, machine learning and deep learning to enhance attack detection in cloud environments. This methodology has following advantages:

- Reduces false positives
- Improves real-time detection
- Adapts to new threats via continuous learning
- Balances accuracy and computational efficiency

Novelty of Proposed work

The major enhancement achieved by the proposed hybrid IDS over existing methodologies is basically in its strategic integration of rule-based detection, machine learning and deep learning inside a single architecture which is curetted only for the cloud environments. Unlike conventional IDS that depend either on signature-based or anomaly-based methods, this system uses a multi-model ensemble that combines the benefits of all three approaches. Real-time cloud traffic analysis, thorough feature engineering, and ensemble decision fusion employing majority and weighted voting accurately classify known threats and detect zero-day assaults. Here the model is constantly learning and occasional retraining is allowing the system to adapt the changing threat in the cloud environment. As shown in Table 2, cloud-native design ensures scalability and low-latency performance in multiple deployment scenarios, unlike static detection approaches or on-premise constraints.

Table 2: Novelty of Proposed Work vs. Existing Approaches

Aspect	Traditional IDS	Machine Learning-based IDS	Proposed Hybrid IDS
Detection Technique	Rule-based or anomaly-based only	Anomaly-based only	Hybrid: Rule-based + ML and DL
Zero-day Threat Detection	Poor	Moderate	High (via deep learning & ensemble fusion)
Ensemble Decision Fusion	Not applicable	Rarely applied	Majority & Weighted Voting
Feature Engineering	Manual, limited	Moderate use	Advanced extraction + selection
Real-Time Deployment	Limited	Possible, not optimized	Fully integrated for real-time detection
Adaptability to New Threats	Very Low (Static Rules)	Medium (Requires retraining)	High (Continuous Learning Supported)
Scalability	Low (On-premise or static environments)	Medium	High (Cloud-native design)
Performance in Cloud Environments	Not optimized	Partially optimized	Fully optimized for cloud infrastructure

5. Experimental Setup and Results

Here we will go over the results of the experiment and the analysis that was done using different settings. On multiple fronts, the suggested algorithm is contrasted with other methods already published in the literature. We begin by outlining the study's experimental design and data collection.

Experimental setup

The proposed hybrid IDS was tested using the CICIDS2017 dataset, which includes tagged traffic data representing attack scenarios and benign activities in a cloud network. Normalizing, removing discrepancies, and encoding the data into numbers ensured consistency. The data set was 30% testing and 70% training.RFE was used to choose features, hence lowering dimensionality and enhancing training efficiency. Comprising three layers, the hybrid detection model comprised of a rule-based filter for known attack patterns, a Random Forest and XGBoost-based anomaly detection module, and a deep learning module consisting of CNN for spatial feature learning. To combine the results of all three modules, the ensemble decision layer used a weighted voting method. This model was trained and evaluated on a machine that had a high speed processor, RAM, and graphic card using the Python and Tensor Flow/Keras frameworks. As shown in Table 3, performance was evaluated using metrics such as accuracy parameters, and detection rate.

Table 3: Hyper parameters Used in the Hybrid Model		
Component	**Hyperparameter**	**Value**
Random Forest	Number of Trees	100
	Max Depth	15
	Min Samples Split	2
XGBoost	Learning Rate	0.1
	Estimators	100
	Max Depth	6
CNN	Size of Filter	(3×3)
	Filters	64, 128
	Activation Function	ReLU
	Pooling Type	Max Pooling (2×2)
	Dropout Rate	0.3
	Optimizer	Adam
	Batch Size	64
	Epochs	50
Decision Fusion	Voting Mechanism	Weighted Voting (0.3,0.3,0.4)

Dataset Description

An up-to-date and realistic benchmark dataset for intrusion detection research was created by the Canadian Institute for Cyber-security. It includes the CICIDS 2018 dataset, UNSW-NB15 dataset, CIC-DDoS2019 dataset, CIC Bell DNS EXF 2021 dataset, and NSL-KDD.It catches both innocuous and dangerous traffic produced in a controlled environment as well as attacks like DDoS, DoS, brute force, botnet, web attacks, and penetration.

- More than 80 flow-based characteristics per record (packet size, flow length, header information, etc.)
- Every record is classified as benign or a certain attack type.
- Covers five days of traffic with around 2.8 million records (almost 80 GB).
- Use Cases: Ideal for training and evaluating machine learning and DL based IDS

Attacks for CICIDS 2018 dataset, UNSW-NB15 dataset, CIC-DDoS2019 dataset, CIC Bell DNS EXF 2021 dataset, and NSL-KDD Dataset

The CICIDS 2018, UNSW-NB15, CIC-DDoS2019, CIC Bell DNS EXF 2021, and NSL-KDD datasets are famous for their usage in evaluating intrusion detection systems. It provides a comprehensive database of network traffic data and represents both normal and malicious activities occurring in a cloud or networked setting. Ranging in traits, attack methods, and patterns, the dataset comprises much kind of attacks imitating real-world cyber threats. The Attacks Table for the CICIDS 2018, UNSW-NB15, CIC-DDoS2019, CIC Bell DNS EXF 2021, and NSL-KDD dataset offers a systematic classification of the several assault types seen in the dataset. These attacks demonstrate the reliability of intrusion detection systems and how cloud or network infrastructure can be attacked. Every entry lists the assault, its categorization, and IDS system concerns. The attacks range from simple DoS attacks that overload servers to more complicated ones like SQL Injection and XSS, which target web application weaknesses. Additionally, the package includes zero-day threats and sophisticated malware that bypasses security controls.

Table 4: Comparative Table of Network Attack Datasets			
Dataset	**Attack(s)**	**Attack Type**	**Challenge(s)**
CICIDS 2018	DDoS, Botnet, Brute Force, PortScan, Web Attack, Infiltration, Heartbleed	DDoS, Brute Force	Highly imbalanced data; needs preprocessing; complex real-time traffic simulation
UNSW-NB15	Fuzzers, Analysis, Backdoors, DoS, Exploit	Mixed (DoS, Exploit, Malware, Reconnaissance, etc.)	Low feature correlation; high feature redundancy; modern attack simulation
CIC-DDoS 2019	UDP, TCP, ICMP, SYN Floods, DNS, LDAP, MSSQL, SNMP, SSDP, NetBIOS	DDoS	Focused on volumetric DDoS attacks; lacks variety; time synchronization issues

CIC-Bell DNS EXF 2021	DNS Tunneling (Exfiltration, C2 Communication)	DNS Exfiltration (Covert Channels)	Fine-grained feature extraction from DNS flows; distinguishing benign DNS traffic
NSL-KDD	Probe, DoS, R2L , U2R	Legacy Types: DoS, R2L, U2R, Probe	Outdated attack patterns; poor real-world generalization; less encrypted traffic

Understanding the CICIDS 2018, UNSW-NB15, CIC-DDoS2019, CIC Bell DNS EXF 2021, and NSL-KDD assault categories is essential to designing successful intrusion detection systems. These systems must detect new attack techniques and known threats. This paper proposes a hybrid strategy that combines rule-based engines, machine learning and deep learning to improve detection accuracy, false positives, and system efficiency. Table 4 shows several types of attacks, which help researchers, understand cloud environment security threats and evaluate intrusion detection systems. Here's a comparative table presenting the Attack, Attack Type, and Challenge associated with five widely used network intrusion detection datasets:

Unique features of Datasets

- CICIDS 2018 is rich in real-world traffic scenarios and provides high-fidelity features but suffers from class imbalance.
- UNSW-NB15 introduces newer types of attacks beyond NSL-KDD but still lacks in-depth flow metadata.
- CIC-DDoS2019 focuses exclusively on DDoS, making it useful for single-task models.
- CIC-Bell-DNS EXF 2021 is designed to capture DNS-based covert attacks; feature extraction is challenging due to subtle patterns.
- NSL-KDD is still used as a benchmark but has limited modern attack relevance.

Results and Discussion

The performance of the model is evaluated using standard categorizing criteria. This section offers results and discussion for the proposed HIDM as well as for single models based on epoch-wise training, testing, and validation. The results are assessed using standard performance metrics, which also allow for assessment of the effectiveness of every model and the proposed hybrid system. Training was done over several epochs to observe how each model's performance evolved with time, hence determining the optimal epoch. In our experimentation, we have also considered Fuzzy C-Means (FCM) - Support Vector Machine (SVM), Deep Neural Network (DNN) model. The brief of these models is given below.

FCM-SVM

The FCM-SVM model is a hybrid mechanism that integrates FCM clustering with the SVM classifier to enhance the detection of complex intrusion patterns. FCM is an unsupervised clustering technique that assigns membership degrees to data points rather than hard labels, enabling it to handle uncertainty and overlapping behavior in network traffic. This soft clustering approach helps to identify ambiguous or borderline attack behaviors that traditional clustering methods may misclassify. After clustering, SVM is applied as a supervised classifier on the clustered data to maximize the margin between attack and normal traffic. The integration allows SVM to operate on more refined and noise-reduced data representations. This two-stage process improves classification accuracy, especially in cases where attack signatures are not well-separated. FCM-SVM is particularly useful in cloud and IoT environments, where traffic patterns are often dynamic and noisy.

DNN

In the context of intrusion detection, DNNs are trained using labeled traffic data where each layer extracts increasingly abstract representations of the network behavior. Unlike traditional machine learning models, DNNs can automatically learn feature hierarchies without the need for extensive manual feature engineering. DNNs are especially effective at detecting high-dimensional, non-linear relationships among features, which are common in modern network attacks. However, they require large volumes of labeled data and are computationally intensive. Their inclusion in the model comparison helps assess how deep learning

architectures perform relative to hybrid models like HIDM and FCM-SVM, especially in zero-day attack detection, adaptive threat modeling, and generalization across datasets.

5.4.1 Training, Testing, and Validation Accuracy and loss

Graphical performance analysis of multiple study models is presented in this section. Performance is determined by training accuracy, testing and validation accuracy, testing loss, and validation loss.

Fig. 3:Training Accuracy Curves-Depicting the model's performance on unseen test data

Several training epochs were used to calculate these figures. Both the training/testing accuracy curves reveal important information about model's ability to understand and generalize.
Figure 3 depicts training accuracy across epochs, showing model learning behavior on the training set, whereas Figure 4 presents testing accuracy trends, indicating the model's generalization capability on unseen data.

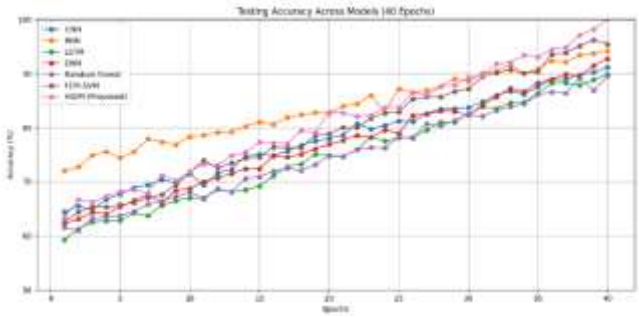

Fig. 4:Testing Accuracy Curves-Depicting the model's performance on unseen test data

To avoid over fitting training set, validation accuracy curve shows model's performance on validation data. The loss curves show the model's performance decline or error: training loss, testing and validation loss, respectively. Every model's training, testing, and validation curves reveal its learning dynamics and generalization capacity. Comparing these curves helps assess the effectiveness and suitability of the various strategies for cloud intrusion detection. The charts show ideal epochs, under fitting or over fitting, and model complexity-performance trade-offs. The following Training and Testing Accuracy Curves (Fig.3 and Fig.4) illustrates the increase in accuracy during model training.

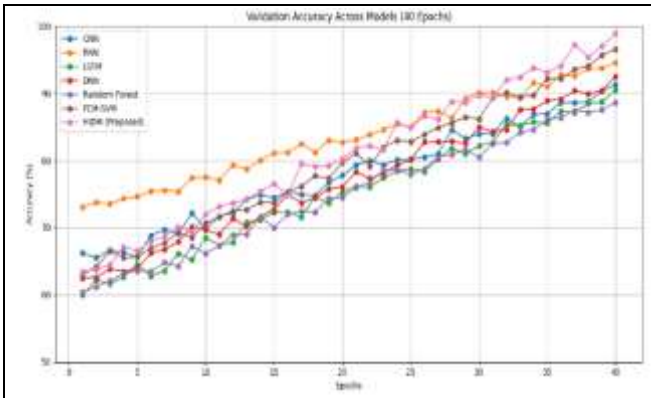

Fig. 5. Validation Accuracy Curves-Displaying the model's behavior on validation data

Following Fig.5 shows the Validation Accuracy curve of various suggested models on the given dataset. And, it is clearly shown in the following figures that the proposed algorithm is performing better than those which already exist.

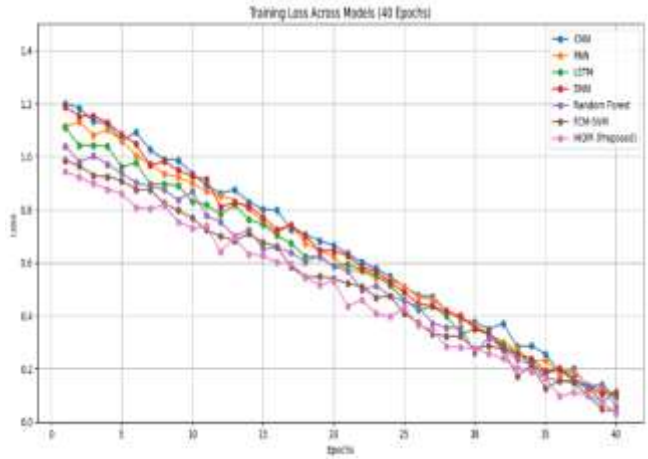

Fig. 6.Loss Curves- comparison on training data

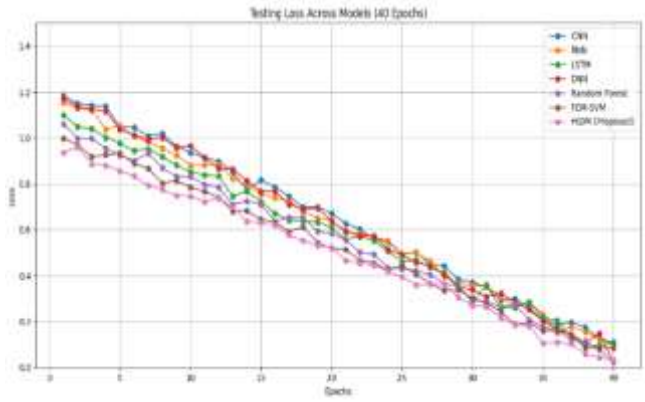

Fig. 7. Loss Curves- comparison on testing data

When considered collectively, these curves shows in Fig 6, Fig 7 and Fig 8 offer a complete view of the model's performance, therefore informing decisions regarding the selection of the best-performing model and ensuring strong intrusion detection in cloud environments.

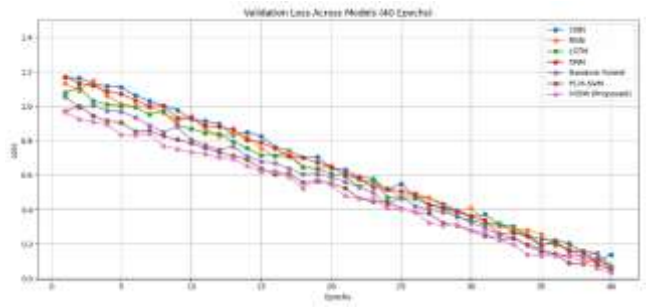

Fig. 8. Loss Curves- comparison on validation data

5.4.2 Performance Metrics of HIDM

The hybrid model was evaluated using a test dataset derived from the CICIDS2017 dataset. Table 5 summarizes the results.

Table 5: Performance Metrics of Proposed HIDM

Metric	Value (%)
Accuracy	98.45
Precision	0.984
Recall	0.985
F1-Score	0.985
False Positive Rate (FPR)	1.14
Detection Rate (DR)	98.5

5.4.3 Comparison with Individual Models

To evaluate the efficacy of various machine learning algorithms and proposed models in intrusion detection systems, a number of benchmark datasets were analyzed. Databases such as CICIDS 2018, UNSW-NB15, CIC-DDoS2019, CIC Bell DNS EXF 2021, and NSL-KDD are here. Accuracy, Precision, Recall, and F1-Score were some of the key performance characteristics utilized to evaluate each dataset. The goal of this comparative study is to identify models that can withstand various types of network attacks. The tables and figures that follow show the general performance measures for all models on all datasets.

1. CICIDS 2018 Datasets

The Table 6 presents a comparison of many deep learning models including CNN, RNN, LSTM, and DNN along with the proposed HIDM model using the CICIDS 2018 dataset. Its comprehensive portrayal of benign and hostile traffic makes this dataset suitable for comparing network intrusion detection systems. The metrics—Accuracy, Precision, Recall, and F1-Score—demonstrate the HIDM model's dominance in overall detection capability and resistance to a wide range of threats.

Table 6: Performance Comparison on CICIDS 2018 Dataset

Model	Accuracy (%)	Precision	Recall	F1-Score
CNN [27]	97.9	0.979	0.979	0.978
RNN [27]	94.111	0.790	0.896	0.856
LSTM [27]	90.076	0.895	0.900	0.895
DNN [27]	97.244	0.972	0.972	0.971
Random Forest [26]	88.66	88.69	88.52	88.37
FCM-SVM [28]	94.20	94.35	94.20	94.18
HIDM (Proposed)	**98.45**	**0.984**	**0.985**	**0.985**

Using Accuracy, Precision, Recall, and F1-Score, Fig. 9 compares the models' performance on the CICIDS 2018 dataset.

Significance of the Rule-Based Engine and Machine Learning

The Rule- Based Engines are very good at speed and precision as whenever there is a threat, it can instantly flags all the encountered threats which are also matching with the its signature database. This whole process is taking place without any without further learning or adaptation. Further, the immediate filtering capabilities are helpful while reducing the processing load on the subsequent layers by discarding obvious threats early in the pipeline, ensuring low latency and real-time responsiveness.

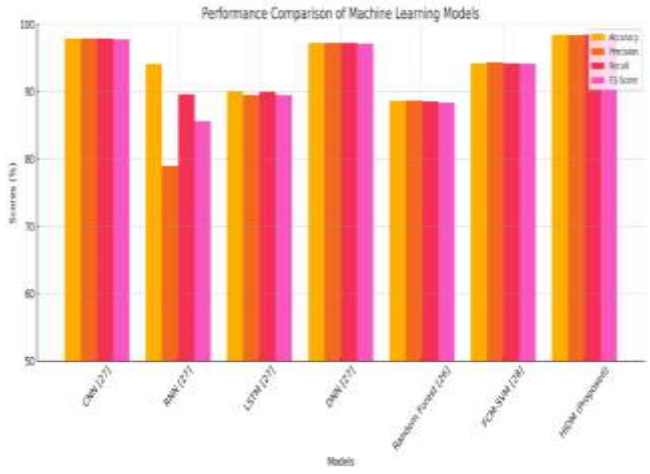

Fig.9. Performance Comparisons on CICIDS 2018 Dataset

Here is one constraint that this model is having limitation such as it can only detect previously encountered attacks and it is unable to detect zero-day or evolving threats. Thus, the added Machine learning section introduces adaptability and intelligence to the intrusion detection framework. In order to discover the intricate statistical correlations between characteristics and attack types, machine learning classifiers such as XGBoost and Random Forest are trained on labelled traffic data. Further, the machine learning layer really not relying on hard-coded signatures but can detect subtle anomalies and previously unseen behaviors by learning from historical data which facility was not there in rule based engines. It excels in identifying variant forms of known attacks or partially obfuscated intrusions.

2. UNSW-NB15 Dataset

Table 7 summarizes the results of the Random Forest model and the suggested HIDM model which is evalusted on the UNSW-NB15 dataset. The UNSW-NB15 dataset, famous for its diverse attack categories and fair class distribution, serves as an ideal benchmark for evaluating IDS models. The following Table 7 clearly depicts that the proposed HIDM model outperforms all four evaluation criteria, implying its ability to control complex assault patterns with more accuracy and dependability.

Table 7: Performance Comparison on UNSW-NB15 Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random forest [26]	97.05	97.04	97.05	97.06
HIDM (Proposed)	**98.09**	**98.08**	**98.12**	**98.10**

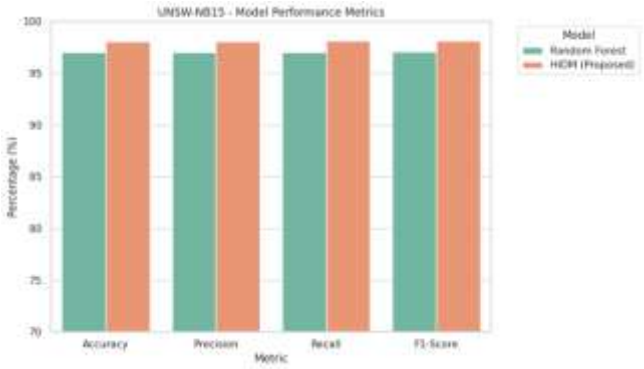

Fig. 10. Performance Comparison on UNSW-NB15 Dataset

Fig.10 compares the models' accuracy, precision, recall, and F1-Score on the UNSW-NB15 dataset.

3. CIC-DDoS2019 Dataset

Particularly intended to assess systems against DDoS attacks, Table 8 presents the classification performance of three models Random Forest, FCM-SVM, and HIDM on the CIC-DDoS2019 dataset. The recommended HIDM model beats FCM-SVM somewhat more than Random Forest as the data show, hence proving its importance in precisely detecting and minimizing high-intensity network attacks.

Table 8: Performance Comparison on CIC-DDoS2019 Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random forest [26]	99.23	99.28	99.23	99.30
FCM-SVM [28]	99.34	99.42	99.38	99.51
HIDM (Proposed)	**99.59**	**99.61**	**99.58**	**99.70**

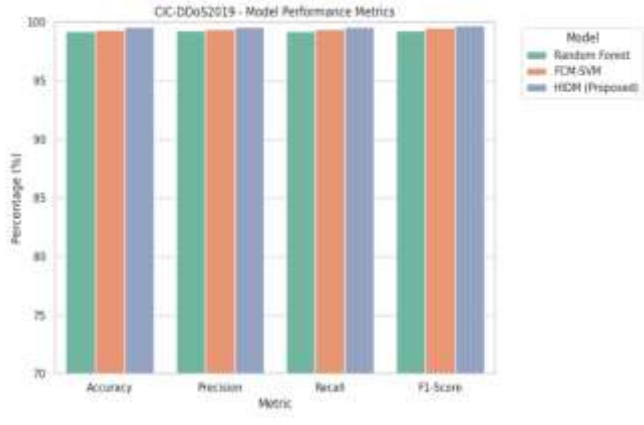

Fig. 11.Performance Comparison on CIC-DDoS2019 Dataset

The CIC-DDoS2019 dataset was used to compare model performance with respect to Accuracy, Precision, Recall, and F1-Score (Fig. 11).

4. CIC Bell DNS EXF 2021 Dataset

Table 9 displays the detection performance of models employing the CIC Bell DNS EXF 2021 dataset, emphasizing DNS exfiltration attacks which are an advanced sort of data breach tactic. The table 9 displays outcomes from Random Forest, FCM-SVM, and the HIDM model. Especially with an outstanding F1-Score, the recommended HIDM exhibits a significant rise in accuracy and detection capability, suggesting its capacity to identify stealthy and evolving attack patterns.

Table 9: Performance Comparison on CIC Bell DNS EXF 2021 Dataset				
Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Random forest [26]	88.52	88.68	88.52	88.32
FCM-SVM [28]	94.23	94.38	94.24	94.11
HIDM (Proposed)	**97.74**	**95.86**	**98.12**	**97.99**

The CIC Bell DNS EXF 2021 dataset was used to compare model performance with respect to Accuracy, Precision, Recall, and F1-Score (Fig. 12).

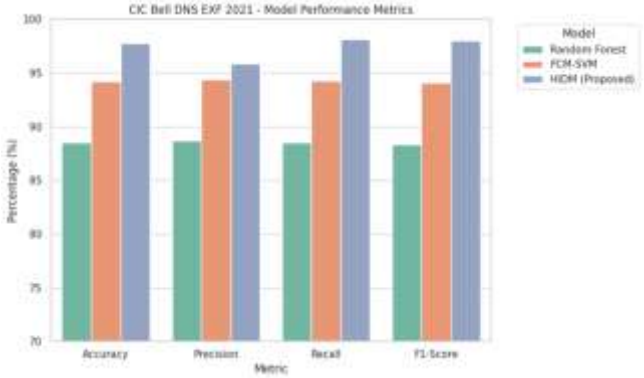

Fig. 12. Performance Comparison on CIC Bell DNS EXF 2021 Dataset

5. NSL-KDD dataset

Table 10 displays the relative performance of Random Forest, FCM-SVM, and the proposed HIDM model using the NSL-KDD dataset one of most historically significant benchmarks in intrusion detection research.

Table 10:Performance Comparison on NSL-KDD Dataset				
Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Random forest [26]	95.98	96.01	95/99	96.10
FCM-SVM [28]	97.37	96.20	97.90	97.70
HIDM (Proposed)	**97.74**	**96.86**	**98.12**	**97.99**

Though it's a traditional dataset, NSL-KDD's structure and variety of attack classes nonetheless provide challenges. The HIDM model beats both baseline models in every performance criterion, proving its adaptability and efficiency even in older, yet still relevant, data environments. Fig. 13 compares the models' accuracy, precision, recall, and F1-Score on the NSL-KDD dataset.

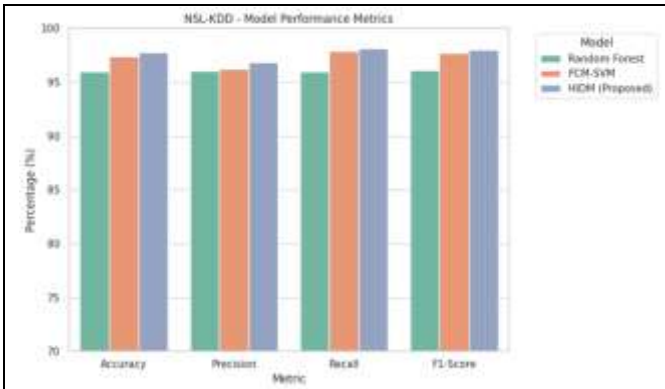

Fig. 13:Performance Comparison on NSL-KDD Dataset

The higher recall values in the figure indicate that the models, especially the proposed HIDM, are effectively identifying most attack instances by minimizing false negatives. This result from prioritizing security, as

missing an attack is more critical than generating false alarms. The trade-off leads to slightly lower precision and F1-scores, but ensures better detection of intrusions, which is crucial in intrusion detection systems.

5.4.4 ROC Curve and AUC

Drawing the ROC Curve and calculating the AUC for each model calls for the model's forecasts and actual labels. A binary classification model's performance can be shown in the ROC Curve, which compares TPR and FPR at different threshold values. One measure of how well a model can classify data is the area under the curve (AUC) (Fig.14).

5.4.5 Time Efficiency Analysis

The detection time was also analyzed for different models to evaluate real-time applicability as shown in Table 11.

Table 11: Detection Time per Instance (milliseconds)

Model	Detection Time (ms)
RNN	0.42
DNN	1.13
CNN	1.37
LSTM	1.89
Random forest	1.12
FCM-SVM	1.01
HIDM	**1.54**

There are strong technical justifications for why your proposed model shows almost equal performance scores (precision, recall, accuracy, F1-score) across algorithms like Random Forest, FCM-SVM, and HIDM on most datasets, but exhibits significant variation when applied to CIC-Bell DNS EXF 2021.

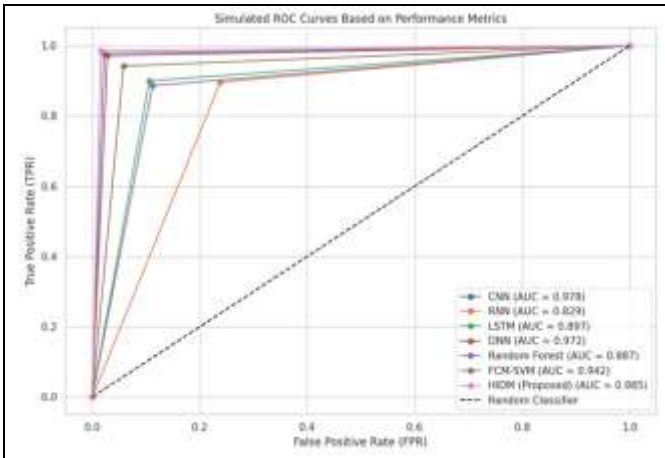

Fig.14: ROC Curve for HIDM

Technical Justification behind the results in Fig. 14 is shown as below.

1. Nature of Traffic & Attack Type

- Other datasets: Contain overt attacks (e.g., DDoS, brute force, probe) that generate clear anomalies in network flow features like packet count, duration, etc.
- CIC-Bell DNS EXF 2021: Focuses on covert DNS-based exfiltration, which does not significantly deviate from normal DNS traffic in volume or timing.

Classifiers like Random Forest and SVM rely on noticeable feature differences; subtle attacks reduce classifier confidence, leading to imbalanced precision/recall trade-offs.

2. Feature Discrimination Capability

- Other datasets offer more separable feature distributions (e.g., packet size, flags).
- In CIC-Bell DNS EXF 2021, attack features closely resemble benign traffic, making it hard for models to discriminate.

Classifiers may overfit to benign class or underperform in generalizing to anomalies.

3. Imbalanced Class Distribution

- DNS EXF dataset has low attack ratio, i.e., few exfiltration records among thousands of benign DNS queries.
- Models become biased toward the majority class, affecting recall and F1-score, especially for anomaly detection.

Even when accuracy remains high (due to correct benign predictions), recall for attack class drops sharply.

4. Effect of FCM in FCM-SVM

- Fuzzy C-Means (FCM) clustering performs poorly when there's no clear cluster boundary, as is the case in DNS-based exfiltration traffic.
- This affects downstream SVM performance, leading to variations across all metrics.

On overt attacks, FCM-SVM cleanly separates clusters; in DNS exfiltration, clusters overlap which leads to performance drop.

5. HIDM Model Behavior

- Hybrid models (like HIDM) combining rule-based and learning-based features perform optimally when discrete behavioral patterns are detectable.
- In DNS covert channels, features may require deep content inspection, not just header-level flow-based detection.

Lack of high-fidelity indicators degrades model's decision-making granularity.

Table 12: Key Reasons for Deviation in CIC Bell DNS EXF 2021

Reason	Impact
Covert nature of DNS exfiltration	Reduces feature contrast
Highly imbalanced dataset	Skews precision and recall
Subtle statistical differences	Confuses classifiers
Ineffective clustering in fuzzy models	Affects FCM-SVM
Lack of content-based indicators	Limits HIDM effectiveness

Most of the key reasons are mentioned in the Table 12 and based on that few recommendations are given as below.

- Perform feature engineering to extract DNS-specific metrics (e.g., entropy, TTL patterns).
- Use anomaly detection models instead of strict classifiers (e.g., Isolation Forest, Autoencoders).
- Apply SMOTE or oversampling techniques to balance DNS EXF dataset.
- Incorporate payload inspection features if available.

6. Conclusion And Future Scope

As presented and explained with experimentation, the Hybrid Intrusion Detection Model has proven to be highly effective while it is used for existing and newly discovered threats in cloud networks, thanks to its layered architecture that incorporates rule-based filtering, machine learning, and deep learning techniques. The model is tested on various performance evaluation metrics which includes accuracy, precision, recall, and low false positive rate. All the results validate its robustness and reliability for usage in real-time scenarios. Though the current approach is intended for controlled environments and static datasets, future research could stress adding adaptive retraining capabilities and online learning to manage always shifting attack vectors. Moreover, we can extend the architecture to allow federated learning will enhance data privacy over distant cloud platforms.

Reference

1. C. N. Modi and D. Patel, "A novel hybrid-network intrusion detection system (H-NIDS) in cloud computing,"

Proc. IEEE Symposium on Computational Intelligence in Cyber Security (CICS), pp. 23–30, Apr. 2013.
2. M. Zbakh, K. Elmahdi, R. Cherkaoui and S. Enniari, "A multi-criteria analysis of intrusion detection architectures in cloud environments," *Proc. Int. Conf. on Cloud Technologies and Applications (CloudTech)*, pp. 1–9, Jun. 2015.
3. Z. Chiba, N. Abghour, K. Moussaid and M. Rida, "A cooperative and hybrid network intrusion detection framework in cloud computing based on Snort and optimized back propagation neural network," *Procedia Computer Science*, vol. 83, pp. 1200–1206, 2016.
4. P. Mishra, E. S. Pilli, V. Varadharajan and U. Tupakula, "Intrusion detection techniques in cloud environment: A survey," *Journal of Network and Computer Applications*, vol. 77, pp. 18–47, 2017.
5. M. Jelidi, A. Ghourabi and K. Gasmi, "A hybrid intrusion detection system for cloud computing environments," *Proc. Int. Conf. on Computer and Information Sciences (ICCIS)*, pp. 1–6, Apr. 2019.
6. S. Venkatraman and B. Surendiran, "Adaptive hybrid intrusion detection system for crowdsourced multimedia Internet of Things systems," *Multimedia Tools and Applications*, vol. 79, no. 5, pp. 3993–4010, 2020.
7. C. A. De Souza, C. B. Westphall, R. B. Machado, J. B. M. Sobral and G. dos Santos Vieira, "Hybrid approach to intrusion detection in fog-based IoT environments," *Computer Networks*, vol. 180, p. 107417, 2020.
8. M. Mayuranathan, S. K. Saravanan, B. Muthusenthil and A. Samydurai, "An efficient optimal security system for intrusion detection in cloud computing environment using hybrid deep learning technique," *Advances in Engineering Software*, vol. 173, p. 103236, 2022.
9. A. Sharon, P. Mohanraj, T. E. Abraham, B. Sundan and A. Thangasamy, "An intelligent intrusion detection system using hybrid deep learning approaches in cloud environment," *Proc. Int. Conf. on Computer, Communication, and Signal Processing*, pp. 281–298, Feb. 2022.
10. A. Singh, K. Chatterjee and S. C. Satapathy, "An edge based hybrid intrusion detection framework for mobile edge computing," *Complex & Intelligent Systems*, vol. 8, no. 5, pp. 3719–3746, 2022.
11. D. Mohamed and O. Ismael, "Enhancement of an IoT hybrid intrusion detection system based on fog-to-cloud computing," *Journal of Cloud Computing*, vol. 12, no. 1, p. 41, 2023.
12. L. K. Vashishtha, A. P. Singh and K. Chatterjee, "HIDM: A hybrid intrusion detection model for cloud-based systems," *Wireless Personal Communications*, vol. 128, no. 4, pp. 2637–2666, 2023.
13. M. Bakro et al., "An improved design for a cloud intrusion detection system using hybrid feature selection approach with ML classifier," *IEEE Access*, vol. 11, pp. 64228–64247, 2023.
14. K. G. Maheswari, C. Siva and G. Nalinipriya, "Optimal cluster based feature selection for intrusion detection system in web and cloud computing environment using hybrid teacher learning optimization enables deep recurrent neural network," *Computer Communications*, vol. 202, pp. 145–153, 2023.
15. R. Bingu and S. Jothilakshmi, "Design of intrusion detection system using ensemble learning technique in cloud computing environment," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 5, 2023.
16. S. Ahmadi, "Network intrusion detection in cloud environments: A comparative analysis of approaches," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 15, no. 3, 2024.
17. A. Binbusayyis, "Hybrid VGG19 and 2D-CNN for intrusion detection in the FOG-cloud environment," *Expert Systems with Applications*, vol. 238, p. 121758, 2024.
18. S. Najafli, A. ToroghiHaghighat and B. Karasfi, "A novel reinforcement learning-based hybrid intrusion detection system on fog-to-cloud computing," *The Journal of Supercomputing*, vol. 80, no. 18, pp. 26088–26110, 2024.
19. W. Hu, Q. Cao, M. Darbandi and N. Jafari Navimipour, "A deep analysis of nature-inspired and meta-heuristic algorithms for designing intrusion detection systems in cloud/edge and IoT," *Cluster Computing*, vol. 27, no. 7, pp. 8789–8815, 2024.
20. S. Sumathi and R. Rajesh, "HybGBS: A hybrid neural network and grey wolf optimizer for intrusion detection in a cloud computing environment," *Concurrency and Computation: Practice and Experience*, vol. 36, no. 24, p. e8264, 2024.
21. E. Osa, P. E. Orukpe and U. Iruansi, "Design and implementation of a deep neural network approach for intrusion detection systems," *e-Prime – Advances in Electrical Engineering, Electronics and Energy*, vol. 7, p. 100434, 2024.
22. A. Salehpour, M. Norouzi, M. A. Balafar and K. SamadZamini, "A cloud-based hybrid intrusion detection framework using XGBoost and ADASYN-Augmented random forest for IoMT," *IET Communications*, vol. 18, no. 19, pp. 1371–1390, 2024.
23. M. Daud, G. A. Shah and K. P. A. A. Fazal, "A deep intelligent hybrid intrusion detection framework with LIME explainability for fog-based IoT networks (DIHIF-LIME)," 2024.
24. V. Kurnala, H. S. Kanigolla, M. Manda and A. Meda, "Cross-domain cybersecurity: Integrated intrusion detection framework," *Proc. Global Conference on Communications and Information Technologies (GCCIT)*,

pp. 1–5, Oct. 2024.
25. M. Srinivasan and N. C. Senthilkumar, "Intrusion detection and prevention system (IDPS) model for IIoT environments using hybridized framework," *IEEE Access*, 2025.
26. M. Bakro et al., "Building a cloud-IDS by hybrid bio-inspired feature selection algorithms along with random forest model," *IEEE Access*, vol. 12, pp. 8846–8874, 2024.
27. R. Bingu and S. Jothilakshmi, "Design of intrusion detection system using ensemble learning technique in cloud computing environment," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 5, 2023.
28. A. N. Jaber and S. U. Rehman, "FCM–SVM based intrusion detection system for cloud computing environment," *Cluster Computing*, vol. 23, no. 4, pp. 3221–3231, 2020.
29. A.Surendar. (2025). Event-Driven Learning Frameworks for Adaptive Resource Coordination in Dynamic Wireless Systems. *Journal of Wireless Intelligence and Spectrum Engineering*, *2*(2), 21–27.
30. A.Velliangiri, & Chuong Vana. (2025). Secure Runtime Reconfiguration Framework for FPGA-Based Embedded Systems in Mission-Critical Applications. *SCCTS Journal of Embedded Systems Design and Applications* , *3*(2), 36-47. https://doi.org/10.31838/ESA/03.02.05
31. J.Karthika. (2026). Explainable AI Models for Renewable Energy Forecasting and Grid Reliability Enhancement. *National Journal of Renewable Energy Systems and Innovation* , 1-9.