Iranian Journal of Fuzzy Systems

Volume 22, Number 6, (2025), pp. 1-20

Original paper

University of Sistan and Baluchestan

Enhance data security with efficient anomaly detection in a real network environment

M. Priyadharshini[1] and B. Vanathi[2]

[1] *Department of Computer Science and Engineering, SRM Valliammai Engineering College, TamilNadu, India*
[2] *Department of Computer Science and Engineering, SRM Valliammai Engineering College, TamilNadu, India*

priyadharshinim.mail@gmail.com,priyadharshinim.cse@srmvalliammai.ac.in,vanathib.cse@srmvalliammai.ac.in

Abstract

Security in share markets is essential to affirm the integrity, stability, and trust of financial transactions, protecting investors from fraud and cyber threats. Current methods face challenges in high-volume attacks due to their scalability and static rule-based mechanisms. To resolve these issues, this paper develops a Hybrid Isolated Fuzzy Logic to detect anomalies in transactions. The Fuzzy Logic System with the Zebra Optimization Algorithm is utilized to enhance the attack detection accuracy. The isolation forest algorithm computes the threshold abnormal score to distinguish the normal from the malicious transactions. The Reinforcement Learning-based Proximal Policy Optimization algorithm dynamically updates network policies. The Network Function Virtualization for Distributed Denial-of-Service Scrubbing is combined to improve scalability and deliver cost-effective mitigation. The experimental analysis utilizing stock anomaly detection datasets is conducted. The experimental outcomes affirm that the proposed model attains an accuracy of 98.90%, a false positive rate of 2.9%, and improved mitigation efficiency than existing methods.

Keywords: Software defined networking, fuzzy logic system, zebra optimization algorithm, reinforcement learning, proximal policy optimization.

1 Introduction

In modern financial markets, real-time transactions, high-frequency trading (HFT), and automated trading platforms have changed stock trading, enabling investors to execute trades in milliseconds [16]. The security of the share markets is a critical concern as the digitization of stock exchanges and online trading platforms increases [23]. In this context, Distributed Denial-of-Service (DDoS) attacks are severe threats to stock market networks [18, 19]. The cyberattacks affect trading operations and lead to manipulated stock prices, liquidity crises, and major economic consequences [10]. Moreover, stock markets work in a highly interconnected and globalized environment; meanwhile, short-lived service disruption is an issue for several financial institutions, affecting traders and investors globally [6]. Software-defined networking (SDN) delivers significant benefits for network management and security and is the solution for managing and securing financial network infrastructures [14].

The SDN demonstrates that legitimate trading transactions remain uninterrupted while filtering out malicious traffic by using Machine Learning (ML)-based traffic profiling and adaptive security policies [25]. This robust proactive security model reduces latency issues, prevents market disruptions, and secures financial assets [13]. Several models are developed to detect anomalies in share market transactions; at the same time, these methods are ineffective due to time consumption, higher false rates, higher delay rates, etc. So, this paper proposes a Hybrid Isolated Fuzzy Logic (HIFL) model. The novelty of the HIFL approach is provided as follows:

Corresponding Author: M. Priyadharshini
Received: August 2024; Revised: August 2025; Accepted: September 2025.
https://doi.org/10.22111/ijfs.2025.49590.8748

• **Hybrid Anomaly Detection:** The research work presents a HIFL approach that improves anomaly classification using an optimized fuzzy controller.

• **Optimization-driven fuzzy controller:** The proposed model utilizes an Improved Zebra Optimization Algorithm (IZOA) to enhance the precision of the anomaly classification. The integration of the Chaotic Map (CM) and Simulated Annealing (SA) with the IZOA algorithm prevents premature convergence, demonstrating robust anomaly detection in high-speed trading environments.

• **Adaptive threshold computation:** The proposed model utilizes an Isolation Forest-based thresholding mechanism to dynamically assess abnormal stock market behaviour.

• **Attack Mitigation Model:** Reinforcement Learning (RL)-based Proximal Policy Optimization (PPO) algorithm (RL-PPO) learns optimal mitigation strategies by analyzing past attack patterns and continuously refines its response mechanisms, adapting to emerging cyber threats in financial networks.

• **SDN and NFV-Enabled Scalable Defence System:** In this work, the SDN and Network Functions Virtualization (NFV) are employed for dynamic security policy enforcement. This dynamically reconfigures SDN-based routing policies to isolate malicious traffic in real-time.

1.1 Contribution

• **Hybrid SDN-based anomaly detection and mitigation model:** This research introduces a HIFL model for real-time monitoring and protection of financial transactions from DDoS attacks.

• **Effective Anomaly Detection Model:** The proposed HIFL approach effectively classifies normal and anomalous stock trading patterns. The IZOA optimizes the fuzzy control system to fine-tune fuzzy membership functions.

• **Adaptive Threshold calculation:** The isolation Forest algorithm (iForest) is employed to calculate the threshold abnormal score, allowing accurate differentiation between normal and DDoS attack traffic.

• **Robust Attack Mitigation Approach:** A RL-PPO is enabled to optimize the attack mitigation step. It learns optimal response strategies from past attack data to mitigate threats more efficiently.

• **NFV-based DDoS scrubbing:** The integrated NFV with SDN supports dynamic, scalable, and cost-effective security policies.

• **Comprehensive Experimental Validation:** The developed model is validated using real-world datasets, namely the Securities and Exchange Board of India (SEBI) and Dow Jones Industrial Average-30 (DJIA 30) stock time series datasets.

The structure of the research work is mentioned as follows: The related works are provided in Section 2, Section 3 explains the system model, Section 4 interprets the working steps of the proposed approach, Section 5 conducts the comprehensive experimental analyses, and Section 6 imparts the conclusion with future works.

2 Literature review

2.1 Anomaly detection in SDN

Chen et al. [4] introduced a network monitoring system for network monitoring system. Here, Long Short-Term Memory (LSTM) and Graph Convolutional Networks (GCN) approaches were integrated for the end-to-end model. It effectively extracted spatial-temporal relationships within data. The outcomes demonstrated its ability in anomaly detection. For effective anomaly detection, Zavrak and Iskefiyeli [26] presented an SDN anomaly detector (SAnDet). In this work, flow features were collected from the OpenFlow switches. Using these features, the SAnDet model has found unknown attacks by a hybrid DL model. The estimations on the Canadian Institute for Cybersecurity Distributed Denial of Service 2019 dataset (CICDDoS2019) utilizing performance parameters exposed the capability of the SAnDet model.

Jafarian et al. [11] established Security Anomalies Detection and Mitigation in the SDN (SADM-SDNC) model. This work has imparted effective features by applying the NetFlow protocol, as well as creating an information ratio

and collecting information. In the detection of attacks, the C-support vector classification algorithm was employed. This work covered empirical evaluations to test the performance of the model. The developed method has reached an improved detection rate. Wang and Wang [22] designed a mitigation SDN system using Convolutional Neural Network and Extreme Learning Machine (CNN-ELM). For the mitigation of the attacks, the Internet Protocol (IP) traceback was exploited, which alerted flow rule commands from the controller by finding abnormal traffic. The efficiency of the system was gauged, which ensured its performance with greater efficiency rates. Abdallah et al. [1] presented a Hybrid anomaly detection System that is beneficial to SDNs by detecting attacks. Two deep learning (DL) methods, namely LSTM and CNN, were combined to generate the hybrid system. The evaluation of the model was performed on the comprehensive SDN-dataset employed in this work. This ensured the effectiveness of the developed system with better accuracy rates.

2.2 SDN-NVF for DDoS attack detection and mitigation

Domínguez-Dorado et al. [7] introduced an SDNFV network for successful attack detection in SDN. This work has applied virtualized and the entropy mechanism with the controller has analyze the flows for the identification of the attacks. The findings demonstrated that the developed approach procures greater rates. Chen et al. [5] established an adversarial Deep Belief Networks-LSTM (DBN-LSTM) model for better attack detection on SDN. In this work, the data were procured and transformed to generate the numerical features. An adversarial dataset and several examples were generated by the Generative Adversarial Networks (GAN). The experimental findings of this work demonstrated that the developed method reached sufficient performance.

Najar and Naik [15] presented a Balanced Random Sampling (BRS)-CNN (BRS-CNN) to identify DDoS Attacks from SDN. To monitor the blocked IP addresses, another monitoring system was developed in this work. This work was alerted via email address with the utilization of a developed in-built email notification system. The performance analysis was carried out with several performance parameters. Jiang et al. [12] implemented a Blockchain-based SDN-targeted DDoS defense model (BSD-Guard) to safeguard the SDN controllers. A secure middle plane with a blockchain model among the data and control planes was generated. Based on information from the collected packet, the new flow's suspect rate was calculated. The blockchain has obtained the suspect list from the secure middle plane. The performance assessments held in this paper affirmed the proficiency of the developed BSD-Guard approach. For the DDoS attacks, Wang et al. [20] established a two-stage detection and mitigation model. This work has deployed two methods, namely CNN and wavelet decomposition, to develop a Multi-Dimensional Deep Convolutional Classifier (MDDCC) for the extraction of multi-dimensional characteristics. For validation of the model, this work granted a comprehensive performance evaluation, indicating that the developed model has higher rates. Table 1 provides the differences between the proposed model and the existing methods. Prior works have leveraged a variety of ML and DL approaches, implemented over controllers. Unlike existing methods, the proposed model integrates fuzzy logic on the Ryu SDN controller, offering improved detection accuracy and efficient mitigation.

Table 1: Comparison of the proposed model with the existing methods

Author Name	Model	Controller
Chen et al. [4]	Network monitoring system	SDN
Zavrak and Iskefiyeli [26]	SAnDet	Floodlight
Jafarian et al. [11]	SADM-SDNC	Floodlight
Wang and Wang [22]	CNN-ELM	RYU
Abdallah et al. [1]	LSTM and CNN	SDN
Domínguez-Dorado et al. [7]	SDNFV network	RYU

Chen et al. [5]	DBN-LSTM	SDN
Najar and Naik [15]	BRS-CNN	POX
Jiang et al. [12]	BSD-Guard	Open Network Operating System (ONOS)
Wang et al. [20]	MDDCC	POX
Proposed Framework	HIFL	Ryu SDN

2.3 Limitations and research gaps

The previous studies developed for DDoS attack detection have some advantages, while these methods encounter several issues. Rule-based traditional methods make false alarms even when legitimate trading activity fluctuates. Traditional ML methods struggle with imbalanced datasets and wrongly classify real transactions as threats. For effective anomaly detection, the proposed model enables real-time fuzzy logic-based decision-making rather than static thresholds. With the utilization of the iForest, the threshold abnormal score is calculated, demonstrating accurate differentiation between normal market activity and cyber threats. This minimizes the false positives and improves the reliability of anomaly detection. Existing methods depend on predefined rules, leading to a higher False Positive Rate (FPR) and False Negative Rate (FNR). The IZOA optimizes membership functions, fuzzy control rules, and scaling factors in the fuzzy controller, ensuring greater classification accuracy. RL-PPO refines the attack mitigation in real time, reducing misclassification errors. Existing methods demand higher computational power and a longer time for the training process. In the proposed model, SDN-based implementation enables centralized control and dynamic resource allocation, reducing computational overhead in stock trading networks. NFV-based DDoS scrubbing demonstrates that attack traffic is filtered at the network edge and reduces the load on trading servers. The abbreviations (Table S1) and the parameter definition (Table S2) are provided in the supplementary file (Part A).

3 System model

The SDN encompasses centralized control, dynamic traffic management, and security enforcement. In controllers, there is a need to scrutinize and oversee forward to the traffic in the controllers to secure the SDN network. Initially, the controller observes the traffic, and then feature extraction and classification are performed. The several pre-defined classes are classified from the incoming data by a trained classification approach. The controllers are allocated for all traffic types; they receive the packets following traffic engineering. Based on queuing and priority policies from the switches and controllers, the data is forwarded to the destination address. Figure 1 illustrates the SDN architecture comprising the data plane, control plane, and application plane.

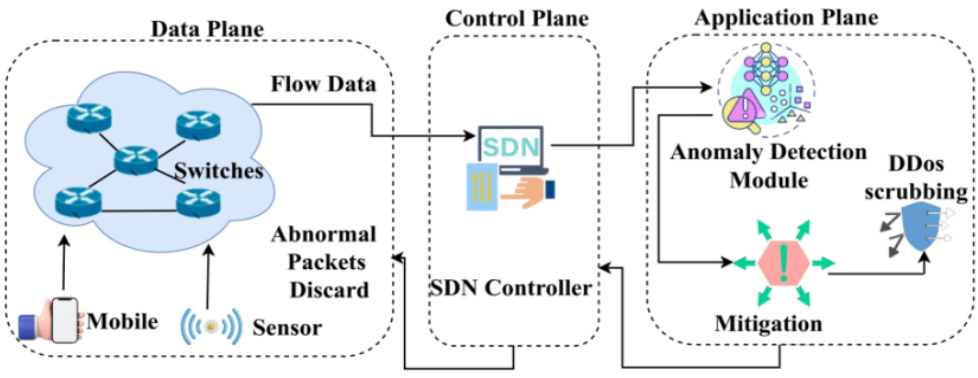


Figure 1: SDN architecture comprising data, control, and application planes

Infrastructure Layer: Generally, the Infrastructure layer encompasses open-source virtual switches linked to the hosts. The controller receives the traffic from the switches by connecting them to the control layer, and the switches acquire the commands. The switches receive, process, and forward market transactions. Utilizing OpenFlow, switches communicate with the SDN controller, enabling real-time policy updates based on detected anomalies.

Control Layer: The control layer acts as the central intelligence of the SDN controller. It maintains the traffic flows, security policies, and real-time decision-making. The controller monitors the incoming share market transactions and captures the important traffic features, including flow behavior, packet size, and latency. Based on these features, unusual trading patterns are identified.

Application Layer: This layer executes both the traffic classifier and the network monitor system. The traffic classifier categorizes the incoming traffic into copious categories by applying ML, and the network oversees the time delay, byte count, and incoming traffic.

Detection and Attack Mitigation: In the controller, this detection module acts as an application to identify network anomalies automatically and to gather network features and traffic information. This module's primary aim is to find DDoS flooding attacks. Once an attack is detected, the DDoS Mitigation module ensures that legitimate financial transactions remain unaffected while blocking malicious traffic. In the SDN network, the mitigation agent acts as a host, and the mitigation server is enabled as an application in the controller.

4 Proposed methodology

Figure 2 illustrates the architecture of a proposed HIFL framework. The system is structured into three layers, including the Infrastructure Layer, the Control Layer, and the Detection and Mitigation Module.

The Infrastructure Layer consists of OpenFlow switches that manage data traffic through defined paths. The control Layer holds an SDN controller that gathers network flow data and handles packet forwarding or dropping based on detected threats. The detection and mitigation module utilizes a Fuzzy Logic System with an IZOA to analyze network traffic for anomalies. If an attack is detected, the system activates RL-PPO to mitigate the threat. The DDoS Scrubbing process, using NFV, helps filter malicious traffic. If no attack is detected, legitimate IPs and ports are stored for normal transaction processing. This architecture enhances the security and reliability of financial transactions in the share market. The pre-processing techniques, detailed steps of IZOA, and computational complexity are provided in the supplementary file (Part B).

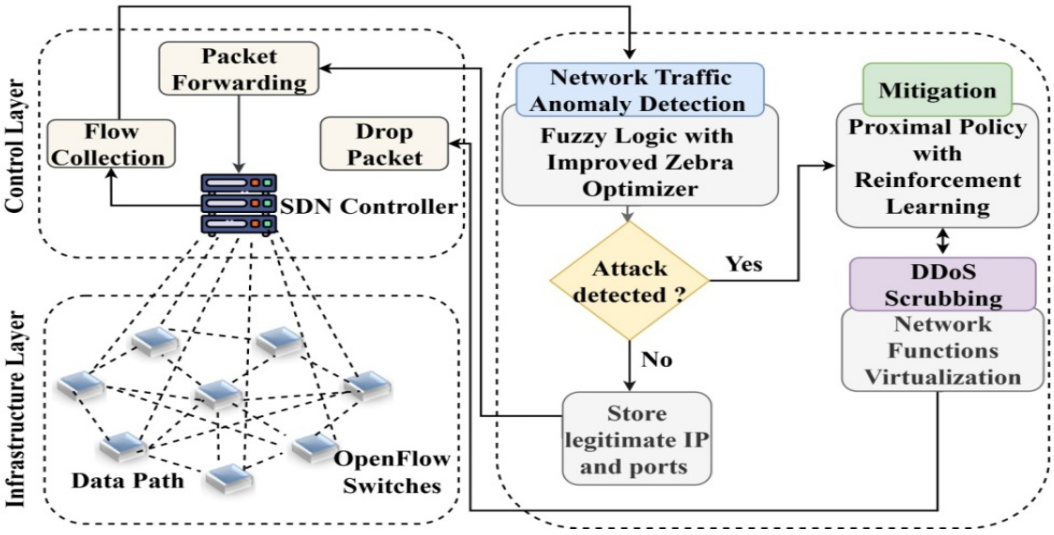


Figure 2: Proposed HIFL architecture

4.1 Data collection

The OpenFlow protocol is generally applied in the SDN environment to monitor the trading platform traffic. The OpenFlow protocol assists in analyzing order flow, identifying unusual trading spikes, and detecting transaction delays. However, the traditional OpenFlow protocol-based methods struggle with high-rate DDoS attacks as attack traffic affects the bandwidth between stock exchange servers and brokerage firms. DDoS attacks on the share market block stock orders, delay HFT, and disrupt financial transactions. This leads to liquidity issues, trading halts, and price manipulation. The NFV model is combined with the SDN to mitigate such risks. It allows the deployment of virtualized

security functions dynamically within stock exchange networks. This reduces the impact of DDoS attacks, demonstrating the smooth operation of automated systems and the real-time stock exchange.

4.2 Real-time traffic anomaly detection

To secure the stock exchange network from cyber attacks, this research work implements a fuzzy logic-based traffic anomaly detection system.

Fuzzy Logic

The fuzzy logic system is crucial as it is effective in handling uncertainty and ambiguity and mimics human reasoning to make decisions [2]. The fuzzification step converts input variables into fuzzy values. Then, a set of fuzzy rules is employed to classify the transactions as normal or anomalous. The defuzzification generates an anomaly score to determine if trading activity is part of a DDoS attack. A Gaussian membership function is employed to determine the fuzzy degree of an element:

$$\zeta = e^{\frac{-(y-\widehat{y})^2}{2\Theta^2}}. \tag{1}$$

In the above equation, ζ is the fuzzy membership degree of the input, the standard deviation that measures normal market fluctuations is denoted as Θ, the expected normal value is represented as $\widehat{y}$, and the observed market variable is indicated as y. The anomaly score is computed using the following equation:

$$\zeta_q = 1 - e^{\frac{-(y_q-\hat{y}_q+\varepsilon)}{2\Theta_q^2}}. \tag{2}$$

In the above equation, ζ_q represents the refined fuzzy anomaly score for feature q, real-time trading data feature is specified as y_q, $\hat{y}_q$ is the mean value of the q^{th}, ε is the small nonzero value, and Θ_q denotes the standard deviation of normal market fluctuations. The Θ_q controls the spread of the Gaussian, influencing the fuzziness of the classification. This enhanced formulation improves sensitivity to subtle DDoS activity. The higher anomaly score ζ_q indicates the transaction is more likely to be anomalous. After the fuzzy logic system calculates the anomaly score ζ_q, the iForest is used to determine the threshold abnormal score. Fuzzy logic helps mitigate the risk of faulty optimization in real-world financial applications; meanwhile, it gets trapped in local market fluctuations rather than finding broader trends. This leads to poor dynamic performance in real-time stock market transactions where rapid price changes and anomalies occur. To resolve this, the improved meta-heuristic algorithm is combined to refine fuzzy rules, demonstrating better adaptability to market conditions. The threshold for the detection of anomalous stock transactions is determined utilizing the iForest, which evaluates the anomaly score and distinguishes irregularities from normal market fluctuations.

• Computing the abnormal score threshold via the Isolation Forest algorithm

The iForest is a supervised ML model developed for anomaly detection, and it works based on isolating data points by randomly generated decision trees [9]. The following equation represents the computation of the threshold abnormal score. This threshold helps to determine whether a transaction is normal or anomalous.

$$S = 2\frac{E(h(y))}{c(m)}. \tag{3}$$

In this equation, S denotes the threshold abnormal score, the average isolation path length for a transaction is denoted as $E(h(y))$ and $c(m)$ indicates the expected path length in the balanced binary tree. After the iForest determines the threshold abnormal score, it is compared with ζ_q.

$$Rule\ 1:\ \text{if}\,\zeta_q \geq \text{threshold}\ \phi, \text{THEN "the transaction is anomalous"}, \tag{4}$$

$$Rule\ 2:\ \text{if}\,\zeta_q < \text{threshold}\ \phi, \text{THEN "the transaction is normal"}. \tag{5}$$

Firstly, the fuzzy logic system computes an anomaly score ζ_q for each transaction, and the *iForest* determines the threshold abnormal score φ based on transaction patterns. Ultimately, the transactions are classified as anomalous or normal by comparing ζ_q with φ.

4.2.1 Fuzzy controller optimization using the improved algorithm

To enhance the accuracy and adaptability of the fuzzy logic-based anomaly detection system, this work incorporates an IZOA to optimize the fuzzy controller. In this work, a Gaussian membership function is selected for input variables such as packet size, flow duration, and flow inter-arrival time, due to its smoothness and flexibility in modeling gradual

transitions in financial traffic. Fuzzy rules are formulated by observing historical network traffic behavior, and a set of rules covers different stock market transaction behaviors. A Mamdani-type fuzzy inference system is deployed for inference. Inputs are fuzzified, fuzzy rules are estimated, and outputs are defuzzified using the centroid method to obtain a crisp anomaly score. In this work, parameters are optimized to enhance the adaptability of the fuzzy controller. This includes K_{pq}, K_{pj}, and K_{pc}, where the differential scale factors are noted as K_{pc}, the integral factor is represented as K_{pj}, and the scale factor is indicated as K_{pq}. The fuzzy controller generates three control outputs, represented as H_{qy}, H_{jy}, and H_{cy}. These outputs are computed based on their respective initial values and the contribution of the optimized parameters. The mathematical formulations of these outputs are given by:

$$H_{qy} = H_{q0} + H_{ay}{}^{*} K_{pq}, \tag{6}$$

$$H_{jy} = H_{j0}{}^{*} H_{jy}{}^{*} K_{pj}, \tag{7}$$

$$H_{cy} = H_{c0} + H_{cy}{}^{*} K - pc. \tag{8}$$

In the above equations, H_{q0}, H_{j0}, and H_{c0} are the initial values of the control outputs.

The steps of the improved zebra optimization are mentioned:

The ZOA is an efficacious optimization algorithm that can determine the value of design variables [17]. This algorithm struggles to resolve high-dimensional problems due to local optima and the diversity of initial solutions. To rectify these problems, the ZOA solution's diversity is enhanced by combining the CM into the initialization phase. The SA is then embedded within ZOA's exploration phase, boosting local search efficiency.

Initialization: In the algorithm, chaos theory explores the search space more thoroughly compared to the random search [8]. The population diversity is enhanced by the initialization of the IZOA using a CM. The circle map is considered the initial point Y, which is computed using the modulo 2π operation, ensuring values are confined within a normalized range of $[0, 2\pi]$. This normalization is crucial to maintain angular consistency since using a different base will represent values incorrectly in the context of circular functions. To introduce nonlinearity and avoid premature convergence, the chaotic sequence is generated via the CM. Equation (9) is employed to compute the CM:

$$CM = Y_{n+1} = Y_{ji} + q - \left(\frac{p}{2\pi}\right) \sin\left(2\pi y_{ji}\right), \ \mathrm{mod}(1). \tag{9}$$

In the above equation, Y_{ji} denotes the current chaotic value for the individual j in the dimension i, $\sin(2\pi y_{ji})$ denotes a sine-based perturbation that ensures nonlinearity in the mapping, Y_{n+1} denotes the n^{th} chaotic number of the chaotic sequence, mod(1) operation ensures that the output remains within the unit interval $[0, 1]$, and p and q are the CM control variables, which are determined as $p = 0.5$ and $q = 0.2$.

Fitness evaluation: As shown in the following equation, the fitness function is assessed by measuring the decrease in classification error rate.

$$F(y_j) = \frac{N_C}{T_C} * 100. \tag{10}$$

In the above equation, $F(y_j)$ denotes the fitness value, the number of misclassified instances is indicated as N_C and the total number of instances is specified as T_C. The mathematical Model of the ZOA algorithm is presented as follows:

In the search space, the zebras' initial position is randomly allocated, and then the matrix indicates their population mathematically. The following equation represents the matrix Z for the initial population:

$$Z = \begin{bmatrix} Z_1 \\ \vdots \\ Z_j \\ \vdots \\ Z_M \end{bmatrix}_{M \times u} = \begin{bmatrix} z_{1,1} & \cdots & z_{1,i} & \cdots & z_{1,u} \\ \vdots & \ddots & \vdots & & \vdots \\ z_{j,1} & \cdots & z_{j,i} & \cdots & z_{j,u} \\ \vdots & & \vdots & \ddots & \vdots \\ z_{M,1} & \cdots & z_{M,i} & \cdots & z_{M,u} \end{bmatrix}_{M \times u}, \tag{11}$$

In the above equation, the number of population members is represented as M, and the value of i^{th} decision variable for the j^{th} zebra is indicated as $z_{j,i}$, and the number of decision variables is denoted as u. Based on the values of the zebra, the objective function is assessed and it is indicated in the following equation.

$$V = \begin{bmatrix} V_1 \\ \vdots \\ V_j \\ \vdots \\ V_M \end{bmatrix} = \begin{bmatrix} V(Z_1) \\ \vdots \\ V(Z_j) \\ \vdots \\ V(Z_M) \end{bmatrix}_{M\times 1}, \tag{12}$$

In the above equation, the vector of objective function values is represented as the V and the objective function value of the j^{th} zebra is noted as V_j.

• *Foraging Behavior*

The simulations of zebra behavior determine the population members' updating in this foraging phase. Equation (13) represents the mathematical model to update the position of the zebras in this phase.

$$Z_{j,i}^{new,K1} = z_{j,i} + rd \cdot (PB_i - 1.z_{j,i}). \tag{13}$$

$$Z_j = \begin{cases} Z_{j,i}^{new,K1}, & V_j^{new,K1} < V_j; \\ Z_j, & else, \end{cases} \tag{14}$$

In the above equations, the random number is indicated as rd that has interval $[0, 1]$, Z_j indicates the current solution vector, pioneer zebra is denoted as PB_i, $z_{j,i}$ is the current value of the decision variable, $Z_{j,i}^{new,K1}$ is the i^{th} dimension value of the new status, $V_j^{new,K1}$ denotes the objective function value of the new status.

• *Defense strategies to protect them from predators*

Two attack strategies are there, the first one is zebra selects the escape strategy when it is attacked by the lion. The second one is that zebras select the proactive strategy when facing predator attacks. The two strategies of zebra are represented as R_1 and R_2. The following equation implies the update condition to update the new position of the zebra with a higher value for the objective function.

$$Z_{j,i}^{\text{new},K2} = \begin{cases} R_1 : z_{j,i} + S.(2rd - 1).\left(1 - \frac{g}{G}\right).z_{j,i}, & O_r \leq 0.5; \\ R_2 : z_{j,i} + rd \cdot (SZ - I._{z_{j,i}}) & else \end{cases} \tag{15}$$

$$Z_j = \begin{cases} Z_{j,i}^{\text{new},K2}, & V_j^{\text{new},K2} < V_j \\ Z_j & \text{else} \end{cases} \tag{16}$$

In the above equation (15), $Z_{j,i}^{\text{new},K2}$ indicates the updated position, S is a constant number equal to 0.01, rd is the random number, O_r indicates the probability of the strategies, iteration number is mentioned as g, and G denotes the maximum number of iterations, status of the attacked zebra is represented as SZ. In equation (16), $V_j^{\text{new},K2}$ denotes the fitness of the new candidate solution.

• *Boosted the exploitation phase*

To enhance the best solution and evade the local optima stagnation problem, this work utilizes SA at the end of every iteration. The chances of selecting the worst solution could be determined by the Boltzmann probability.

$$P = eg - E(G_{solution} - B_{solution}). \tag{17}$$

In the above equation, P represents the acceptance probability of a worse solution, the control parameter is indicated as E, eg is the expected gain, representing the benefit of the current solution in terms of classification accuracy, the generated solution is specified as $G_{solution}$, and the best solution is noted as $B_{solution}$, The fitness function for $G_{solution}$ and $B_{solution}$ computes the ratio of the probabilities. The pseudo-code of the IZOA is represented in Algorithm 1.

Algorithm 1: Pseudo-code of IZOA

Start

1. Input: Optimization problem information
2. Set the population of zebras (M) and the number of iterations indicated as (G)
3. Initialize zebra positions using the objective function evaluation and CM
4. $for\ g = 1 : G$ do
5. Update Pioneer Zebra PB
6. $for\ j = 1 : M$ do
7. Foraging behavior (Phase 1)
8. The j^{th} zebra's state is updated using equation (13)
9. Using equation (14), the position of the j^{th} zebra is updated
10. Defense strategies against predators (Phase 2)
11. if $O_r < 0.5,\ O_r = rd$
12. Strategy 1: Exploitation phase (against the lion)
13. Utilize equation (15) with (R_1) to compute the new status of the j^{th} zebra
14. else
15. Strategy 2: Exploration phase (against other predators)
16. Compute the new status of the j^{th} zebra using Equation (15) with (R_2)
17. end if
18. Apply Equation (16) to update the position of j^{th} zebra
19. end *for*
20. Save the best solution identified up to the current iteration
21. end *for*
22. Use SA
23. $for\ g{=}1toG\ \ do$
24. Calculate Boltzmann probability using Equation (17)
25. Accept new solution if $_P$ is greater than a randomly generated threshold
26. Otherwise, retain the previous best solution
27. end for
28. Output: The best

End IZOA

4.3 Attack mitigation phase

The integration of PPO with real-time traffic attributes allows the system to effectively minimize the impact of attacks while maintaining optimal network performance, offering a robust, adaptive, and efficient approach to attack mitigation in SDN-based financial environments.

4.3.1 RL-based PPO algorithm

PPO is a RL technique that enables the system to progressively adapt and improve its mitigation strategies to reach optimal performance [3]. The agent chooses actions on the basis of real-time market conditions and network traffic patterns, continuously improving its decision-making ability. To formulate the anomaly mitigation problem, a Markov Decision Process (MDP) is established. The MDP consists of various components, including states and actions. The agent $g \in C$ selects actions from a set of actions C, and each environment state is noted as T. The discount factor γ is used to control the significance of future rewards. In the environment state, the agent takes an action e. The state transition is defined as $q(T_{s+1} \mid T_s, g_s)$, and the reward model is noted as $q(p_{s+1} \mid T_s, g_s)$. Here, p_{s+1} is the reward obtained from the environment as a result of the agent's action.

Proximal Policy Optimization Algorithm

The step interface delivers real-time data, termination flags, immediate rewards, and continuous observations. The reset function resets the environment after training and establishes a stable baseline state for further learning iterations. The updating process encompasses the critic network μ and the actor-network θ. The critic network μ estimates the expected returns of policy, and the actor network determines the action probabilities to update the current policy. To improve the chances of selecting an effective action in the state T, the parameters θ are optimized. The actor network receives the current environment state and outputs a new policy. The PPO algorithm acquires the optimal policy through iterative learning. The following equation represents the objective function of the policy, in which the strategy function is denoted as π_θ, the time step is specified as i and $O(\theta)$ is specified as the objective function. It is combined with the estimator of the advantage function $\hat{\mathrm{E}}_i$.

$$O(\theta) = A_i \left[\frac{\pi_\theta(e_i|T_i)}{\pi_{\theta old}(e_i|T_i)} \hat{E}_i \right]. \tag{18}$$

In the above equation, A_i is the advantage estimate for the i^{th} experience, T_i is the state observed by the agent at time step i, $\hat{E}_i$ denotes the estimated reward obtained from T_i when action was taken, e_i is the action taken by the agent at time step i, and π_θ and $\pi_{\theta_{old}}$ are the current policy and previous policy. To maintain stable updates, the objective function is reformulated with a clipping mechanism:

$$O(\theta) = A_i \left[\min \left(q_i(\theta)\hat{E}_i,\ clip\big(q_i(\theta), 1-\epsilon, 1+\epsilon\big)\hat{E}_i \right) \right]. \tag{19}$$

In the above equation, $q_i\ (\theta)$ is the sampling ratio, $clip(\cdot)$ is the clipping function, and ϵ denotes the clipping parameter. Equation (19) represents the objective function of the online network with restriction, where ϵ is computed with a 0.2 value. $q_i(\theta)$ ought to be near to 1 for the principle of importance sampling. The PPO algorithm efficiently speeds up the learning process by utilizing past experience through avoiding unnecessary processes. After each policy update, the strategy is adjusted using importance sampling.

$$q_i(\theta) = \frac{\pi_\theta(e_i|T_i)}{\pi_{\theta old}(e_i|T_i)}. \tag{20}$$

To update the policy parameters, a gradient ascent step is performed:

$$\theta_{i+1} = \theta_i + \alpha \nabla_\theta O(\theta_i). \tag{21}$$

Equation (21) implies the optimal policies, which are improved with the use of metrics based on the gradient-based algorithm. The learning rate is specified as α, θ_i is the current policy parameter at timestep i, θ_{i+1} is the updated policy parameter, and $\nabla_\theta O(\theta_i)$ is the gradient of the objective function. The clipping parameter is typically set to 0.2 to ensure stable and conservative policy updates. The mitigation module effectively adapts to evolving attack patterns, optimizing flow table modifications in SDN-based share market environments.

4.3.2 NFV for DDoS Scrubbing

The consolidation of the NFV for DDoS scrubbing strengthens the attack mitigation module by improving scalability, adaptability, and cost efficiency. NFV differs from traditional hardware-based security models since it allows dynamic scaling of virtual security functions. It rapidly evolves attack patterns by programmable security policies. Moreover, NFV seamlessly integrates with the PPO-based anomaly mitigation module, enabling intelligent and automated attack response strategies that optimize both security and network performance. Ultimately, NFV enhances the overall resilience of trading platforms, minimizing downtime and ensuring uninterrupted financial transactions in the stock market. Algorithm 2 presents the pseudo-code representation of the mitigation process.

Algorithm 2: Mitigation Process

1. Initialize replay buffer, critic network μ, actor-network θ, training step T_p, learning rate λ_c, λ_a, discount factor γ, and clip factor ε.
2. *for* n episode *do*
3. Acquire observations from the environment
4. Implement action e_i using current policy
5. Estimate reward p_{s+1}, log probability of e_i, policy entropy $T(t(n))$
6. Store transition $\{T[n], e_i, p_{s+1}, T[n+1]\}$ into experience buffer
7. Use the experience buffer to update the PPO
8. *if* DDoS attacks *then*
9. Find the destination IP address receiving the highest number of flows
10. Detect source IP addresses sending to the same destination as the attacker
11. *if* IPs and ports are on the safe list *then*
12. Forward packets
13. else
14. Drop packets

5 Results and discussion

This segment exhibits the overall experimental analysis of the developed method based on anomaly detection in share market transactions. The comparison methods, definitions of evaluation parameters, and Parameter evaluation are provided in the supplementary file (Part C).

5.1 Experimental setup

The proposed method was implemented using Python 3.9 and TensorFlow on a system with an Intel Core i7 processor, 32 GB RAM, and an NVIDIA RTX 3080 GPU. Model training and testing are conducted on Ubuntu 20.04. The average training time per epoch is approximately 12 seconds. In this virtualized space, Mininet, supporting OpenFlow 1.3 with an SDN controller, is deployed. To analyze the network traffic, Wireshark is employed, and virtual network topologies are formed the MiniEdit. Resource consumption remained within practical limits, supporting real-time deployment. These details demonstrate the method's feasibility for real-world applications.

5.2 Hyperparameter selection and tuning strategy

Table 2 furnishes the ranges for both IZOA and PPO hyperparameters. To ensure optimal convergence and performance, the IZOA and PPO are configured through systematic tuning. For IZOA, a population size of 50 is selected after evaluating values ranging from 30 to 100, ensuring a balance between exploration and computational efficiency while avoiding excessive resource utilization. The mutation factor is set to 4 after empirical testing, as it introduces sufficient diversity into the search space without destabilizing convergence. The iteration count is fixed at 200 based on convergence trend analysis across 5 experimental runs, which provided adequate optimization time without unnecessary computational overhead.

Table 2: Hyperparameters with optimal values

Algorithm	Hyperparameter	Values	Algorithm	Hyperparameter	Values
IZOA	Population Size	50	PPO	Entropy Factor	0.5
	Iteration Count	200		Learning rate	0.001
				Discount Factor	0.95
	Mutation Factor	4		Clipping Threshold	0.2

For the PPO algorithm, the learning rate of 0.001 is selected after testing values from 0.0001 to 0.01, where 0.001 consistently provided stable learning without overshooting. The discount factor is set to 0.95 to balance short-term and long-term rewards, particularly relevant for continuous traffic patterns. The clipping threshold of 0.2 is adopted from common PPO best practices and confirmed empirically to prevent large, destabilizing policy shifts. The entropy factor is tuned to 0.5 to ensure sufficient exploration, avoiding early convergence to suboptimal policies. All hyperparameters are finalized based on validation accuracy, reward stability, and convergence time.

5.3 Dataset description

• **Securities and Exchange Board of India**: The SEBI encompasses instances of manipulations in India (https://www.sebi.gov.in/). It covers cases of manipulations in diverse indices, which are released on the Bombay Stock Exchange (BSE). This incorporates adjudication orders, release orders, show-cause notices, and court orders. The division indices corresponding to different types of manipulation are provided in Table 3.

Table 3: Indices based on manipulation

Manipulation category	Indices used	Overall indices
Volume	3	12
Price	4	13
Common for volume and price		9

• **DJIA 30 Stock Time Series:** The DJIA 30 stock time series dataset encompasses historical stock data for 30 prominent companies listed in the DJIA index (https://www.kaggle.com/datasets/szrlee/stock-time-series-20050101-to-20171231). Key features of the dataset are represented in Table 4.

Table 4: Key features of the dataset

Features	Description	Features	Description
Date	Trading date	High	Highest price during the trading session
Open	Stock's opening price on the date given	Close	Stock's closing price on the given date
Low	Lowest price during the trading session	Adj Close	Closing price adjusted for dividends and splits.

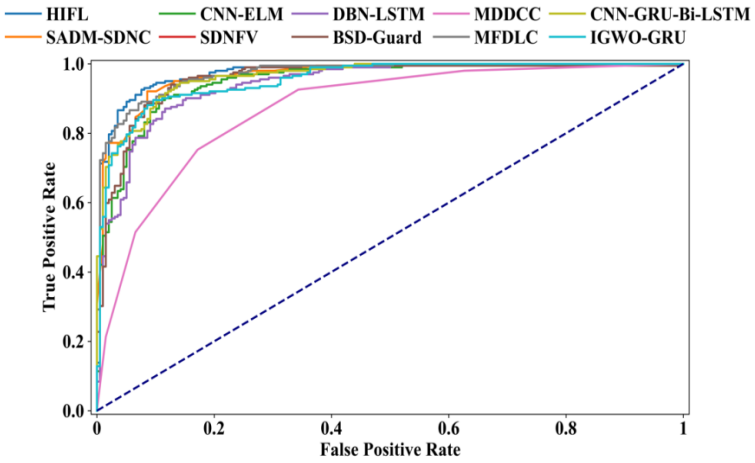
5.4 Evaluation of the detection phase

The comparative assessment of methods using F1-score, recall, precision, accuracy, FPR, and FNR is represented in Table 5. The proposed model reaches higher performance, highlighting the model's effectiveness in accurately identifying both legitimate and fraudulent transactions. Moreover, the proposed model attained lower FPR and FNR by enabling two-stage optimization. The IZOA tunes key hyperparameters globally, while the PPO algorithm is utilized to obtain the optimal policy by refining decision boundaries. This improves the detection precision and minimizes false alarms.

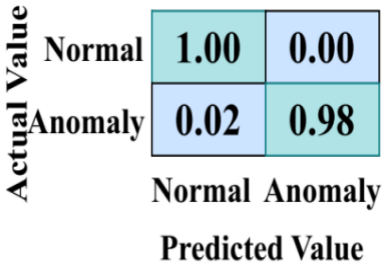
Table 5: Comparative Assessment

Methods	Accuracy	Precision	Recall	Specificity	F1-score	FPR	FNR
Proposed	98.90%	97.40%	96.40%	97.10%	96.90%	2.9%	3.6%
SADM-SDNC [11]	93.50%	91.20%	92.70%	90.80%	91.94%	9.2%	7.3%
CNN-ELM [22]	93.70%	92.80%	93.90%	94.10%	93.35%	5.9%	6.1%
SDNFV [7]	93.20%	89.70%	92.10%	90.70%	90.88%	9.3%	7.9%
DBN-LSTM [5]	94.80%	90.30%	93.30%	92.90%	91.77%	7.1%	6.7%
BSD-Guard [12]	92.60%	91.80%	91.90%	90.80%	91.85%	9.2%	8.1%
MDDCC [20]	93.60%	91.80%	92.70%	92.10%	92.75%	7.9%	7.3%
MFDLC [21]	93.60%	93.10%	92.60%	90.20%	92.85%	9.8%	7.4%
CNN-GRU-Bi-LSTM [27]	94.30%	90.70%	93.80%	92.10%	92.22%	7.9%	6.2%
IGWO-GRU [24]	94.10%	89.90%	93.40%	91.90%	91.62%	8.1%	6.6%

The ROC analysis between existing detection methods and the developed HIFL is represented in Figure 3 (a). Compared to existing methods, the developed HIFL method yields a greater 0.989 Area Under the Curve (AUC) value. Figure 3 (b) depicts the confusion matrix of the developed HIFL approach. It specifies that the proposed attack detection method effectively classifies the number of normal and anomaly instances with better performance.



(b)



(b)

Figure 3: ROC and confusion matrix of the proposed anomaly detection (a)AUC analysis (b) confusion matrix

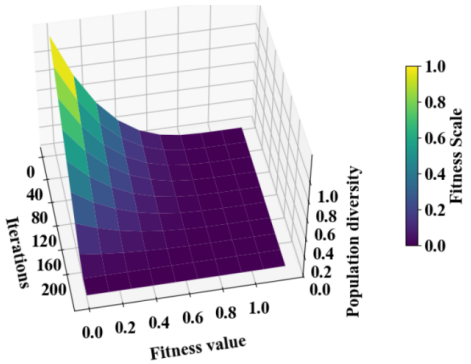


Figure 4: (a)

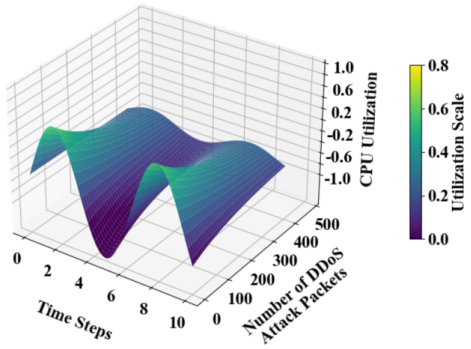


Figure 5: (b)

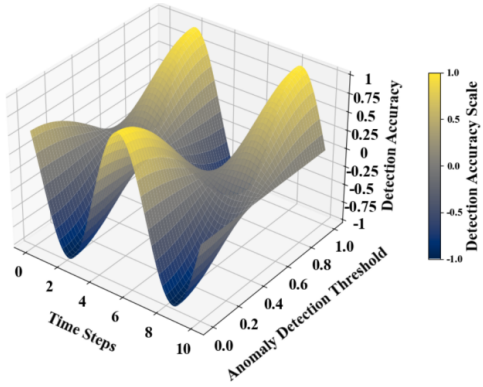


Figure 6: (c)

Figure 7: Performance evaluation of the proposed model (a) Convergence analysis, (b) Impact of DDoS attacks, (c) anomaly detection threshold analysis

Figure 7(a) represents the convergence analysis of the IZOA algorithm evolving across iterations during DDoS detection optimization. It visualizes convergence behavior and highlights the effectiveness of enhancements such as adaptive mutation and fuzzy-based evaluation. Figure 7(b) illustrates how the network performance metrics, such as CPU utilization, are affected by the number of DDoS attack packets over time. It highlights the system's response to increasing attack intensity and the effectiveness of mitigation strategies. Figure 7(c) illustrates the impact of varying detection thresholds on anomaly detection accuracy. The plotted results confirm the model's robustness and precision under varying operational conditions.

5.5 Evaluation of the mitigation phase

With the utilization of the key parameters, beneficial insights are exhibited for the mitigation system's performance. This allows an understanding of the performance of the detection model and helps in informed decisions to improve the network's lifetime. The mitigation estimation is conducted based on the three factors and analyzed as follows:

- **Mitigating packet drop**

Figure 5 represents the CPU utilization of the Ryu controller. After using the proposed mitigation model, CPU utilization initially increases but then stabilizes, ensuring the model's capability to detect and mitigate the attack. This outcome specifies the performance of the SDN-based anomaly detection and mitigation model in ensuring secure transactions by optimizing resource utilization and reducing the impact of cyber threats in stock market trading.

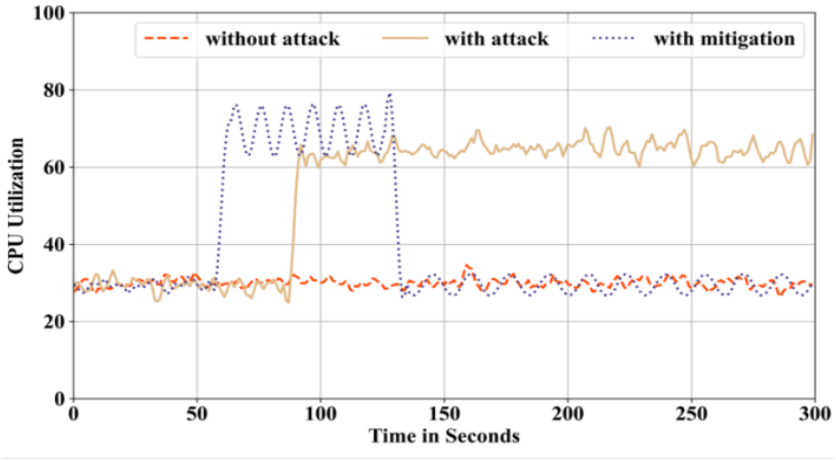


Figure 8: CPU Utilization of Ryu Controller

- **Response time analysis**

Detection and Mitigation Efficiency of HIFL is provided in Table 6. The experiments on real-world datasets demonstrate that the developed HIFL model reaches an attack detection time of 0.4 seconds. The mitigation strategy dynamically adjusts mitigation policies based on attack patterns. The existing methods exhibit high latency because of rule-based filtering; meanwhile, the RL-PPO reduces the mitigation time than the existing methods. The developed HIFL model demonstrates that the CPU utilization of SDN controllers remains within 60% under attack conditions, avoiding system slowdowns.

Table 6: Detection and Mitigation Efficiency of HIFL

Methods	Detection Time	Response Time	CPU Utilization
Proposed	0.4s	1.1s	60%
SADM-SDNC	2.5s	2.6s	75%
CNN-ELM	2.9s	3.5s	79%
SDNFV	3.6s	4.1s	80%
DBN-LSTM	4.6s	5.9s	85%
BSD-Guard	5.6s	6.8s	75%
MDDCC	5.7s	6.9s	80%
MFDLC	6.3s	7.2s	70%
CNN-GRU-Bi-LSTM	6.8s	7.6s	75%
IGWO-GRU	7.4s	8.4s	90%

- **Traffic analysis with mitigation module**

The network traffic in terms of the packets per second over time is analyzed in Figure 6. The green line denotes traffic volume when mitigation is disabled, demonstrating higher packet rates due to the overwhelming effect of the DDoS attack. The blue line indicates the traffic when the mitigation is enabled, in which the packet rates are lower. This indicates that the proposed mitigation model reduces malicious traffic.

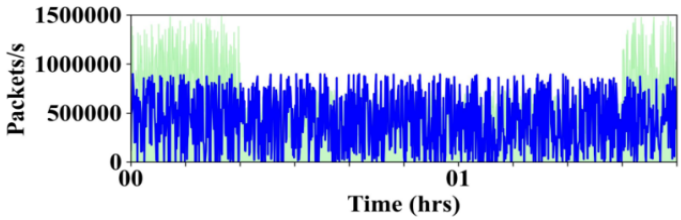


Figure 9: Network Traffic with and without HIFL Mitigation

5.6 Computational cost analysis

Table 7 indicates the computational cost analysis of the proposed HIFL method. The proposed model delivers the finest balance of computational efficiency, cost-effectiveness, and speed. It attains higher accuracy with lower computational overhead and faster response times. This makes it a practical and scalable solution for protecting SDN-based share market transactions from DDoS attacks, ensuring both network security and financial stability with minimal economic impact.

Table 7: Computational cost analysis (H=High, L=Low, M=Medium)

Method	Training Time (s)	Hardware Cost	Computational Cost	Energy Consumption	Maintenance Cost	Overall Cost Efficiency
Proposed	0.9	L	L	L	L	H
SADM-SDNC	3.6	M	M	M	H	M
CNN-ELM	6.4	H	H	H	H	M
SDNFV	7.7	M	M	M	M	M
DBN-LSTM	7.9	H	H	H	H	L
BSD-Guard	8.2	M	M	M	M	M
MDDCC	8.3	M	M	M	M	M
MFDLC	9.1	H	H	H	H	L
CNN-GRU-Bi-LSTM	9.5	H	H	H	H	L
IGWO-GRU	10.2	H	H	H	H	L

5.7 Ablation study

To assess the contribution of each module in the proposed architecture, an ablation study analysis is performed by removing individual components and observing the performance degradation. Table 8 provides the ablation study of the proposed system. The results ensure that each element, IZOA optimization, PPO tuning, and isolation Forest, plays a vital role in maximizing detection accuracy and reducing false positives.

Table 8: Ablation study analysis

Model Variant	Accuracy	FPR
Proposed HIFL	98.90%	2.9%
Without IZOA	97.50%	6.3%
Without PPO	96.20%	5.9%
Without Isolation Forest	95.10%	5.8%

5.8 Statistical analysis

The Wilcoxon signed-rank test is conducted to compare the proposed HIFL-SDN model with baseline methods. The results in Table 9 demonstrate that the improvements achieved by the proposed model are statistically significant ($p<0.05$) across existing frameworks. This validates that the performance gains are consistent and not due to random fluctuations.

Table 9: Wilcoxon signed-rank test analysis

Methods	p-Value
Proposed vs SADM-SDNC	0.003
Proposed vs CNN-ELM	0.007
Proposed vs SDNFV	0.004
Proposed vs DBN-LSTM	0.002
Proposed vs BSD-Guard	0.010
Proposed vs MDDCC	0.006
Proposed vs MFDLC	0.002
Proposed vs CNN-GRU-Bi-LSTM	0.005
Proposed vs IGWO-GRU	0.003

5.9 Discussion

Effective detection and mitigation are required to ensure the security and reliability of network systems, as DDoS attacks have been increasing. The performance evaluation scrutinizes the efficiency of the HIFL model using several key parameters.

Practical Advantages of the HIFL Approach

• **Improved Detection Accuracy:** The developed HIFL model has a greater accuracy of 98.90% than DBN-LSTM and CNN-ELM, ensuring a greater probability of accurately detecting both legitimate and malicious traffic.

• **Minimized False Positive Rates:** The developed HIFL model holds an FPR of 2.9%, ensuring its ability to reduce incorrect classifications of legitimate traffic as anomalous. This is more important for applications in financial markets, where small disruptions cause significant losses.

• **Real-Time Anomaly Detection and Mitigation:** The proposed HIFL effectively adjusts the mitigation strategies through the integration of the RL-PPO. This affirms the minimal latency and efficient traffic management during attacks.

• **Adaptability and Robustness:** The adaptability to dynamic attack patterns is improved by integrating fuzzy logic optimized with IZOA. CM and SA demonstrate robust global and local optimization.

Practical implications of using the proposed system

The practical implications of using the proposed HIFL model in live settings are multifarious. Initially, real-time traffic profiling abilities aid in the detection of malicious activities, leading to decreased disruptions to legitimate operations. Specifically, this is important for financial markets and other domains, where compromised data integrity can affect economic losses. Next, the inclusion of NFV ensures scalable and flexible resource allocation and minimizes the reliance on costly hardware infrastructure. This facilitates the deployment among diverse network environments and minimizes operational costs. The developed HIFL model has minimized the FPR value, which ensures that legitimate traffic is not affected. This is important to demonstrate the uninterrupted service and maintain user trust. The utilization of the PPO algorithm dynamically adjusts the mitigation strategies based on the evolving nature of threats, which improves the system's adaptability.

6 Conclusion

For competently securing the data in share markets, this research work frames an attack detection and mitigation framework called HIFL. This work imparts a successful attack detection by utilizing the fuzzy logic system with an optimized fuzzy controller. Here, IZOA optimizes the fuzzy controller by adding a chaotic map population initialization. The iForest is exploited for the computation of threshold abnormal scores. Following attack detection, the attack mitigation step is performed by the RL-PPO. In addition, this module is combined with the NFV to avoid the possible massive volume of DDoS traffic. To validate the proposed HIFL approach, the SEBI and DJIA 30 Stock Time Series datasets are employed. The performance of the proposed method is scrutinized through extensive experimental analysis. The comparisons reveal the ability of the proposed model as it reaches higher rates. Eventually, the HIFL method is granted as the robust anomaly detection method with a greater recall of 96.40%, accuracy of 98.90%, specificity of 97.10%, F1-score of 96.90%, precision of 97.40%, and lower FNR of 3.6%, and FPR of 2.9%. Future work will be included for further progress in the current work. Some future works to improve the robustness and adaptability of the detection and mitigation strategies are represented as follows:

• **Real-Time Adaptation and Learning:** The online learning techniques will be introduced, where the model can update itself based on attack patterns.

• **Integration of Deep Learning Techniques:** Future works will investigate the consolidation of DL models, including Recurrent Neural Networks (RNNs) or CNNs, for innovative anomaly detection and traffic classification.

• **Hybrid Security Frameworks:** The HIFL will be integrated with other security frameworks, like blockchain for decentralized trust management or Multi-Agent Systems for distributed attack detection and mitigation.

Conflict of interest

The authors declare that they have no conflict of interest.

Human and Animal Rights

This article does not contain any studies with human or animal subjects performed by any of the authors.

Availability of data and material

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] M. Abdallah, N. An Le Khac, H. Jahromi, A. Delia Jurcut, *A hybrid CNN-LSTM based approach for anomaly detection systems in SDNs*, In Proceedings of the 16th International Conference on Availability, Reliability and Security, (2021), 1-7. https://doi.org/10.1145/3465481.3469190

[2] M. G. Abdolrasol, A. Ayob, A. H. Mutlag, T. S. Ustun, *Optimal fuzzy logic controller based PSO for photovoltaic system*, Energy Reports, **9** (2023), 427-434. https://doi.org/10.1016/j.egyr.2022.11.039

[3] M. Alonso, H. Amaris, D. Martin, A. de la Escalera, *Proximal policy optimization for energy management of electric vehicles and PV storage units*, Energies, **16**(15) (2023), 5689. https://doi.org/10.3390/en16155689

[4] H. Chen, P. Chen, B. Wang, X. Yu, X. Chen, D. Ma, Z. Zheng, *Graph neural network based robust anomaly detection at service level in SDN driven microservice system*, Computer Networks, **239** (2024), 110135. https://doi.\org/10.1016/j.comnet.2023.110135

[5] L. Chen, Z. Wang, R. Huo, T. Huang, *An adversarial DBN-LSTM method for detecting and defending against DDoS attacks in SDN environments*, Algorithms, **16**(4) (2023), 197. https://doi.org/10.3390/a16040197

[6] A. Dadhania, P. Dave, J. Bhatia, R. Mehta, M. Kumhar, S. Tanwar, A. Alabdulatif, *Software defined network and graph neural network-based anomaly detection scheme for high speed networks*, Cyber Security and Applications, **3** (2025), 100079. https://doi.org/10.1016/j.csa.2024.100079

[7] M. Domínguez-Dorado, J. Calle-Cancho, J. Galeano-Brajones, F. J. Rodríguez-Pére, D. Cortés-Polo, *Detection and mitigation of security threats using virtualized network functions in software-defined networks*, Applied Sciences, **14**(1) (2023), 374. https://doi.org/10.3390/app14010374

[8] Z. Elgamal, A. Q. M. Sabri, M. Tubishat, D. Tbaishat, S. N. Makhadmeh, O. A. Alomari, *Improved reptile search optimization algorithm using chaotic map and simulated annealing for feature selection in medical field*, IEEE Access, **10** (2022), 51428-51446. https://doi.org/10.1109/ACCESS.2022.3174854

[9] R. Guo, X. Zhu, T. Liu, *Automatic detection of crop lodging from multitemporal satellite data based on the isolation forest algorithm*, Computers and Electronics in Agriculture, **215** (2023), 108415. https://doi.org/10.1016/j.compag.2023.108415

[10] A. Hirsi, M. A. Alhartomi, L. Audah, A. Salh, N. Bin Mad Sahar, S. Ahmed, G. O. Ansa, A. Farah, *Comprehensive analysis of ddos anomaly detection in software-defined networks*, IEEE Access, (2025). https://doi.org/10.1109/ACCESS.2025.3535943

[11] T. Jafarian, M. Masdari, A. Ghaffari, K. Majidzadeh, *SADM-SDNC: Security anomaly detection and mitigation in software-defined networking using C-support vector classification*, Computing, **103**(4) (2021), 641-673. https://doi.org/10.1007/s00607-020-00866-x

[12] S. Jiang, L. Yang, X. Gao, Y. Zhou, T. Feng, Y. Song, K. Liu, G. Cheng, *BSD-Guard: A collaborative blockchain-based approach for detection and mitigation of SDN-Targeted DDoS attacks*, Security and Communication Networks, **2022**(1) (2022), 1608689. https://doi.org/10.1155/2022/1608689

[13] A. V. Kachavimath, D. G. Narayan, *A hybrid deep learning model with consensus-based feature selection for DDoS attacks detection in SDN*, Procedia Computer Science, **252** (2025), 643-652. https://doi.org/10.1016/j.procs.2025.01.024

[14] Y. Medjadba, H. Drid, M. Rahouti, *Intrusion detection in software-defined networking using hybrid Bayesian model averaging for reliable uncertainty quantification*, Computer Networks, (2025), 111436. https://doi.org/10.1016/j.comnet.2025.111436

[15] A. A. Najar, S. M. Naik, *Cyber-secure SDN: A CNN-based approach for efficient detection and mitigation of DDoS attacks*, Computers and Security, **139** (2024), 103716. https://doi.org/10.1016/j.cose.2024.103716

[16] T. Park, *Enhancing anomaly detection in financial markets with an llm-based multi-agent framework*, arXiv preprint arXiv:2403.19735, (2024). https://doi.org/10.48550/arXiv.2403.19735

[17] C. Pazhanimuthu, G. Saravanan, K. P. Suresh, R. S. Kumar, *Performance analysis of voltage profile improvement in AVR system using zebra optimization algorithms based on PID controller*, e-Prime-Advances in Electrical Engineering, Electronics and Energy, **6** (2023), 100380. https://doi.org/10.1016/j.prime.2023.100380

[18] L. Praharaj, D. Gupta, M. Gupta, *Efficient federated transfer learning-based network anomaly detection for co-operative smart farming infrastructure*, Smart Agricultural Technology, **10** (2025), 100727. https://doi.org/10.101\6/j.atech.2024.100727

[19] M. Shariatzadeh, M. J. Rostami, M. Eftekhari, *An adaptive image encryption scheme guided by fuzzy models*, Iranian Journal of Fuzzy Systems, **21** (2022), 1-8. https://doi.org/10.22111/ijfs.2023.44875.7915

[20] K. Wang, Y. Fu, X. Duan, T. Liu, *Detection and mitigation of DDoS attacks based on multi-dimensional characteristics in SDN*, Scientific Reports, **14**(1) (2024), 16421. https://doi.org/10.1038/s41598-024-66907-z

[21] K. Wang, Y. Fu, X. Duan, T. Liu, J. Xu, *Abnormal traffic detection system in SDN based on deep learning hybrid models*, Computer Communications, **216** (2024), 183-194. https://doi.org/10.1016/j.comcom.2023.12.041

[22] J. Wang, L. Wang, *SDN-Defend: A lightweight online attack detection and mitigation system for DDoS attacks in SDN*, Sensors, **22**(21) (2022), 8287. https://doi.org/10.3390/s22218287

[23] X. Wang, H. Wang, B. Bhandari, L. Cheng, *AI-empowered methods for smart energy consumption: A review of load forecasting, anomaly detection and demand response*, International Journal of Precision Engineering and Manufacturing-Green Technology, **11**(3) (2024), 963-993. https://doi.org/10.1007/s40684-023-00537-0

[24] W. Yang, Y. Shan, J. Wang, Y. Yao, *An industrial network intrusion detection algorithm based on IGWO-GRU*, Cluster Computing, **27**(6) (2024), 7199-7217.

[25] M. Yue, H. Yan, R. Han, Z. Wu, *A DDoS attack detection method based on IQR and DFFCNN in SDN*, Journal of Network and Computer Applications, (2025), 104203. https://doi.org/10.1016/j.jnca.2025.104203

[26] S. Zavrak, M. Iskefiyeli, *Flow-based intrusion detection on software-defined networks: A multivariate time series anomaly detection approach*, Neural Computing and Applications, **35**(16) (2023), 12175-12193. https://doi.org/10\.1007/s00521-023-08376-5

[27] Z. Zulfiqar, S. U. Malik, S. A. Moqurrab, Z. Zulfiqar, U. Yaseen, G. Srivastava, *DeepDetect: An innovative hybrid deep learning framework for anomaly detection in IoT networks*, Journal of Computational Science, **83** (2024), 102426. https://doi.org/10.1016/j.jocs.2024.102426