RESEARCH PAPER

A Blockchain-Based Framework for Distributed Denial-of-Service (DDoS) Mitigation in Cloud Environments: A Machine Learning and Trust-Driven Approach

Shilpa Bhatia[1], Ramesh Chandra Sahoo[2], Arvind Kumar[3]
[1,2] *School of Engineering and Technology, Manav Rachna International Institute of Research and Studies, Haryana, India* Shilpa33bhatia@gmail.com, rcsahoo.set@mriu.edu.in
[3]*School of Computer Applications, Manav Rachna International Institute of Research and Studies, Haryana, India,* Akdangi@gmail.com

Abstract: *The onset of mass dependence on cloud-based services has rendered it as a prime target for Distributed Denial-of-Service (DDoS) attacks, especially for critical applications. The most common way of attack is to send across tons of malicious traffic which eventually makes the system unavailable for genuine use. A potential solution to mitigate this issue has been proposed in this paper. A novel framework which runs on dual engine of supervised machine learning and blockchain which takes care of the trust management to stand against the DDoS attacks against cloud environments. It works on the logic of recovery which is embedded in smart contracts that essentially allows for trust re-evaluation and enables the system to self-correct itself and continuously evolve. Experiments were conducted using simulated attack scenarios with both volumetric and application-layer characteristics. The blockchain layer is supported by underlying cryptographic mechanisms that ensure secure validation, authenticated participation, and tamper- resistant record keeping. The framework achieved a detection accuracy of* ***96.4%****, a packet drop rate of* ***91.5%****, reduced average latency to* ***140 ms****, minimised system downtime to* ***under 2%****, and decreased CPU overhead by approximately* ***40%*** *compared to conventional static approaches. Evaluation metrics such as precision (****94.6%****), recall (****95.2%****), and F1-score (****94.9%****) further validate the reliability of the model. The use of blockchain in this model helps in making it a reliable system that is resilient and adaptable in environments that are hostile.*

Keywords: *Blockchain Security, Cryptography, Cloud Computing, DDoS Mitigation, Machine Learning Smart Contracts*
How to cite this article: *Bhatia S, Sahoo RC, Kumar A. A Blockchain-Based Framework for Distributed Denial-of-Service (DDoS) Mitigation in Cloud Environments: A Machine Learning and Trust-Driven Approach. Int J Drug Deliv Technol. 2026;16(12s): 583-593. DOI: 10.25258/ijddt.16.12s.71.*

1. Introduction

The introduction of cloud computing to the world has completely remodeled the current digital landscape, which has led to the enablement of on-demand services, adaptive scaling, and cost-effective management of resources. The maintenance of cloud services in terms of availability and integrity has now become extremely important because there has been a sudden shift and dependence of businesses and users on cloud infrastructures for hosting their key applications [1]. However, this shift comes with its own drawbacks in terms of a massive amount of attention from individuals involved in cybercrime, especially through DDoS attacks. The attackers aim to overwhelm the target system with an enormous amount of malicious traffic in order to bring down its channels that are secure channels, which results in complete exhaustion of computational and network resources and eventually renders the service to be unavailable and becomes inaccessible for those users who are typically authentic [2].

DDoS mitigation has various components to it; however, there are some conventional mechanisms which include techniques like firewalls, blacklisting, intrusion detection systems (IDS), and rate limiting techniques. These methods do provide some layer of protection, but they only remain superficial and have been seen to eventually fall apart once a high-capacity intrusion is detected, and the patterns of attack are mostly sly and unobtrusive [3]. There are various reasons for these shortcomings, some of which are that the decision-making process is centralised, occurrences of false positives are extremely high, and the system is not adaptive in real-time [4]. Moreover, the accountability and credible transparency are often lacking in these systems, and these systems are highly reactive in nature too, which creates a requirement for manual intervention.

The current technologies, however, such as blockchain and ML, can be excessively leveraged to address these challenges with the number of tools and resources they present to our disposal. The characteristic of the ML

model to be able to learn patterns can be employed to learn the pattern of the traffic and detect anomalies in real time [5]. This provides the system with novel vectors which create a defence mechanism in order to tackle the attacks. Blockchain, which is known for its decentralised, tamper-proof ledger, offers complementary strengths such as trustless consensus, audit trails that are immutable, and programmable automation via smart contracts. The addition of blockchain on top of ML essentially creates a dual-powered system which has the capability of being both an intelligent decision-maker and a system that provides a transparent and secure execution with utmost control over it.

1.1. Motivation

The continuous evolution and expansion of platforms for cloud computing have resulted in a corresponding increase in threats with regard to the security of the system, especially with DDoS attacks, which have emerged as one of the most devastating forms that are highly prevalent. The effect of these attacks is not only limited to the disruption in the continuity of the services, rather it also brings down the trust built within the user in the system, which consequently leads to the incurring of huge financial losses. The characteristics of the present security mechanism infrastructure, that is, its reactive nature, decision-making process that is centralised and a lack of adaptive intelligence, are insufficient to protect them. This leads to the need of having a security mechanism which is not just intelligent and autonomous but is also decentralised, which can efficiently respond to the threat in real time. By integrating the pattern recognition ability of ML and transparency and automation offered by the blockchain, we are motivated to create a robust system that overcomes existing limitations in DDoS mitigation.

1.2. Contributions

A novel framework of the system is being put forward by this paper, which is dual-engined for any system to be able to mitigate the ever-rising DDoS attacks in the cloud space by making extensive use of ML for an accurate classification and identification of the incoming traffi,c along with the blockchain layer which takes care of the trust management and validation. It provides a contribution to the field of DDoS mitigation, which is comprehensive as well as multi-dimensional and is based on cloud through the following key elements:

1. Intelligent Detection Module: We present a classification engine based on an ML algorithm, i.e. Random Forest, which has been trained over the CICDDoS2019 dataset. The model extracts flow-level features and analyses them, such as packet count, entropy, and inter-arrival time, in order to identify malicious traffic patterns with high accuracy.
2. Decentralised Trust Management: To enhance the integrity of the decisions made and reduce single-point vulnerabilities, we design a private network backed by blockchain, which is based on Ethereum. Smart contracts on this chain dynamically assign and update trust scores for each source IP based on the behaviour that is observed.
3. Smart Contract Driven Mitigation: A programmable mechanism is incorporated for recovery into the blockchain layer, which enables the system to adaptively reevaluate trust thresholds. Suspicious entities can be temporarily blocked or gradually reinstated based on their historical behaviour, promoting both agility and fairness in mitigation.
4. Comprehensive Performance Evaluation: The system is compared against traditional approaches using key metrics such as packet drop rate, latency, system downtime, CPU and network overhead, and self-recovery capabilities.
5. Transparency, Reproducibility & Real-World Applicability: By integrating auditability of the blockchain layer with prediction capability of the ML layer, the framework supports end-to-end transparency and reproducibility.

With the help of this approach, the proposed system not only advances its performance technically in cloud security but also fosters a model for governance which is rooted in trust, automation, and decentralisation. In contrast to the previous efforts which were conducted, the presented study incorporates a hybrid approach in the architecture of the model, which ensures the transparency of decision, adaptability of the system in real-time and upgradation in the recovery mechanism, which was lacking in previous works.

2. Literature Review

Recent studies show that combining machine learning with blockchain can improve DDoS detection accuracy, as demonstrated by Chaira et al.[1], through their approach relies on static decision logic without adaptive trust recovery. Jawahar et al. [6] propose

Ethereum- based smar contracts for blacklisting attackers; however, model achieves limited detection rates, but post-misclassification recovery for benign nodes is not addressed. Poonia et al. [2] introduce a decentralized signalling mechanism using blockchain to coordinate mitigation, yet real-time ML- based detection is absent. Karmode et al.[3]. Present decentralized scrubbing nodes for cost efficient traffic filtering, but the framework does not include intelligent classification or adaptive trust logic. Wnag and Li[8] enhance DDoS detection precision using deep learning models; nevertheless, decentralized trust enforcement is not considered. Jayadev et al. [10] analyze blockchain performance under volumetric DDoS attacks, but proactive detection and trust – aware mitigation remain unexplored. Kavita et al. [13] propose an Ant Lion–based Advanced Encryption Standard with Blockchain (ALAESB) model to enhance the security and privacy of healthcare data. The model achieves high throughput (850 Kbps), low latency (as low as 30 ms), and reduced encryption/decryption times. The review highlights gaps integrating blockchain reputation with real-time ML detection, limited evaluation on standard DDoS datasets, and unresolved scalability and overhead issues (Table 1).

Table 1: Comparison of literature

Study & Year	Technique	Dataset / Evaluation	Blockchain Component	ML/AI Used	Limitations
Chaira et al. (2025)	MEC + RF, Transformer, LightGBM	CICDDoS2019	Logging & decentralised MEC	Yes	No trust scoring or recovery logic
Poonia et al. (2023)	Ethereum smart contracts for blacklisting	Gas cost, latency, scalability (modelled)	Signalling and mitigation coordination	No	No ML classifier, lacks adaptive mitigation
Karmode et al. (2024)	Blockchain-based decentralised scrubbing nodes	DDoS-type evaluation (volumetric, SYN, UDP)	Traffic filter coordination	No	No ML or parameter tuning
Wani et al. (2021)	Layered blockchain trust model for IoT	Strategy-level architecture	Multi-tier distributed ledger	No	Theoretical model lacks real-time mitigation
Jawahar et al. (2024)	ANN + Blockchain blacklisting in SDN	Simulated attack environment	Ethereum-based blacklists	Yes (ANN)	Low accuracy (~72.5%), no recovery logic
Kumari et al. (2025)	RF classifier with Blockchain logging	CICIDS2017, CICIDS2018	Immutable decision logging	Yes (RF)	No reintegration or fairness logic
Wang & Li et al. (2021)	Transformer-based DDoS detection in SDN	SDN-specific traffic benchmarks	Not applicable	Yes (Transformer)	No trust model or blockchain usage
Ouhssini et al. (2024)	CNN-LSTM-Transformer + genetic feature selection	High-fidelity cloud simulations	Not applicable	Yes (Deep hybrid)	No post-detection enforcement
Jayadev et al. (2024)	Ethereum vs. Hyperledger under DDoS stress	Consensus, latency, sync metrics	Benchmarking blockchain behaviour	No	No mitigation engine or classifier
Dandugudum & Tallapally (2024)	IGhostTaV2Net ensemble model with SDN flow control	Modern DDoS testbeds	Not applicable	Yes (GhostNet, TaNet)	No trust or audit logging
Ramadass et al. (2024)	DL + Blockchain SDN with NTRU authentication	DDoS flood scenarios	Smart contracts, digital signatures	Yes (DL)	Static response, no trust-based adaptation
Kavita et al. (2023)	Ant Lion AES + Blockchain encryption	Encryption latency & throughput	Session key logs & Merkle validation	No	Not for DDoS; lacks ML and dynamic mitigation

3. Problem Statement

With the exponential growth of digital services, cloud computing has become a backbone for the storage of data, web services, and enterprise operations. Its elasticity, pay-as-you-go model, and high availability have accelerated its adoption across industries. However, this growing dependence has also made cloud infrastructures prime targets for DDoS attacks. DDoS attacks exploit vulnerabilities of the network by flooding it with services constituted by malicious traffic, which, as a result, overwhelms computing and network resources. The consequences include increased latency, downtime, degradation of service, and ultimately, denial of access to legitimate users, which leads to severe financial and reputational losses. This paper addresses this gap.

4. Proposed Methodology

This work puts forth a framework which is an ensemble of some of the most advanced technologies present in recent times, i.e. ML for the real-time detection of DDoS and the blockchain technology, which provides several features such as trust enforcement, decision automation, and secure logging.

4.1. System Overview

There are five independent layers which form the complete architecture of the solution presented for the problem, which is used to process and respond to incoming traffic. This modular design is characterised by responsiveness, transparency, and self-corrective behaviour, which makes the system extremely reliable and sturdy against the ever-evolving landscape of threats. Figure 1 gives a flowchart of the model workflow.

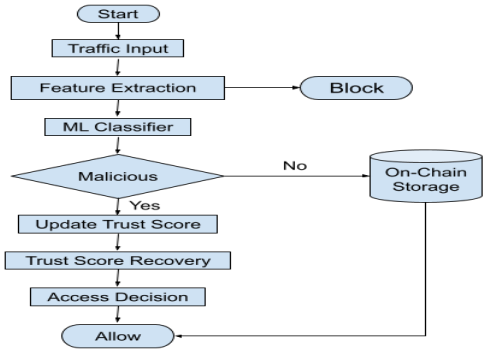

Figure 1: Flow chart of the proposed model.

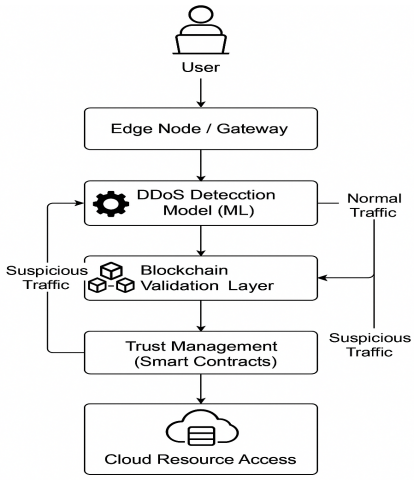

Figure 2: Architecture of blockchain model.

To realise the experimental setup, the system was synthesized in a simulated environment inside cloud which is deployed using Docker containers within Virtual Private Cloud. The idea is to be able to emulate the real environment within cloud which allows precise control over the patterns of traffic, for intensity of attacks and params of the system for reproducibility. A private Ethereum blockchain was instantiated using Ganache to measure transaction latency, trust update overhead, and smart contract execution costs.

4.2. Traffic Preprocessing and Feature Extraction

Raw traffic is captured with tools such as packet sniffer (e.g., Wireshark, Scapy) and is grouped into flows using the CICDDoS2019 dataset. A flow can typically be defined as a unidirectional sequence of packets with common attributes (e.g., IP address, port, protocol). For every flow that takes places, several features are calculated such as, duration of flow (T_f) which is time between the first and last packet, count of packets (P_c) which is total number of packets per flow, Packet Size Metrics which are minimum, maximum, mean length, Inter-Arrival Time (I_t) which is time gap between consecutive packets, and Entropy Measures which are packet size variation, source/destination variability. The data obtained after processing is transformed into a feature matrix (X), $X \in \mathbb{R}^{nXd}$ where n is the number of flow records and d is the number of features. This matrix then becomes the input data for the ML detection module.

4.3. Anomaly Detection with Machine Learning

In this work, a random forest classifier has been implemented, which pertains to the perseverance and

resistance of the algorithm towards overfitting and interpretability. Multiple decision trees T_1, T_2, T_3 ….., T_k have been implemented to form the larger random forest model, where each of these decision trees is trained on arbitrary data samples and feature subsets. The ensemble decision is computed as given in equation 1:

$$\boldsymbol{Y = f(X) = Majority\ Vote\ (T_1(X), T_2(X), .., T_k(X))} \quad \textbf{(1)}$$

Where:
X is the input feature vector.
$Y \in \{0,1\}$ is the predicted class (0 = benign, 1 = malicious)
T_i is the i^{th} decision tree in the forest.
A stratified k-fold cross-validation method has been executed on CICDDoS2019 data as a part of the training and validation of the model. Feature importance is ranked via information gain, and top-k features are retained using Recursive Feature Elimination (RFE).

4.4. Blockchain Integration Layer

Each prediction made by the ML module is logged in a private Ethereum blockchain. This ensures tamper-proof recording of decisions made by the system. Each transaction is represented by

$$TX = timestamp, source\ IP, ML\ label, trust\ source, decision$$

The permanence and integrity of these logs are ensured by a tree structure called a Merkle tree. Consensus is achieved via Proof-of-Authority (PoA) to balance security and computational efficiency. The life cycle of the traffic interactions and their trust scores are managed by Smart contracts, which are written in Solidity. Each address (or IP token) maps to a smart contract state variable to track its trust score. Figure 2 outlines the working of the blockchain layer of the proposed architecture.

4.4.1 Cryptographic Foundations of the Blockchain Layer

The proposed blockchain-based mitigation framework is built upon well-established cryptography-based mechanisms that ensure data integrity, authenticated participation, and secure coordination among decentralised components.

At the ledger level, cryptographic hash functions derived from fundamental cryptographic principles, are used to generate fixed-length, immutable representations of detection results, trust score updates, and mitigation actions before they are recorded in a block. These cryptographic hashes are organised within a Merkle tree structure, enabling efficient integrity verification. Any alteration to stored mitigation records results in a hash mismatch, thereby making tampering immediately detectable and ensuring transparent auditability through cryptographic verification. Within the Proof-of-Authority (POA) consensus mechanism, validator nodes rely on public-private key cryptography to digitally sign validation transactions. Through digital signature cryptography, authorised validator identifies are verified, impersonation attacks are prevented, and non-repudiation of mitigation decisions recorded on the distributed ledger is guaranteed.

Secure inter-layer communication between Machine Learning detection module, blockchain nodes, and smart contract interfaces is protected using cryptography-enabled Transport Layer Security protocols. This ensures confidentiality, integrity, and authenticity of data exchanged during real-time traffic classification and mitigation enforcement. At the application layer, HMAC-based cryptography is implemented to authenticate API communications. Each request is verified using a shared secret- derived cryptographic token, ensuring message integrity and origin authentication while mitigating spoofed, replayed, and bot-generated flooding attempts at the service entry point.

Collectively these cryptography-driven mechanisms form the security foundation of the proposed architecture, reinforcing the integrity, authenticity and verifiability of blockchain-based mitigation decisions while maintaining computational efficiency.

4.5. Trust Score Evaluation

To maintain a long-term reputation for each source, a dynamic trust score $S_i \in [0,1]$ is assigned. Trust scores are updated on each interaction using the following rule defined in equation 2:

$$\boldsymbol{S_i(t+1) = S_i(t) + \alpha\,(L - P)} \quad \textbf{(2)}$$

Where:
$S_i(t)$: Trust score of IP i at the time t.
L: Label assigned by ML (1 for benign, -1 for malicious)
P: Whether the prediction was correct (1) or incorrect (0)
α: Trust update learning rate (empirically set to 0.1)

The scoring mechanism is based on a reward and penalty method where the consistent, benign

behaviours are rewarded while the malicious patterns are penalised.

4.6. Smart Contract Logic for Access Control

Access based on trust is autonomously enforced by smart contracts where the outcomes for decision are made by ML module, which ensures transparency and an evolution of trust that is tamper-proof. The decentralization of logic of mitigation eliminates the occurrence of single point of failure and that eventually prevents the system from manipulation. There are thresholds for trust defined already which helps the system in determining and executing the blockage or reinstatement of access. This decentralised enforcement improves resilience, auditability, and fairness in large-scale cloud environments. Smart contracts enforce access control via conditional execution. Requests are accepted or blocked based on the trust score relative to a threshold as outlined in Algorithm 1:

Algorithm 1: Smart Contract
Input: msg.sender - Address of the requesting IP
trustScore [msg.sender]: Trust Score of the source IP

Output: Access granted or denied

Step 1: Check the trust score associated with **msg.sender**
function **Access Control(msg.sender)**
Step 2: If trust score ≥ threshold (0.5), allow access
if **trustScore[msg.sender]** ≥ 0.5, then **allowRequest()**
Step 3: Else, block the request
else **blockRequest()**
Step 4: Log the IP as malicious for auditing, **logMaliciousAttempt(msg.sender)**
Step 5: End

Real-time decision for access is enforced using this algorithm. Any request that has a threshold below the trust score is rejected the audit logs for which are stored over the blockchain for verification and auditing. Each decision tree operates in $O\,(d \cdot log\, n)$, leading to an overall complexity of $O\,(k \cdot d \cdot log\, n)$ for the Random Forest model. Hence, the computational complexity of this smart contract is $O(1)$ because the decision is based on a simple check of the condition.

4.7. Auto-Recovery and Self-healing Capabilities

An auto-recovery protocol is included in the system, which helps in ensuring the fairness and adaptability of the framework. In the event of an IP sustaining its benign behaviour over time, it can regain the trust of the system after having been classified and marked as malicious in previous events with the use of the relation given in equation 3.

$$S_i^{recover}\,(t+1) = min\,(1, S_i(t) + \beta \cdot t) \quad (3)$$

(3)

Where: t is Time duration since the last malicious classification and $\beta$is Recovery coefficient (typically 0.05)

This approach avoids permanent blacklisting and allows for the recalibration of trust from the system. This makes it efficient enough to identify false positives or any temporary anomalies in the network. The methodology outlined tell about how detection, verification and a mechanism for recovery has been tied into the system. It ensures high detection accuracy, low false positives, decentralised trust, and automatic self-healing, all essential features for modern cloud-based DDoS defence.

Algorithm 2: DDoS Detection and Trust-Based Mitigation
Input: T- Incoming traffic dataset,
F- pre-trained Random Forest Classifier
Si- Initial Trust Score mapping for all Ips

Output: Autonomous classification of traffic, updated trust scores on blockchain and access decisions enforced via smart contracts

Step 1: while (**T** receives a packet) do
Step 2: Group packets into flows F using IP, port, and protocol
Step 3: Extract features **X = [T_f, P_c, I_t, SizeMetrics, Entropy]**
Step 4: Predict label **Y = RF_Model(X)**
Step 5: Retrieve the current trust score S_i from the blockchain
Step 6: if Y == Malicious then
Step 7: Update trust **$S_i \leftarrow S_i - \alpha \times (1 - P)$**
Step 8: if **S_i < Threshold** then
Step 9: Block traffic
Step 10: Log **TX** on the blockchain
Step 11: else
Step 12: Allow traffic
Step 13: Log **TX** on blockchain
Step 14: else
Step 15: Update trust **$S_i \leftarrow S_i + \alpha \times (1 - P)$**
Step 16: Allow traffic
Step 17: Log **TX on Blockchain**
Step 18: End If
Step 19: Periodically apply recovery: **$S_i \leftarrow \min(1, S_i + \beta \times t)$**
Step 20: End While

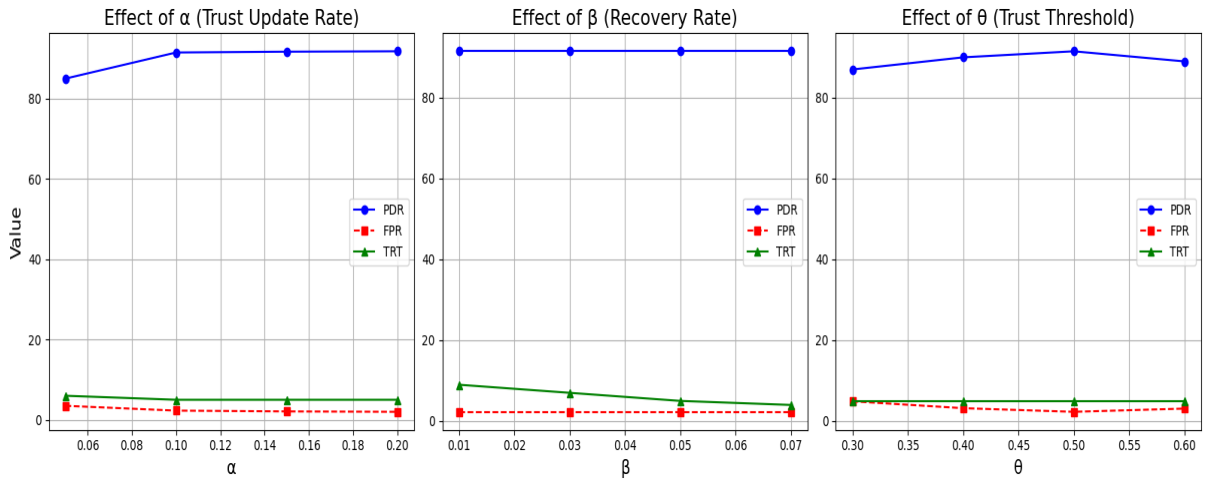

Figure 3: Effect of core trust parameters - α, β, and θ on key performance indicators.

Algorithm 2 details the detection and trust process. Meaningful features are extracted from the incoming traffic, followed by the traffic being classified by using an ML algorithm (RF), after which a trust profile is created and updated dynamically for each of these IP addresses based on the behaviour observed. The auditability and transparency of all the decisions made are ensured by recording them immutably in the blockchain network. Whether an IP will get access or not is autonomously granted or denied by the smart contracts, which efficiently reduces the probability of error created due to human intervention and enhances the operational continuity of the system. Figure 3 gives a graphical representation of how the model performed under changing parameter values. The performance of the system was simulated by varying the core parameters, which can directly influence the response of the system, either by being aggressive or conservative in terms of classifying the behaviour of the traffic.

The reason to select CICDDoS2019 dataset is due to the realistic nature of it, in addition to the diversity of modern types of DDoS attacks and also, the fact that it has been prime choices in most of the recent literatures, which makes it a better candidate for comparative analysis. This data incorporates attacks that are volumetric in nature, are protocol based and act in application layer as in the Syn floods or UDP floods, DNS amplification, and HTTP-based attacks. A flow-level feature has been provided by this dataset which are derived from the raw incoming traffic, which includes signature all the way from temporal, statistical and entropy-based, which makes the dataset entirely well-suited for DDoS detection based on supervised ML.

5. Result Analysis and Evaluation

The acquired results highlight the effectiveness of the proposed framework, with its responses being highly efficient, and the impact of the system, with the help of several evaluation parameters and performance indicators that are seamlessly measurable.

5.1. Experimental setup and evaluation metrics

To simulate realistic DDoS scenarios, experiments were conducted using the CICDDoS2019 dataset within a controlled cloud environment deployed on a Virtual Private Cloud using Docker containers. A private Ethereum blockchain (Ganache) was employed for trust management, with smart contracts implemented in Solidity. The Random Forest classifier (100 estimators) was trained using an 80:20 stratified train-test split to preserve class balance between benign and attack traffic, and the framework was evaluated over a 7-day controlled traffic injection period to assess behavioral trust dynamics.

The model has shown consistent performance, by achieving 96.4% accuracy, 94.7% precision, 95.2% recall, and a 94.9% F1 score in Figure 4, which indicates balanced false positive and false negative rates which is essentially critical for trust preservation. The ROC curve which has been generated from predicted probability scores in Figure 5 attains an AUC of approximately 0.98, that confirms high sensitivity across varying thresholds and supporting adaptive deployment under different security levels.

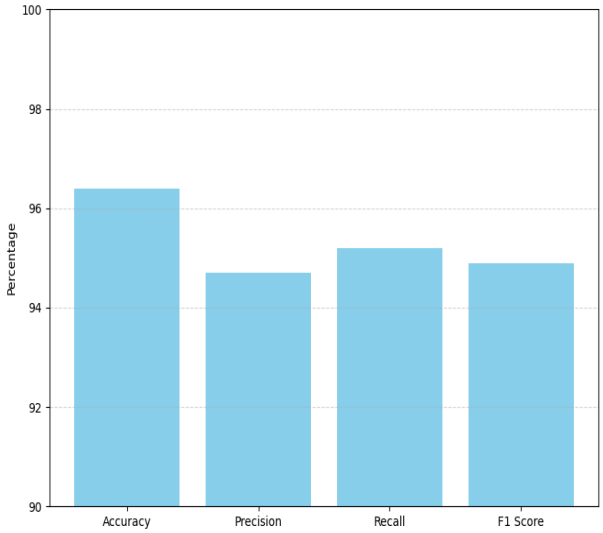

Figure 4: Performance of the proposed

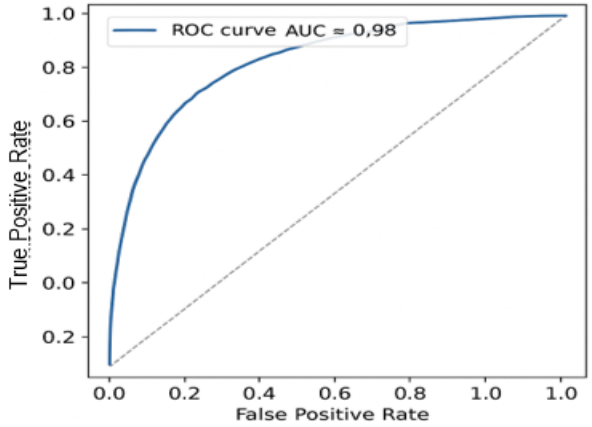

Figure 5: ROC Curve of proposed model

In order to compute various metrics, the ratio of malicious packets was calculated against the total generated malicious packets during the period of attack, which gives PDR. System Latency was computed as the average of the end-to-end delay which was introduced by detection, evaluation of trust and other components which were enforced. CPU and network overhead were computed by comparing resource utilisation before and after activating the proposed framework, averaged over multiple attack cycles to ensure stability and reproducibility.

The results which have been obtained by the model help us highlight significantly about the effectiveness of the proposed framework while also taking care of two things, i.e., identification and neutralization of DDoS traffic and maintaining close to zero interruptions to the performance of the system. As it can be seen in Table 6, the numbers advocate for the greater accuracy achieved by the model and operational continuity than static defences. A more nuanced dimension of adaptability is added to the system which essentially owes to the auto-recovery mechanism, which ensures complete fairness in the event of temporary misclassification. Table 2 gives a comparative analysis with the traditional framework.

Table 2: Comparative Results for Evaluation Metrics

Metric	Proposed Framework	Traditional Firewall	Rate Limiter
Packet Drop Rate (%)	91.5	75.4	68.2
False Positive Rate (%)	2.3	8.7	6.4
False Negative Rate (%)	3.1	10.9	12.8
System Latency (ms)	140	225	190
Downtime (minutes/day)	<2	12	8
CPU Overhead (%)	+15	+28	+22
Network Overhead (%)	+8	+20	+17
Auto-Recovery Time (min)	5	N/A	N/A

The choice of evaluation metrics on which we evaluated our model does not just come from an effort to reinforce how robust the model is or to validate the accuracy of detection, rather also highlights the contributions made beyond the frameworks which used conventional approaches for DDoS detection and mitigation. Metrics like PDR and FPR confirm the core capability of the system in being able to confidently distinguish between malicious traffic and the ones that are benign traffic. However, the model's adaptive and equitable nature is underscored by ARE, which was inadequately captured in previous works. The high score achieved for ARE validates the effectiveness at scale for the proposed work. In the previous schemes, the trust score was based on applied

statistics or a threshold bound, which would penalise benign traffic even when it changes its behaviour. Taking care of that, the dynamic evolution of trust has been enabled such that over time, the traffic is re-evaluated and the score is corrected based on the feedback from real-time classification. This not only ensures higher attack interception but also supports fast, fair, and autonomous recovery of misclassified benign entities. Moreover, the proposed system anchors down all the decisions that are made as a trail in the blockchain, which is auditable.

5.2. Comparison with Related Work

The model put forth was compared against a blockchain-supported SDN framework [11] and a deep learning-based framework [12]. These prior works have shown significant strengths in the mitigation task; however, each one has its own limitations. The challenges faced by each of these work mostly revolve around the lack of an integrated system that offers explainability, dynamic trust handling, and decentralised auditability simultaneously.

The ML modules helps the system in adapting itself to the evolution of traffic and its pattern unlike static firewalls and rate limiting factors. The integration of decentralization allows for enforcement of trust and mitigation aware of recovery channelized by blockchain, which helps enable a kind of decision making that is both fair and accountable and targets benign entities well. This setup leads for the system to have a lower latency, reduced downtime, and improved fairness compared to conventional defences.

Table 3 provides a structured comparison between the proposed model and two recent state-of-the-art frameworks. While Dandugudum et al. [11] demonstrate high accuracy using an ensemble deep learning classifier optimised for SDN traffic, their model lacks mechanisms for post-classification fairness. Similarly, although Ramadass et al. [12] integrate Ethereum-based authentication within an SDN controller, their framework focuses primarily on enforcing flow rules through blockchain validation rather than adapting to changing trust dynamics. On the other hand, the model which has been proposed in this paper, has been set up for evolution of trust, auto recovery and auditability that is transparent, which offers a mechanism that is not only modular but also aware of fairness. The computation of trust has been taken care of by α, β, and θ values, which gives the system a freedom to keep updating the reputation of the user based on consistency of the behaviour shown. This tell that the system not only detects false positives but keep correcting for it as well in real time with the help of different metrics such as **Trust Recovery Time (TRT)** and **Auto-Recovery Efficiency (ARE)**, neither of which is present in the compared works

Table 3: Comparison with Related Work

Feature / Metric	Proposed Model	Dandugudum & Tallapally (2025) [11]	Ramadass et al. (2024) [12]
Detection Technique	Random Forest–based ML with real-time behavioural monitoring	Ensemble DL (IGhostTaV2Net with SBB Optimisation)	Deep learning classifier integrated with SDN
Enforcement Layer	Blockchain-based trust audit and decentralised signalling	SDN-based flow rule redirection	Ethereum-based smart contracts with NTRU authentication
Trust Scoring	Dynamic, parameterised trust score using tunable α, β, and θ	Not implemented	Not implemented
Recovery Logic	Auto-recovery mechanism for misclassified benign entities; tracked using TRT & ARE	Not included	Not included
Auditability	Full blockchain logging of actions, classifications, and trust transitions	No audit trail	Partial logging via blockchain contracts
Dataset Used	CICIDS2017/2018 (configurable); adaptable to other labeled traffic datasets	CICIDS2017; proprietary test environment	Real-time simulated SDN testbed with high-volume traffic
Accuracy	~99.34% (RF); trade-off tuned with interpretability and recovery metrics	~99.82%	~98.5%
False Positive Handling	Includes fairness metrics (Trust Recovery Time, Auto-Recovery Efficiency)	Low FPR, but no adaptive fairness logic	Acceptable FPR, but no mitigation for benign flagging
Blockchain Overhead	Lightweight signalling and trust log propagation; Ethereum or Hyperledger-compatible	None	Moderate latency observed in Ethereum smart contract transactions
Key Contribution	Modular integration of ML, adaptive trust, and blockchain for fairness-aware mitigation	Accuracy-optimised ensemble learning in SDN environments	Secure flow authentication in a blockchain-SDN hybrid system

The steady rise in Figure 6, reaching ~92% by the final epoch, reflects effective learning convergence. Figure 7 indicates improved discrimination between benign

and malicious flows. Figure 8 shows that the proposed model attains a 91.5% PDR, marginally outperforming the baseline strategies. Figure 9 presents the lightweight trust scoring making the model deployable in edge and IoT contexts where resource constraints are a challenge. Figure 10 captures the trust score evolution of a representative network node, demonstrating the adaptive nature of the scoring mechanism. Figure 11 illustrates that the majority of benign entities recover within a narrow range (~5 seconds), indicating fast auto-repair.

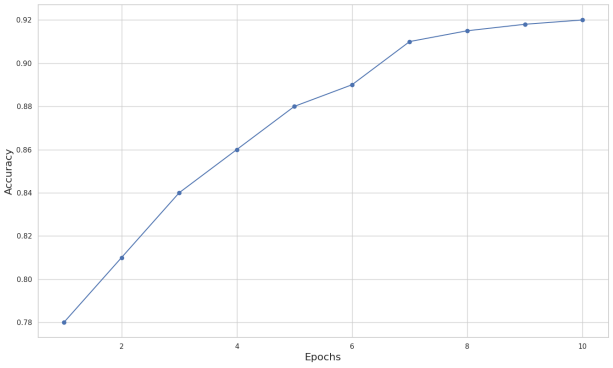

Figure 6: Model accuracy over epoch

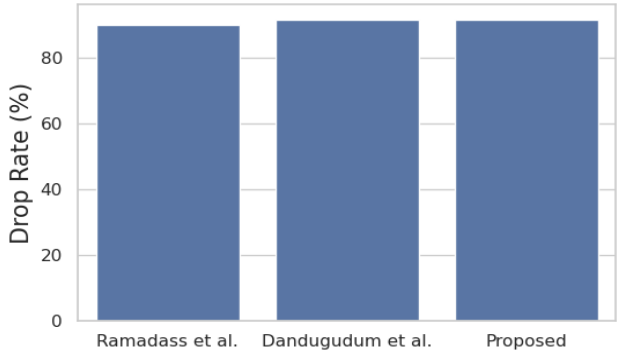

Figure 7: Comparison of FPR across models.

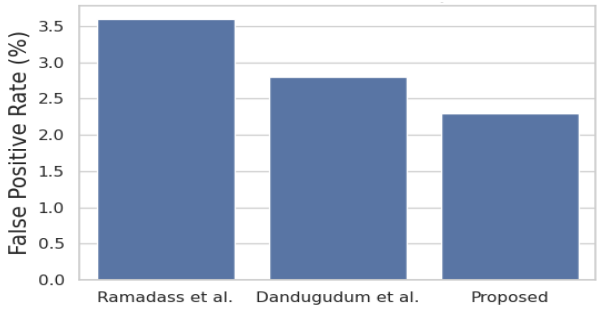

Figure 8: PDR comparison across models.

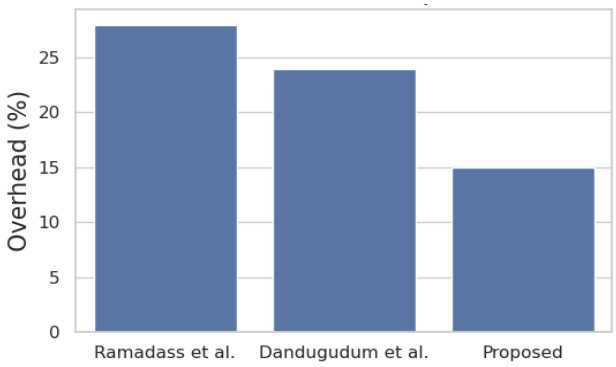

Figure 9: CPU Overhead comparison

6. Conclusion and Future Work

The system under consideration works by integrating supervised ML to detect DDoS and trust enforcement which is driven by blockchain to give it an ability to be autonomous, act with resilience and be transparent. The results which have been obtained show a superior performance shown by the proposed work over other work that has been carried out. The model shows 96.4% accuracy for detection which is supplemented by reduction in false positives, a very less disruption of service and very minimal computation power required. The smart contract-based trust mechanism enables real-time revocation, decentralized verification, and tamper-proof auditability without reliance on centralized gateways. The integration of cryptographic primitives strengthens the blockchain layer by ensuring authenticated validation, tamper-evident logging, and secure communication, enhancing decentralised trust without impacting system performance. Future enhancements include extending the framework to multi-cloud deployments, incorporating online learning for adaptive model refinement, and integrating real-time QoS and latency feedback into trust decision logic. While the strengths of the system is considered, it does have potential limitations owning to the transaction latency which is introduced by the trust updates based on blockchain. Also, this implementation is based on ML model that was trained offline, incorporation of online or continual learning continual learning mechanisms would further enhance adaptability against zero-day or evolving attack patterns.

References

[1] Chaira, Mahmoud, Abdelkader Belhenniche, and Roman Chertovskih. "Enhancing DDoS Attacks Mitigation Using Machine Learning and Blockchain-Based Mobile Edge Computing in IoT" *Computation* 13, no. 7: 158. 2025

[2] Laxmi Poonia, "DDoS Mitigation by Blockchain With Approach of Cost Model", *IJRITCC*, vol. 11, no. 9, pp. 3873–3880, Nov. 2023.

[3] S. Karmode, "Innovative DDoS Mitigation: A Blockchain-Based Decentralized Traffic Scrubbing Approach", JoSCNDS, vol. 1, no. 2, pp. 33–40, Aug. 2024.

[4] Wani S, "Distributed Denial of Service (DDoS) Mitigation Using Blockchain—A Comprehensive Insight", . *Symmetry*, 13(2):227, 2021

[5] Kithmini Godewatte Arachchige, "Blockchain-Enabled Mitigation Strategies for Distributed Denial of Service Attacks in IoT Sensor Networks: An Experimental Approach", Computers Material and Continua, Vol. 81, Issue 3, pp. 3679-3705, 2024

[6] A, J., P, K., C, V.K. *et al.* "DDoS mitigation using blockchain and machine learning techniques". *Multimed Tools Appl* 83, 60265–60278, 2024

[7] Kalpana R, Sridevi S, A Post-Quantum Cryptography and Machine Learning-Driven Framework for Securing Blockchain-Based Supply Chain. Int J Drug Deliv Technol. 2026;16(4s): 30-43; DOI: 10.25258/ijddt.16.30-43.

[8] Wang, Haomin, and Wei Li. "DDosTC: A Transformer-Based Network Attack Detection Hybrid Mechanism in SDN" *Sensors* 21, no. 15: 5047, 2021

[9] M ouhssini, "DeepDefend: A comprehensive framework for DDoS attack detection and prevention in cloud computing", Journal of King Saud University-Computer and Information Sciences, Vol. 36, Issue 2, 2024

[10] Vijay Jayadev et al. "Assessing the Performance of Ethereum and Hyperledger Fabric Under DDoS Attacks for Cyber-Physical Systems". In Proceedings of the 19th International Conference on Availability, Reliability and Security (ARES '24). Association for Computing Machinery, New York, NY, USA, Article 48, 1–6. 2024.

[11] Mahesh, D., Tallapally, "Advanced SDN-based network security: an ensemble optimized deep learning-based framework for mitigating DDoS attacks with intrusion detection". *Cluster Comput* 28, 331, 2025.

[12] Parthasarathy Ramadass, "BSDN-HMTD: A blockchain supported SDN framework for detecting DDoS attacks using deep learning method". Egyptian Informatics Journal, Volume 27, 2024.

[13] Kavita, K. , "A novel optimization-based blockchain technology using health care data for enhancing security and privacy in the medical system", Journal of Discrete Mathematical Sciences and Cryptography, 27:8, 2483–2494, 2024

[14] Arvind, S., "Advancing cyber threat detection through deep learning in management information systems", Journal of Discrete Mathematical Sciences and Cryptography, 27:8, 2409–2417, 2024

[15] Mallikarjuna, "Comparing the roles of cryptography and blockchain technology in relation to Internet of Things", Journal of Discrete Mathematical Sciences and Cryptography, 27:7, 2015–2025, 2024

[16] Li, S., Qin, T., & Min, G. Blockchain-based digital forensics investigation framework in the internet of things and social systems. IEEE Transactions on Computational Social Systems, 6(6), 1433-1441 (2019).

[17] Dai, Hong-Ning, Zibin Zheng, and Yan Zhang. "Blockchain for Internet of Things: A survey." IEEE internet of things journal 6.5 : 8076-8094 (2019).

[18] Viriyasitavat, Wattana, Li Da Xu, Zhuming Bi, and Danupol Hoonsopon. "Blockchain technology for applications in internet of things mapping from system design perspective." IEEE Internet of Things Journal 6, no. 5 : 8155-8168 (2019).

[19] Yousefi, Samuel, and Babak Mohamadpour Tosarkani. "Exploring the role of blockchain technology in improving sustainable supply chain performance: a system-analysis-based approach." IEEE Transactions on Engineering Management (2023).

[20] Kushwaha, S. S., Joshi, S., & Gupta, A. K. (2023). An efficient approach to secure smart contract of Ethereum blockchain using hybrid security analysis approach. Journal of Discrete Mathematical Sciences and Cryptography, 26(5), 1499-1517.

[21] Mallikarjuna, Basetty. "Feedback-based fuzzy resource management in IoT-based-cloud." International Journal of Fog Computing (IJFC) 3, no. 1 : 1-21 (2020).

[22] M. T. Islam, S. Karunasekera, and R. Buyya, "Performance and cost-efficient spark job scheduling based on deep reinforcement learning in cloud computing environments," IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 7, pp. 1695–1710, Jul. (2021).

[23] Chattopadhyay, S., Sahoo, A. K., & Jasola, S. (2023). Improvement in DDoS attack detection in software defined network using ML algorithm. *J. Discrete Math. Sci. Cryptogr*, *26*(7), 2025-2044.