Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 47 No. 02 (2026)

Adaptive Cloud Security Framework Based On Deep Reinforcement Learning For Cyber Threat Mitigation

[1]Shyam Sundar M [2]Dr. V. Kathiresan

[1]Research Scholar , Department of Computer Science, AVP College of arts And Science (Autonomous),Thirumurugan Poondi, Chettipalayam, A.Thirumuruganpoondi, Tamil Nadu 641652

[2]Research Supervisor & Associate Professor, Principal, AVP College of Arts and Science

Thirumurugan Poondi, Chettipalayam, A.Thirumuruganpoondi, Tamil Nadu 641652

Abstract: The rapid adoption of cloud computing has transformed modern information technology infrastructures by offering scalable, flexible, and cost-effective services. However, this widespread adoption has also introduced significant security challenges, including data breaches, unauthorized access, and sophisticated cyberattacks. Traditional security mechanisms often struggle to cope with the dynamic and large-scale nature of cloud environments, creating a need for intelligent and adaptive defense systems. In this context, techniques from Artificial Intelligence and deep Learning have emerged as powerful tools for enhancing cloud security. To address these challenges, this paper proposes an intelligent cyber defence framework (ICDF) that leverages techniques from Artificial Intelligence and deep reinforcement learning (DRL) for enhanced cloud security. The proposed approach integrates deep reinforcement learning algorithms to enable autonomous and adaptive decision-making in threat detection and mitigation. Unlike conventional machine learning methods, reinforcement learning allows the system to learn optimal security policies through continuous interaction with the cloud environment. By modeling cybersecurity as a sequential decision-making problem, the system dynamically identifies vulnerabilities, detects anomalies, and responds to cyber threats in real time. In this framework, the deep reinforcement learning agent analyzes network traffic patterns, user behaviors, and system activities to detect malicious actions such as distributed denial-of-service (DDoS) attacks, unauthorized access, and data breaches. The agent continuously improves its performance by receiving feedback in the form of rewards and penalties, thereby optimizing defense strategies over time. This adaptive learning capability enhances the system's ability to respond to evolving and previously unseen cyber threats. Experimental results demonstrate that the proposed AI-driven cyber defence model significantly improves detection accuracy, reduces response time, and enhances resilience against complex attacks compared to traditional security approaches. Overall, this research highlights the effectiveness of combining artificial intelligence with deep reinforcement learning to develop robust, scalable, and self-adaptive cloud security solutions.

Keywords: DDoS attacks, Internet of things, Deep learning, Cybersecurity, Explainable AI and deep reinforcement learning.

1. Introduction

The rapid growth of cloud computing has transformed the modern digital ecosystem by providing scalable, flexible, and cost-effective computing resources for individuals, organizations, and enterprises [1-3]. Cloud platforms support a wide range of services including data storage, virtualization, distributed computing, and real-time application deployment. Despite these advantages, the increasing dependence on cloud infrastructure has also exposed systems to sophisticated cyber threats such as malware attacks, unauthorized access, distributed denial-of-service (DDoS) attacks [4], ransomware, and data breaches. These evolving threats present significant challenges to maintaining secure and reliable cloud environments within the field of Cybersecurity. Traditional cloud security mechanisms mainly rely on static rule-based systems, signature-based

detection, and predefined security policies. Although these approaches can identify known attacks, they often fail to detect newly emerging or adaptive cyber threats in dynamic cloud environments [5-7]. Furthermore, the massive volume of cloud-generated data and continuously changing network behavior make manual monitoring and conventional security systems insufficient for real-time threat mitigation. Consequently, intelligent and adaptive security frameworks are required to improve cloud defense mechanisms.

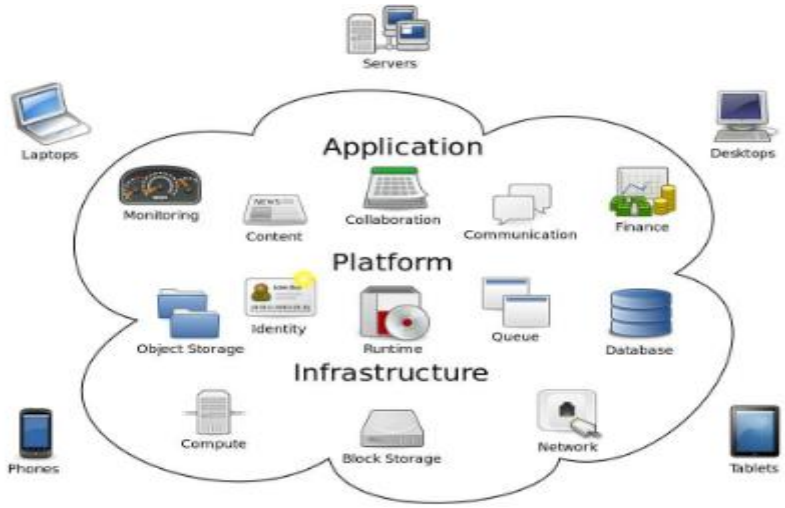


Figure 1. Cloud Computing Architectural Framework

Recent advancements in Artificial Intelligence and Machine Learning have provided promising solutions for enhancing cybersecurity systems [8-10]. AI-driven security models can automatically analyze network traffic, identify abnormal patterns, and make intelligent decisions for attack detection and prevention. Among various machine learning approaches, deep reinforcement learning has emerged as a powerful technique for developing adaptive cybersecurity frameworks capable of learning optimal defense strategies through continuous interaction with the environment [11]. Deep reinforcement learning combines the representation learning capability of deep neural networks with the decision-making mechanism of reinforcement learning. In this approach, an intelligent agent interacts with the cloud environment, observes system states, and performs security actions based on reward-driven learning. By continuously updating its policies through experience, the agent can dynamically adapt to changing attack patterns and optimize defense strategies in real time. This capability makes deep reinforcement learning highly suitable for cloud security applications where cyber threats evolve rapidly and unpredictably.

In this research, an Adaptive Cloud Security Framework Based on Deep Reinforcement Learning for Cyber Threat Mitigation is proposed to enhance the security and resilience of cloud computing environments. The proposed framework utilizes deep reinforcement learning algorithms to monitor cloud activities, detect malicious behaviors, and implement adaptive response mechanisms against cyber threats. The system continuously learns from network conditions and attack scenarios to improve threat detection accuracy and minimize security vulnerabilities. The proposed framework aims to provide intelligent threat mitigation by integrating automated attack detection, dynamic policy optimization, and adaptive response strategies. By leveraging deep reinforcement learning techniques, the system can effectively identify complex attack patterns, reduce false alarm rates, and enhance real-time decision-making capabilities. Furthermore, the adaptive learning capability of the framework enables continuous improvement in security performance without requiring manual rule updates. Overall, the integration of deep reinforcement learning with cloud cybersecurity provides a promising approach for developing next-generation intelligent defense systems. The proposed adaptive framework contributes toward creating secure, scalable, and self-learning cloud infrastructures capable of defending against modern cyber threats in highly dynamic computing environments.

The rest of the research is structured as follows as, section 2 reviews the some of the recent techniques for the detection DDoS using advanced mitigation techniques. section 3 presents the process of the proposed methodology. section 4 provides the results and discussion. section 5 deals with the conclusion and future work.

2. Literature Review

The section 2 reviews the some of the recent techniques for the detection DDoS using advanced mitigation techniques.

Ayeomoni et al [12] developed and evaluated a comprehensive intelligent cyber defense framework integrating multiple ML algorithms including deep neural networks, ensemble methods, and reinforcement learning agents deployed across a heterogeneous cloud testbed comprising 847 virtual machines distributed across three cloud service providers. The system processed 23.6 terabytes of network traffic data over six months, encompassing normal operations and 15 distinct attack scenarios including DDoS, advanced persistent threats, data exfiltration, and zero-day exploits. Our hybrid deep learning architecture combining convolutional and recurrent neural networks achieved 97.3% detection accuracy with only 0.8% false positive rate, substantially outperforming baseline methods (SVM: 89.4%, Random Forest: 91.7%). The reinforcement learning-based automated response system reduced mean time to mitigation from 42 minutes to 3.7 minutes while minimizing service disruption. Explainable AI techniques provided interpretable insights into attack patterns and model decision-making processes, addressing the black-box criticism often leveled at deep learning approaches. Performance analysis demonstrated the framework's scalability, processing 1.2 million transactions per second with sub-100ms latency. This research advances the state-of-the-art in cloud security by demonstrating that AI driven approaches can deliver superior threat detection capabilities, faster response times, and adaptive defense mechanisms while maintaining operational efficiency. The findings hold significant implications for cloud service providers, enterprise security operations centers, and the broader cybersecurity community in developing next-generation intelligent defense systems capable of combating evolving threats in dynamic cloud environments.

Afraji et al [13] developed a Artificial Intelligence (AI) model creation process and any consequent decisions explainable and transparent. The use of deep learning enhances the capability of cybersecurity in handling DDoS attacks and preventing or controlling them. But it has to be a part of a more large-scope platform, based on multiple types of longitudinal or cross-sectional data combined with high efficiency, explainable AI. The article ends with call to proceed with studying and advancing the AI application in response to new threats, and make the most of it to enhance protection of the contemporary networked environment.

Vadisetty et al [14] introduced a AI-based cybersecurity model and cloud optimization and showcases some of the important developments, analytical problems, and future research avenues. We compare the traditional approaches with AI-based solutions and evaluate the effect of AI on performance aspects of response time, fault tolerance, energy efficiency as well as security resilience. Integrating AI in cloud computing helps in operating the infrastructure efficiently, and also enables the cloud infrastructures to be autonomous and self-healing.

Ahmad et al [15] presented a VECGLSTM, an attack detection model integrating Variable Long Short-Term Memory (VLSTM), capsule networks, and the Enhanced Gannet Optimization Algorithm (EGOA), is introduced. This hybrid approach enhances accuracy, reduces false positives, and dynamically adapts to evolving threats. EGOA is employed for its superior optimization capability, ensuring faster convergence and resilience. Additionally, Chaotic Cryptographic Pelican Tunicate Swarm Optimization (CCPTSO) is proposed for privacy-preserving key management. This model combines chaotic cryptographic techniques with the Pelican Tunicate Swarm Optimization Algorithm (PTSOA), leveraging the pelican algorithm's exploration strength and the tunicate swarm's exploitation ability for optimal encryption security. Performance evaluation demonstrates 99.675% accuracy, 99.5175% recall, 99.7075% precision, and 99.615% F1-score, along with reduced training (1.79s), encryption (0.986s), and decryption (1.029s) times. This research significantly enhances CC security by providing a scalable, adaptive framework that effectively counters evolving cyber threats while ensuring efficient key management.

Tyagadurgam et al [16] proposed an intelligent cybersecurity intrusion detection system that can recognize complex attack patterns in network data. To apply a comprehensive data pre-processing pipeline over the CICIDS2017 dataset, it has performed numerical feature extraction, z-score normalization, and one-hot encoding for recognizing multi-class labels and SMOTE for the solution of class-imbalance problem. While its

performance outpaced all others, the Bi-LSTM model improved at capturing both forward and backward time dependencies and obtained A 99% F1-score, a 98.51% accuracy rate, a 99% precision rate, and a 98% recall rate. Training and validation curves indicated strong generalization, and the normalized confusion matrix confirmed high classification accuracy across diverse intrusion types. A comparative analysis showed that the Bi-LSTM model outperformed traditional classifiers such as Naïve Bayes and Deep Multilayer Perceptron, establishing its effectiveness for advanced intrusion detection in intelligent cybersecurity systems. The study offers practical advice for choosing the best IDS models depending on certain network settings and security needs.

Hiregowja Kumara et al [17] developed robust intrusion detection systems capable of discerning between normal network behaviour and potential threats. The methodology used in this study entails the use of multiple deep learning models, including LSTM (Long Short-Term Memory), BiLSTM (Bidirectional Long ShortTerm Memory), and BiLSTM with an attention mechanism. These models are meticulously trained and tested using the KddCup'99 dataset, a benchmark in the field of intrusion detection. The process includes data preprocessing, model training, hyperparameter tuning, and evaluation using metrics like accuracy, precision, recall, and F1-score. The results of this study reveal that the BiLSTM model with an attention mechanism emerges as the most effective solution, achieving an exceptional accuracy of 99%. This model showcases superior performance in accurately identifying network intrusions while maintaining high precision and recall, making it a compelling choice for network protection in cloud environments.

Shaik et al [18] proposed an innovative approach to fortify cloud computing security through the integration of deep learning, with a specific focus on artificial neural networks (ANNs). By harnessing the adaptive capabilities of ANNs, the study aims to detect and mitigate evolving security threats within diverse cloud environments. The research methodology involves the meticulous selection of neural network architectures, comprehensive training datasets, and rigorous evaluations, including considerations for real-world scenarios and dynamic threat landscapes. Results and analysis showcase the effectiveness of the artificial neural network approach, providing nuanced insights into detection accuracy, false positive rates, and response times under various conditions. Moreover, the paper discusses the potential for transfer learning and ongoing adaptation mechanisms to enhance the robustness of the proposed security framework. This contribution adds significant depth to the discourse on cloud security, offering a detailed roadmap for practitioners and decision-makers seeking advanced, adaptive solutions in the face of increasingly sophisticated and dynamic cyber threats. The integration of deep learning, particularly ANNs, emerges as a promising avenue for elevating the security posture of cloud environments in an ever-evolving digital ecosystem.

Aswini et al [19] introduced a novel Hybrid Red Panda Simulated Feature Selection with a Machine Learning-Based Intrusion Detection method for enhancing security in cloud infrastructure. The model collects network traffic data from diverse datasets, including UNSW-NB15, Edge IIoT, TON-IoT, NSL-KDD, Cryptojacking attack time series, and BoT-IoT. To address class imbalance, these datasets are balanced using the Synthetic Minority Over-sampling Technique. The steps taken during preprocessing, such as cleaning the data, applying one-hot encoding, and performing Z-score normalization, is crucial for providing high-quality data. The proposed hybrid optimization method combines Red Panda Optimizer and Simulated Annealing to select optimal features, reducing computational complexity and improving detection efficiency. An Ensemble-based Gradient Boosting Regression Tree is employed for anomaly detection, fine-tuned through grid search to achieve robust performance. To enhance decision-making transparency, Shapley Additive Explanations and Local Interpretable Model-Agnostic Explanations are utilized, offering feature-level and instance-specific insights. A comprehensive evaluation of the proposed framework significantly outperforms existing methods, achieving an accuracy of 99.6% and a precision of 99.35%, demonstrating superior reliability. This work provides a robust and interpretable approach to enhancing cloud security and offers a scalable solution for mitigating cyber threats in diverse cloud environments.

Aldawsari et al [20] deep learning methods—like autoencoders and recurrent neural networks (RNNs)—manage cloud complexity by using past data to learn To detect deviations. In order to improve cloud security, the study looks into the real-world implementation of AI-powered security features such automated incident response, predictive analytics, and self-healing systems. This study offers practical insights for cloud service providers,

security experts, and decision-makers by tackling issues including data privacy, model interpretability, and infrastructure integration and to use AI and ML to improve their cloud security strategy.

Alzu'bi et al [21] introduced a deep learning methodology for detecting and classifying distributed denial of service (DDoS) attacks, addressing a significant security concern within IoT environments. An effective procedure of deep transfer learning is applied to utilize deep learning backbones, which is then evaluated on two benchmarking datasets of DDoS attacks in terms of accuracy and time complexity. By leveraging several deep architectures, the study conducts thorough binary and multiclass experiments, each varying in the complexity of classifying attack types and demonstrating real-world scenarios. Additionally, this study employs an explainable artificial intelligence (XAI) AI technique to elucidate the contribution of extracted features in the process of attack detection. The experimental results demonstrate the effectiveness of the proposed method, achieving a recall of 99.39% by the XAI bidirectional long short-term memory (XAI-BiLSTM) model.

Afraji et al [22] proposed to build superior datasets and use accurate algorithm to improve the security models. This paper focuses on explainability as a way of making the AI model creation process and any consequent decisions explainable and transparent. The use of deep learning enhances the capability of cybersecurity in handling DDoS attacks and preventing or controlling them. But it has to be a part of a more large-scope platform, based on multiple types of longitudinal or cross-sectional data combined with high efficiency, explainable AI. The article ends with call to proceed with studying and advancing the AI application in response to new threats, and make the most of it to enhance protection of the contemporary networked environment.

3. Proposed Methodology

This research work proposes an intelligent cyber defence framework that leverages techniques from Artificial Intelligence and deep reinforcement learning for enhanced cloud security. The proposed approach integrates deep reinforcement learning algorithms to enable autonomous and adaptive decision-making in threat detection and mitigation. Unlike conventional machine learning methods, reinforcement learning allows the system to learn optimal security policies through continuous interaction with the cloud environment. By modeling cybersecurity as a sequential decision-making problem, the system dynamically identifies vulnerabilities, detects anomalies, and responds to cyber threats in real time. In this framework, the deep reinforcement learning agent analyzes network traffic patterns, user behaviors, and system activities to detect malicious actions such as distributed denial-of-service (DDoS) attacks, unauthorized access, and data breaches. The agent continuously improves its performance by receiving feedback in the form of rewards and penalties, thereby optimizing defense strategies over time. This adaptive learning capability enhances the system's ability to respond to evolving and previously unseen cyber threats.

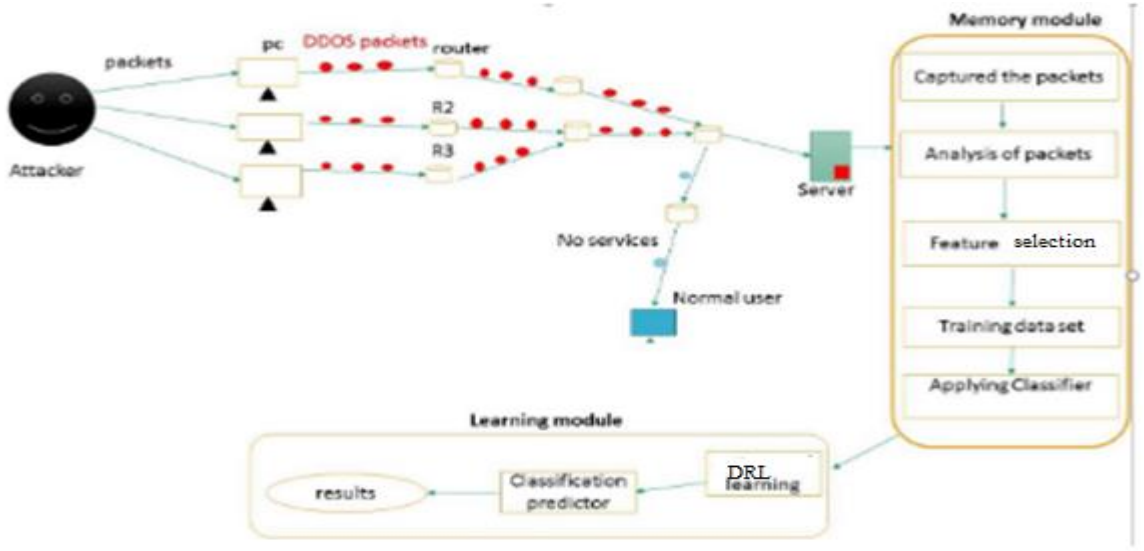


Figure 1. The process of the proposed methodology

3.1. System Model and Problem Formulation

The intelligent cyber defence framework is designed for SDN-based cloud environment. It comprises four modules: data preprocessing, feature extraction, anomaly score prediction, and classification. The SDN controller employs this approach and monitors each switch individually for DDoS attack traffic irregularities. The SDN agents at switches perform data preprocessing to convert the raw traffic from network flows and process it for normalized data. Using Recursive Feature Elimination (RFE), relevant features (Source IP address, Destination IP address, and times-tamp) are chosen. Then, using time series techniques, any malicious traffic by releasing anomaly scores. Finally, the final anomaly scores to classify the traffic sample as normal or DDoS. Each module is briefly explained below.

The first and major task is Information Gathering, information gathering is a process in which find out the different vulnerabilities of the victim machines in order to attack at victim site. Information Gathering involves all the information about the ongoing services, open or closed ports and other weaknesses. In this case attacker might get information about the weak points of victim and can attack conveniently. Every service that cloud computing provides, have specific port number like: http uses port no 80, 20 to 23 is being used by TCP and UDP performing different functionalities, ftp runs on 990 but sometimes on 21 as well and moreover. Summing up, information gathering is such a process which provides all the relevant information about the vulnerabilities of any system so that attacker might attack accordingly. Nmap scanner is a tool, which is used for the purpose of information gathering. it only needs an IP address of the victim machine, once it has that IP, it starts scanning the whole system that shows different activities, different services that another system is providing, open and closed ports thus all those activities which are going on can be shown up even though operating system which is used by another system would also be shown. Once the opened port is found. An attack would be generated, in this case using Distributed denial of service attack, which includes further different attacks like ping of death, DDOS is an extremely worst attack that ruin all the services of the system. DDOS apply million number of packets that would be merely intolerable and make the all services down. ParrotSec have a mechanism, it includes shell or terminal through which all the operations can be performed by applying commands as like other operating systems such as Kali and Ubuntu, ParrotSec is also command line interface and by giving commands ParrotSec perform all the operations thus at terminal, the command PING IP would be applied and thus more than 65000 packets would be sent to the victim site and all the services became down. This is how attack would be generated. Next phase is detection phase, In this scenario, cloud-based website would be targeted, Nmap would scan the all vulnerabilities of our target website in this way it's all abnormalities would bring into the scene. Once the weakened ports are shown, a python-based script would be generated which includes DDoS. Once DDoS do attack on the specific website it may brought their services down onto their knees. Next phase is detection phase, Sniffing is the procedure, which monitors and capture every packet during transmission. It monitors all the inbound and outbound traffic on the internet and make analysis of each packet. Sniffing overall sniff every packet in order to analyses it properly. In this case, sniffing would be done at server side. There are many tools for sniffing purposes but most of the important one is Wireshark. Wireshark do sniff each and every packet deeply. After the overall analysis of packets, a large data set has been created which implies at classifier.

3.2. Data preprocessing

In machine learning, data preprocessing is crucial in generating accurate and valuable results [23]. Data preprocessing improves data quality by handling missing or incomplete data, smoothing out noise, and addressing discrepancies. The following steps are involved in the preprocessing stage:

1. The correlated features get removed by selecting only one feature among many with a > 80% correlation. The Pearson correlation coefficient is employed, which gives a value between -1 and + 1 and can determine if two features have a linear relationship. The covariance of two features (p, q) is calculated using Eq. (1), where the cov (p, q) represents the covariance between two features. In contrast, σp and σq repre- sent the standard deviation of p and q, respectively.

$$\rho(p,q) = \frac{cov(p,q)}{\sigma_p \sigma_q} = \frac{E[(p-E[q])(q-E[q])]}{\sigma_p \sigma_q} \quad (1)$$

2. To remove any incomplete data, we need to eliminate the rows that have missing values.

3. To replace infinite values with a maximum feasible value.

4. The data is normalized using the min-max scaling method, which involves applying the equation specified in Eq. (2). Here, z represents the value of a feature fe, while z' denotes the corresponding normalized feature value. The minimum and maximum values of the feature are denoted as min fe , max fe , respectively.

$$z' = \frac{z - min_{fe}}{max_{fe} - min_{fe}} \qquad (2)$$

5. The label encoding technique converts the categorical label column into a binary numerical array. Here DDoS is assigned the value of 1, while normal is assigned the value of 0.

3.3. Feature Selection

Following the preprocessing phase, we perform feature selection on normalized data using the hybrid RFE approach described in the current paper. As DDoS attack patterns are in huge and high dimensional data overfitting, poor model interpretability, and longer calculation times are all possible consequences associated with it. The effectiveness of DDoS detection algorithms can be enhanced by using RFE to pick a subset of the most useful characteristics, thereby lowering the dimensionality. RFE constructs a model and selects the optimal or worst features based on their ranks, using a basic DT method as an estimator. This method employs information entropy as a crucial measure for feature selection. It computes the information gain for each sample to divide it layer by layer until at least one sample type is separated. Based on the threshold, the source IP address and destination IP address have the highest ranking of all features and are selected for our work on DDoS detection. During a DDoS attack, the number of flow entries with Unique Source IP Address (USIA) may grow due to fake and randomly produced IP addresses. In contrast, the number of Normalized Unique Destination IP Address (NUDIA) may not vary much compared to usual. Still, the normalized value of this statistic concerning the total number of packets in the flow table decreases. These two features are, therefore, independently used as time series in the attack detection procedure to identify potential cases of a DDoS attack. In addition to these two features, timestamps and class labels are also considered in this work, as they are associated with IPs. The feature list considered for our work is tabulated below in Table 1.

Table 1. features selected

Sno	Feature name
1	Unique Source IP Address (USIA)
2	Normalized Unique Destination IP Address (NUDIA)
3	Timestamp
4	Label

3.4. Anomaly score prediction module

In this phase, separately analyze USIA and NUDA features to determine their anomaly scores (score 1 and score2) at time t. The anomaly prediction module is illustrated in Fig. 2. We apply the USIA feature to both the ARIMA and chaos theory methods to obtain score 1 . For the NUDA feature, we pass it through exponential filters and dynamic threshold to get score2 .

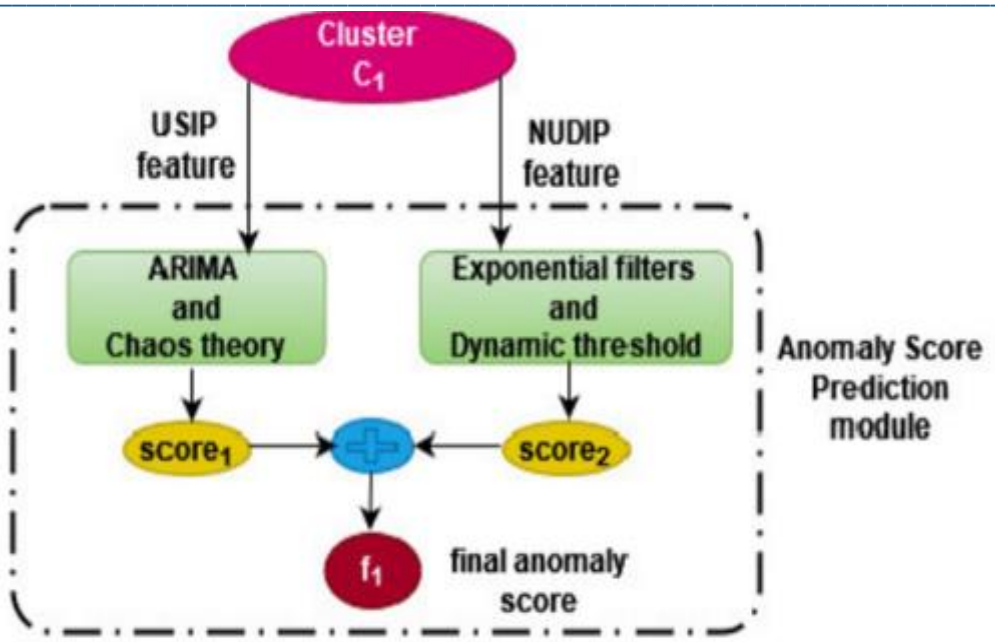


Figure.2. Anomaly score prediction module

3.4.1. Processing of USIA feature

As said earlier USIA is passed as time series to ARIMA. ARIMA (p, d, q), is a three-tuple time-series forecasting statistical model, where p is the lag order, d denotes the number of times raw observations differenced, and q is the order of Moving Averages (MA) or lagged forecast errors as seen in Eq. 3.

$$\boldsymbol{Z'(t) = c + \varphi_1 \times Z'_{(t-1)} + \cdots \ldots + \varphi_p \times Z'_{(t-p)} + \theta_1 \times \varepsilon_{(t-1)} + \cdots + \theta_q \times \varepsilon_{(t-q)} + \varepsilon_t} \quad (3)$$

In Eq. (3), the term Z′(t) is a time series, φ1 and theta1 are the first Auto Regression (AR) and MA terms, and p and q are the order of AR and MA terms, respectively, and finally, ε t is the error. ARIMA captures the trends and seasonality of the network traffic and allows checking for any spikes and fluctuations in the traffic. Spikes relate to abnormal traffic. To find an inaccuracy in prediction error, we use the Lyapunov exponent, as depicted in Eq. (4) below:

$$\lambda t = 1/tln\left(\left|\frac{pt}{p0}\right|\right) \quad (4)$$

where p0, p t and λ t represent the first prediction error, the tth prediction error, and the Lyapunov exponent at the t th instance, respectively. Positive exponents imply DDoS traffic, while negative exponents indicate regular traffic.

According to algorithm 1, the ARIMA model estimates the attack trend of the sample set Z, where z t is an exponential function of time t. To construct the ARIMA model, set Model 1 to be true. For training the model, n samples of source IPs are stored in Z. If Z is non-stationary, apply differencing d > 1 to achieve a stationary time series. Differentiating the time series makes Z suitable for stationary time series analysis and modeling. The Box-Cox transformation stabilizes the variance of a time series variable y. The Box-Cox transformation is a mathematical technique that adjusts the data distribution to make it more suitable for analysis and modeling. It is also possible to use either Akaike's information criterion, the corrected version of this criterion (AICc), or the Bayesian Information Criterion (BIC) to select the order model. By minimizing these criteria, the best model was selected. The peaks in the density plot specify the anomalies. The ARIMA model generates the standard feature pattern, but no attack instances should occur during the model generation. After model generation, the Model1 flag is set to FALSE, and the training phase is completed. In the testing phase normal model estimates the value, ẑt, for each subsequent incoming traffic sample, z t. If any attack traffic is coming the model predicts an abnormal

behavior by generating the spikes. The prediction error's chaos calculates an anomaly score. The prediction error p t is determined using Eq. (5)

$$pt = |(z_t - \hat{z}_t)| \quad (5)$$

To assign an anomaly score for different outcomes of prediction errors, the Lyapunov exponent (λ) is used. According to Eq. (5), a positive value of λ indicates attack traffic (score 1 = 1), while a negative value suggests normal traffic (score 1 = 0).

Algorithm 1 Prediction of Anomaly score1

INPUT: *USIA feature set* $\{z_1, z_2, \ldots, zt\}$
OUTPUT: *prediction of* $score_1$

// Training phase
Set $Model_1$ *= TRUE*
$z_{count} = 0$;
$Z \leftarrow \{z_1, z_2, \ldots, z_t\}$ *// USIA feature*
At a time instance t
  z_{count} ++
  Train ARIMA $_{(p, d, q) for}$ Z
  After completion of the training phase set $Model_1$ *= FALSE*
// Testing phase
  $\hat{z}_t \leftarrow$ *ARIMA* $(p, d, q)(\{z_{t-p-d}, \ldots, z_{t-1}\})$
  $p_t \leftarrow mod\ (z_t - \hat{z}_t)$ *// Estimate prediction error*
  calculate Lyapunov exponent λ_t
  if $\lambda_t \leq 0$, *then*
    Normal traffic (score = 0)
  else
    DDoS traffic (score = 1)
  end if
end

3.4.2. Processing of NUDA feature:

Here exponential smoothing forecasting model is used, which gives weights to the earlier and new observations for forecasting. New observations have higher weight than earlier observations based on the smoothing constant α, which makes things look smoother. The value of α is between 0 and 1 as per Eq. (6) where E i signifies the smoothed data and x denotes the original data.

$$E_t = \alpha . E_{t-1} + (1 - \alpha) . Z_t \quad (6)$$

In algorithm 2, the n samples of the NUDIA feature are used and stored in Y as a time series to generate the model. Y is estimated by two exponential filters, f1, and f2, with their exponential constants α1 as 0.1 and α2 as 0.8 and their absolute difference stored in Ad f. The rolling median generates a median time-series, M. The least distance between each case and the remaining samples is determined and stored in a set, ld. The mean μ ld and standard deviation σ ld are computed. Model 2 is set to False once the above-stated values are determined. Now for each y t feature of upcoming traffic, the process mentioned above is repeated, and the least distance ld t is calculated. If it is less than the threshold value η = μ ld + q*σ ld, the traffic instance is considered normal (score2 = 0); otherwise, it is abnormal (score2 = 1).

Algorithm 2 Prediction of anomaly score2 using smoothing filters

INPUT: *NUDIA feature set Y*
OUTPUT: *Anomaly score*$_2$
Model$_2 \leftarrow$ *True*
$y_{cnt} \leftarrow 0$
$Y = \{y_1, y_2, y_3, \ldots y_t\}$
At time t
  y_{cnt}++
  If $y_{cnt} \geq 1$
    Initialize and assign two filters f1, f2 with Y[1]
    From y[2:] find
    $f1_i = \alpha_1 f1_{i-1} + (1 - \alpha_1).y_i$
    $f2_i = \alpha_2 f1_{i-1} + (1 - \alpha_2).yi$
    Calculate the Absolute difference AD for the output of two filters,
    Followed by median time series M = median $(AD_i, \ldots AD_{i+w})$
    Find the least distance ld and its mean and SD
    calculate threshold $\eta = \mu_m + q * \sigma_m$
  end if
  find $f1_t, f2_t, AD_t, m_t, ld_t$ \\ *testing phase*
  if $ld_t < \eta$ *then*
    Normal traffic (score$_2 = 0$*)*
    else
    Attack traffic (score$_2 = 1$*)*
  end if
end

The anomaly score prediction module collects score1 and score2 from the above methods and performs AND ing operation to obtain the final anomaly score f. All the collected final scores are fed to next the module for DDoS detection.

3.5. Classification using Deep Reinforcement Learning

The utilization of a rule-based method for network event correlation is very important in the identification of DDoS assaults in network settings. The methodology encompasses the gathering of data from multiple network nodes, performing preprocessing to assure uniformity, and afterward using predetermined correlation rules specifically designed to detect patterns indicative of DDoS attacks. These rules look at the spatial, temporal, and rate-based parts of network traffic, keeping an eye out for sudden traffic spikes, strange protocols, or high resource usage. A rule triggers an alert describing the nature and severity of potential DDoS activity. This alert initiates further research to protect network resources. These rules are updated based on real-world incidents and emerging threats to provide proactive and adaptive DDoS detection and prevention. Due to the event correlation, detecting the attack traffic too early with reasonable accuracy is possible, which may reduce the economic loss and huge damage to resources in the cloud network.

According to algorithm 3, calculate the threshold η for classifying abnormal and normal traffic. The rule-based classifier function calculates the sum of final anomaly scores for all clusters and checks if it exceeds the threshold to determine the traffic type. The main loop simulates the continuous processing of incoming traffic samples. Inside the loop, anomaly scores are calculated for each cluster and collected in the cluster scores list. If abnormal traffic is detected, an alarm is raised by the controller. The corresponding IP address and its corresponding switch are added to the discarded list. A defense mechanism can stop a DDoS attack but also stop any packets sent to a victim's IP address. As a result, immediately after the attack, the controller must change the activity of the flow entries.

Input Specification: State$_i$, which is made up of host factors including CPU, RAM, bandwidth, and disk utilization and capacity, is the input for the scheduler Model [24]. The host Million Instructions PerSecond

(MIPS), response time, cost per unit time, power features, cost per hour, cost per minute, and number of tasks assigned to this host are also listed. The computational (CPU), memory (RAM), and I/O (disk and bandwidth) capabilities of different hosts would vary. Such factors are essential for scheduling decisions because tasks in an edge-cloud environment compute, memory, and I/O limits. Additionally, by hiding the hosts that have no tasks, allowing several tasks to be assigned to a compact group of hosts could guarantee low energy use. Faster disk read/write rates on the host could enable the completion of I/O-intensive operations and avoid SLA violations. In the feature vector referred to asFV_i^{Hosts}, each of these parameters is specified for every host. The assignments in a_i are divided into: $n_i$and $a_{i-1} \backslash l_i$, two separate groups. The first set of parameters includes the task CPU, RAM, bandwidth, and storage space needs.

Output Specification: Depending on the input State$_i$, proposed model needs to assign hostsfortasks in a_i in the beginning of interval SI_i, and results referred $Action_i$ include host assignments for new tasks n_i and migration decisions for tasks from previous periods that are still in progress $\in a_{i-1} \backslash l_i$.Each task is transferred must be migratableto the new server asm_i which is $\subseteq a_i$ according to the feasibility criteria. Additionally, whenever a host h is assigned to a particular task T, h should not become overloaded after allocation, i.e., h is appropriate for T. As a result, $Action_i$ by Equation (2) in such a way that it applies to the intervalSI_i, $\forall T \in n_i \cup m_i, \{T\} \leftarrow Action_i(T)$,

$$Action_i = \begin{cases} h \in Hosts \forall t \in n_i \\ h_{new} \in Hosts \forall t \in m_i \; if \; t \; is \; to \; be \; migrated \end{cases} \quad (7)$$

appropriate for $t \; \forall \; t \in n_i \cup m_i$is subject to Action$_i$. Neural networks may produce host-task allocation preferences. This indicates that the model offers an ordered list of hosts instead of one for every task. Additionally, a penalty is applied to uncontrolled behavior. Specifically, this captures two features of punishment: (1) the percentage of tasks that the simulation tried to transfer but was unable to do so is known as the migration penalty; and (2) the number of hosts who were granted greater priority but were unable to complete a task is added for each task to determine the attack.

- **Entropy sequence formation based data distribution**

The entropy is a measure of the randomness and uncertainty of data distribution. To compute the entropy sequence of the malware, first divide the raw bytes of the malware into continuous data blocks (the data are represented in hexadecimal: 00h-FFh), then compute the entropy of each block, and finally connect the entropy of each block according to the order of the blocks to form the entropy sequence. The value range of each byte is [0, 255]. It is crucial to ensure that all data in the block can be used to compute the entropy value. So, the size of the block is set to 256. When computing the entropy sequence, if the length of the last block is less than 128 bytes, the block will be discarded. Otherwise, the block is supplemented with data zero to make its length reach 256. For each block, the method of computing entropy is as follows:

$$H(x) = -\sum_{i=0}^{255} p(x_i) * log2p(x_i) \quad (8)$$

where *xi* represents a specific raw byte value and *pi* represents the probability (frequency) of this value in the block, *H*(*x*) represents the entropy value of the block, and the range of its value is zero to eight. When all bytes in the block are equal, the value of entropy is zero. If all the values in the block are different, the value of entropy is eight. If the raw byte of the malware is divided into N blocks, we represent the entropy sequence as *Hs* = *h*1*, h*2*, hn*. It can be seen that the entropy sequences of the same malware family samples are very similar, the entropy sequence distributions of different family samples are quite different. Thus the proposed model efficiently identify the malware attacks.

Algorithm 3. Detection of DDoS

```
Input: Final Anomaly Scores from different clusters
# Define a function for event correlation
    1. def Rule based classifier (cluster_scores):
# Calculate the sum of anomaly scores for all clusters
    2. total_score = sum(cluster_scores)
# Check if the total_score is greater than the threshold
    3. if total_score > threshold:
    4. return "Abnormal Traffic"
    5. else:
    6. return "Normal Traffic"
# Main loop to process incoming traffic samples
    7. while True:
    8. incoming_traffic = receive_traffic_sample()  # Receive a new traffic sample
# Process the traffic sample and obtain anomaly scores for each cluster
    9. cluster scores = []
    10. for cluster in clusters:
    11. score1, score2 = calculate_anomaly_scores(incoming_traffic, cluster)
    12. final_score = score1 AND score2  # Perform AND operation as described
    13. cluster_scores.append (final_score)
    14. calculate threshold η based on mean and standard deviation of final scores.
# Perform event correlation to determine traffic type
    15. traffic_type = Rule based classifier(cluster_scores)
```

Acknowledging the increasing presence of multiple controllers, switches, and routers as the network expands, the framework is tailored to accommodate the network's growth. Its design leverages the hierarchical structure of SDN-based cloud networks, enabling robust event correlation at different network levels. Event correlation ensures early and accurate detection of malicious traffic, effectively reducing false positives. By proactively addressing scalability concerns and adapting to complex, multi-tiered network infrastructures, the framework demonstrates its capacity to maintain efficiency and accuracy even in the face of substantial network expansion.

4. Results and Discussion

This section offers a detailed analysis of the findings comparing the model to a number of industry standard methodologies. Also, describes the experimental setup, evaluation metrics, and data collection. CloudSim is used to enable edge node properties including reaction time, cost, and power. For the constraint fulfillment module, new software as well as input and output preprocessing have been developed. The loss function is computed using CloudSim performance tracking and storage service.

- **Dataset Details**

The dataset used to evaluate the proposed approach is CICDDoS2019 which the Canadian Institute gave for Cybersecurity [25]. This dataset is the naive dataset with more modern attacking methods. Reflection and exploitation attacks are the most common types of attacks in the dataset. These attacks mask the intruder's identity by sending packets to servers from the target IP address, causing the target victim's bandwidth to become overburdened with response packets. The dataset is composed of 88 features. It provides 12 types of DDoS attacks, namely NTP, DNS, LDAP, NetBIOS, UDP, UDP-Lag, SSDP, SYN, TFTP, SNMP, MSSQL, and Web DDoS [26]. Considering the experimental configuration's network infrastructure, the interval was t = 1 min. The following results are based solely on examining the dataset CICDDoS 2019. Overall, 500 traffic samples were employed for our experiments. The first 200 samples train the network's normal behavior, while the remaining 300 test it.

- **Performance Evaluation**

To evaluate the performance of the proposed method, implemented a prototype system. The system is programmed in JAVA. To ensure the accuracy and reliability of the experimental results, this work uses a 10-fold cross-validation method. In the experiment, the dataset is divided into ten parts. In each test process, nine parts are selected as the training set, and the remaining one part is used as the test set. This proposed work adopts the external quality metrics as Accuracy, Recall, Precision, F-measure.

Precision is defined as the ratio of correctly found positive observations to all of the expected positive observations.

$$\text{Precision} = TP/(TP+FP) \quad (9)$$

Recall is defined the ratio of correctly identified positive observations to the over-all observations.

$$\text{Recall} = TP/(TP+FN) \quad (10)$$

F1 score is defined as the weighted average of Precision as well as Recall. As a result, it takes false positives and false negatives.

$$\text{F1 Score} = 2*(\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \quad (11)$$

Accuracy is calculated in terms of positives and negatives as follows:

$$\text{Accuracy} = (TP+FP)/(TP+TN+FP+FN) \quad (12)$$

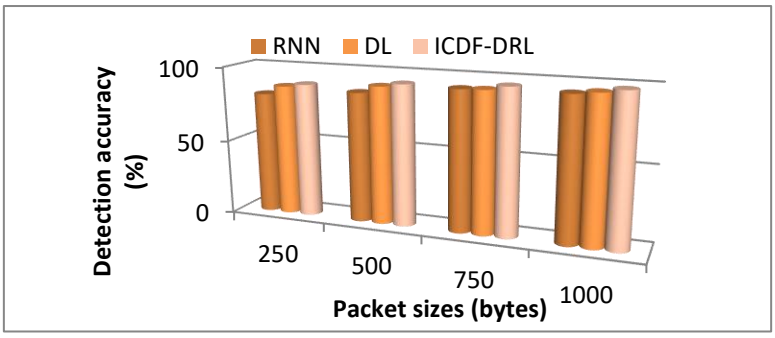


Figure 3. Detection accuracy of the proposed ICDF-DRL and existing methods for malware detection

From the figure 3. it is identified that the proposed ICDF-DRL method has high detection accuracy as 98.24% for the packet size is 1000 bytes than the existing DL and RNN based techniques. Thus the result explains that the proposed ICDF-DRL method is greater to the existing algorithms in terms of better detecting results with high accuracy rate. ICDF-DRL learning methods are quite robust to noise in the training data and thus leads to attain better accuracy rate with avoiding local optima problem. These results further confirm that the proposed method can more effectively and more stably recognize the piracy software.

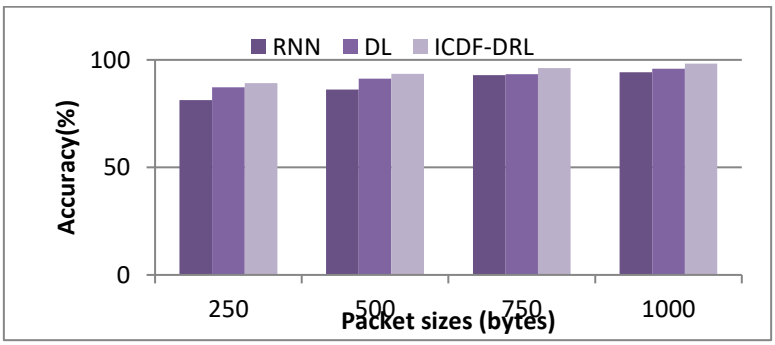


Figure 4.Accuracy comparison between the proposed and existing machine learning methods for malware detection

It can be seen from Figure 4. that the accuracy comparison between the proposed and existing method for malware detection. The proposed method ICDF-DRL has high accuracy rate as 98.14% where as the DL method has 97.46% and the DRL method has 85.71%.

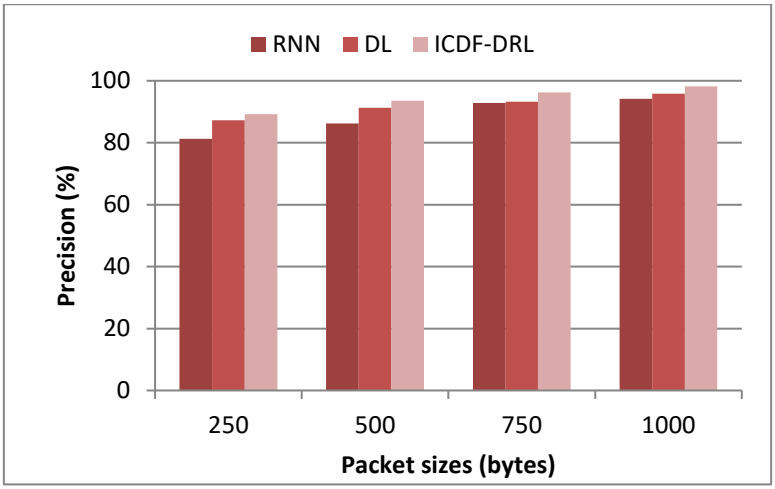


Figure 5. Precision comparison between the proposed and existing machine learning methods for malware detection

The figure 5. illustrate the precision comparison results between the proposed and existing methods. There are two classes, i.e. malicious and benign used in malware dataset. Overall, 90% of classes are predicted for malicious files, with 10% miss classifications error. Figure 5. shows that the proposed malware detection method outperforms as compared to two existing machine learning-based works in terms of precision performance. From the result is noted that the proposed ICDF-DRL method has high precision results than the existing methods.

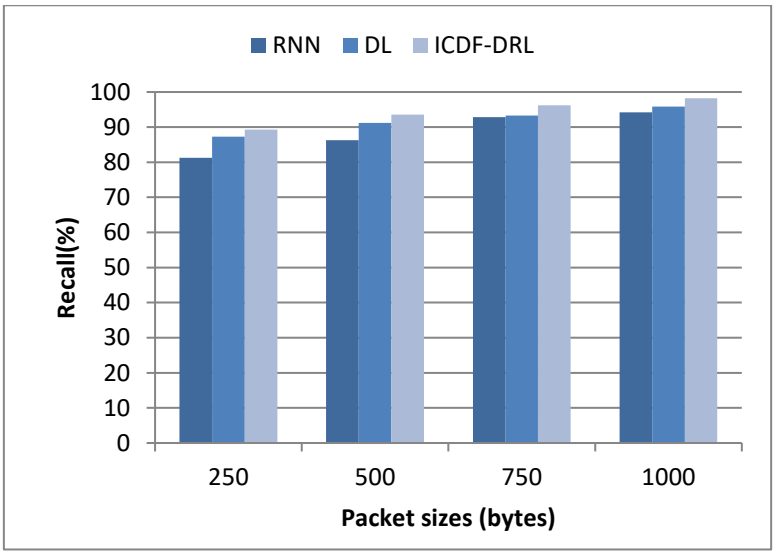
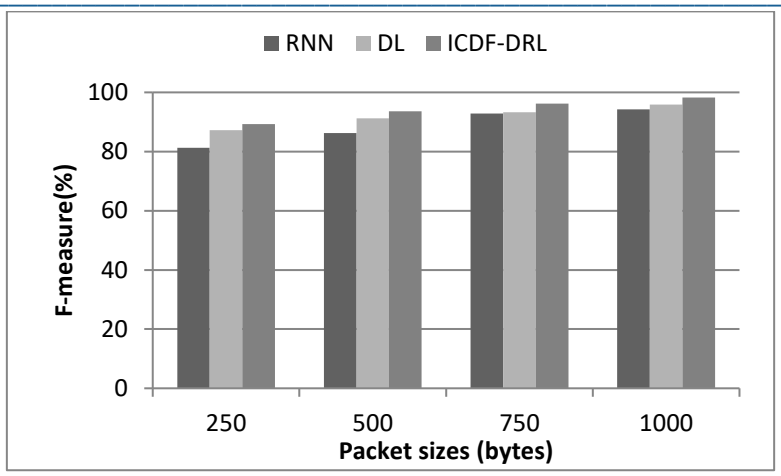


Figure 6. Comparison result of recall between the proposed and existing machine learning methods for malware detection

The figure 6. illustrate the recall comparison result between the proposed and existing method for malware detection. from the figure it is identified that the proposed method has high recall results than the existing SVM, Tensor flow based methods. The proposed method ICDF-DRL has high recall rate as 94.68% where as the Tensor flow method has 91.47% and the SVM method has 85.24%when the image ratio is 229*229. On the other hand, the proposed method ICDF-DRL has high recall rate as 91.67% where as the DL method has 85.29% and



the RNN method has 71.24%.

Figure 6. Comparison result of F-measure between the proposed and existing machine learning methods for malware detection

The figure 6. illustrate the f-measure comparison result between the proposed and existing method for malware detection. From the figure it is identified that the proposed method has high recall results than the existing RNN, DL methods. The proposed ICDF-DRL method has high f-measure rate as 95.67% where as the Tensor flow method has 92.68% and the SVM method has 87.25%when the image ratio is 229*229. On the other hand, the proposed ICDF-DRL method has high f-measure rate as 92.84% where as the DL has 86.24% and the RNN method has 81.27%.

5. Conclusion

This research work presented an adaptive cloud security framework based on deep reinforcement learning for effective cyber threat mitigation in cloud computing environments. With the rapid growth of cloud services and increasing sophistication of cyberattacks, traditional static security mechanisms are no longer sufficient to protect dynamic cloud infrastructures. To address these challenges, the proposed framework integrated techniques from Artificial Intelligence and deep reinforcement learning to develop an intelligent and self-adaptive cybersecurity solution. The proposed framework continuously monitored cloud network activities, analyzed system behavior, and dynamically identified potential cyber threats such as malware attacks, unauthorized access, and distributed denial-of-service attacks. By utilizing deep reinforcement learning, the intelligent agent learned optimal defense strategies through continuous interaction with the cloud environment. This adaptive learning capability enabled the system to improve its decision-making process over time and respond effectively to evolving attack patterns. The integration of deep neural networks with reinforcement learning enhanced the framework's ability to process large-scale cloud data, detect anomalies accurately, and reduce false alarm rates. Furthermore, the proposed system improved real-time threat detection and automated mitigation processes, thereby increasing the overall resilience and reliability of cloud infrastructures. Experimental analysis demonstrated that the adaptive framework achieved better detection accuracy, faster response time, and improved mitigation efficiency compared to conventional cybersecurity approaches. The dynamic policy optimization mechanism also contributed to enhanced scalability and adaptability in highly distributed cloud environments. Overall, the proposed deep reinforcement learning-based cloud security framework provides a promising solution for intelligent cyber defense in modern cloud computing systems. The research highlights the potential of adaptive AI-driven security models in addressing complex and continuously evolving cyber threats. Future work may focus on integrating federated learning, blockchain security, and explainable artificial intelligence techniques to further improve privacy, transparency, and robustness in cloud cybersecurity systems.

REFERENCES

1. Devi, T. A., & Jain, A. (2024, May). Enhancing cloud security with deep learning-based intrusion detection in cloud computing environments. In *2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)* (pp. 541-546). IEEE.
2. Mohamud, A. H., Khalif, A. A., Hirsi, A., Ahmed, M. E., Mohamed, J. I., Aden, M. A., ... & Ahmed, S. (2026, January). Machine Learning for Cyber Threat Detection in Cloud Computing: Trends, Challenges, and Future Directions. In *2026 1st International Conference on Innovations in Information and Communication Technologies (IICT)* (pp. 1-6). IEEE.
3. Dondapati, K., Deevi, D. P., Allur, N. S., Chetlapalli, H., Kodadi, S., & Perumal, T. (2022). Strengthening cloud security through machine learning-driven intrusion detection, signature recognition, and anomalybased threat detection systems for enhanced protection and risk mitigation. *International Journal of Engineering Research and Science & Technology*, *18*(1), 88-102.
4. Sharma, H. (2024). The role of artificial intelligence and machine learning in strengthening cloud security: A comprehensive review and analysis. *International Journal of Advanced Research in Computer and Communication Engineering*, *13*(8), 36-44.
5. Mahalakshmi, K. V., Neha, S., Abdullah, A., & Daniel, A. Enhancements in security for cloud computing: Utilizing deep learning and machine learning. In *Progressive Computational Intelligence, Information Technology and Networking* (pp. 800-805). CRC Press.
6. Ch, R., Naresh, B., Prasanna, D. L., Radhika, E., Chander, N., & Kolakar, A. (2025, August). Enhancing Cloud Security Through a Comprehensive Survey of Machine Learning Based Intrusion Detection Systems. In *2025 International Conference on Applications of Machine Intelligence and Data Analytics (ICAMIDA)* (pp. 1-5). IEEE.
7. Kharbanda, N. S. (2024). Comparative Review of Supervised vs. Unsupervised Learning in Cloud Security Applications. *Int. Res. J. Eng. Technol*, *11*(9).
8. Ahmad, W., Rasool, A., Javed, A. R., Baker, T., & Jalil, Z. (2021). Cyber security in iot-based cloud computing: A comprehensive survey. *Electronics*, *11*(1), 16.
9. Abdullayeva, F. (2023). Cyber resilience and cyber security issues of intelligent cloud computing systems. *Results in Control and Optimization*, *12*, 100268.
10. Mykhaylova, O., Korol, M., & Kyrychok, R. (2024). Research and analysis of issues and challenges in ensuring cyber security in cloud computing. *Cybersecurity Providing in Information and Telecommunication Systems II 2024*, *3826*, 30-39.
11. Prathima, C., Vyakaranam, V. C., Rajendran, R. K., Padmaja, N. N., Bharathi, M., & Balaji, V. (2025). Applications of the convergence of cyber security and cloud computing. In *Convergence of Cybersecurity and Cloud Computing* (pp. 37-52). IGI Global Scientific Publishing.
12. Ayeomoni, O. (2024). Intelligent Cyber Defense: Leveraging AI and Machine Learning Algorithms for Cloud Security. *Applied Sciences, Computing, and Energy*, *1*(1), 246-275.
13. Afraji, D. M. A. A., Lloret, J., & Peñalver, L. (2025). Deep learning-driven defense strategies for mitigating DDoS attacks in cloud computing environments. *Cyber Security and Applications*, *3*, 100085.
14. Vadisetty, R., Polamarasetti, A., Guntupalli, R., Rongali, S. K., Raghunath, V., Jyothi, V. K., & Kudithipudi, K. (2022). AI-Driven Cybersecurity: Enhancing Cloud Security with Machine Learning and AI Agents. *Available at SSRN 5284922*.
15. Ahmad, S., Arif, M., Mehfuz, S., Ahmad, J., & Nazim, M. (2025). Deep Learning-based cloud security: innovative attack detection and privacy focused key management. *IEEE Transactions on Computers*.
16. Tyagadurgam, M. S. V., Gangineni, V. N., Pabbineedi, S., Penmetsa, M., Bhumireddy, J. R., & Chalasani, R. (2022). Designing an Intelligent Cybersecurity Intrusion Identify Framework Using Advanced Machine Learning Models in Cloud Computing. *Universal Library of Engineering Technology*, (Issue).
17. Hiregowja Kumara, S. (2025). *Machine Learning Driven Network Protection in Cloud Computing Environments* (Doctoral dissertation, Dublin, National College of Ireland).
18. Shaik, S. P. (2021). Enhancing Cloud Computing Security Through Deep Learning: An Artificial Neural Network Approach.

19. Aswini, J., Rekha, K. S., Rosaline, R. A. A., & Sivaneshkumar, A. (2025). Enhancing security in cloud computing systems using hybrid feature selection and ensemble-based machine learning for intrusion detection. *Evolving Systems*, *16*(3), 101.
20. Aldawsari, H., & Kouchay, S. A. (2023). Integrating AI and Machine Learning Algorithms in Cloud Security Frameworks for Enhanced Proactive Threat Detection and Mitigation. *Journal of Emerging Threat Management*.
21. Alzu'bi, A., Albashayreh, A., Abuarqoub, A., & Alfawair, M. A. (2024). Explainable AI-based DDoS attacks classification using deep transfer learning. *Computers, Materials & Continua*, *80*(3), 3785-3802.
22. Afraji, D. M. A. A., Lloret, J., & Peñalver, L. (2025). Deep learning-driven defense strategies for mitigating DDoS attacks in cloud computing environments. *Cyber Security and Applications*, *3*, 100085.
23. Daraghmeh, M., Agarwal, A., & Jararweh, Y. (2023, December). Cloud workload categorization using various data preprocessing and clustering techniques. In *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing* (pp. 1-10).
24. Li, S. E. (2023). Deep reinforcement learning. In *Reinforcement learning for sequential decision and optimal control* (pp. 365-402). Singapore: Springer Nature Singapore.
25. Sharafaldin I, Lashkari AH, Hakak S, Ghorbani AA (2019) Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," In 2019 International Carnahan Conference on Security Technology (ICCST), pp. 1–, IEEE
26. Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A (2019) A survey of network-based intrusion detection data sets. Comput Secur 86:147–167