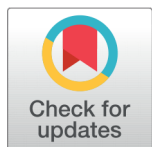


ORIGINAL ARTICLE

 OPEN ACCESS

Received: 22/06/2026

Accepted: 20/07/2026

Published: 05/08/2026

Citation: Mangaiyarkarasi V, Malathi S (2026) Enhancing Security in Software-Defined Networking from ARP Spoofing Attack Using Adaptive Encryption-Assisted Explainable Mish-Activated Recurrent Neural Network Framework. Indian Journal of Science and Technology 19(29): 2050-2061. <https://doi.org/10.17485/IJST/V19i29.942>

* Corresponding author.

visitmalathi@gmail.com**Funding:** None**Competing Interests:** None

Copyright: © 2026 Mangaiyarkarasi & Malathi. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment ([iSee](#))

ISSN

Print: 0974-6846

Electronic: 0974-5645

Enhancing Security in Software-Defined Networking from ARP Spoofing Attack Using Adaptive Encryption-Assisted Explainable Mish-Activated Recurrent Neural Network Framework

V Mangaiyarkarasi^{1*}, S Malathi²

¹ A Veeriyar Vandayar Memorial Sri Pushpam College (Autonomous), Poondi -613503, Thanjavur, Affiliated to Bharathidasan University, Tiruchirappalli, Tamil Nadu, India

² Department of Computer Science, Swami Dayananda College of Arts & Science, Manjakkudi, Thiruvavur, Affiliated to Bharathidasan University, Tiruchirappalli, Tamil Nadu, India

Abstract

Objectives: To develop an MARNN framework for accurate ARP spoofing-based MITM attack detection and secure communication in SDN environments.

Method: The approach involves preprocessing raw data from the ARP Spoofing-Based MITM Attack Dataset using Variable Stability Scaling (Var-SS) normalization, followed by the MARNN model for accurate detection of ARP spoofing attacks. At the same time, SHAP and PDP-ICE provide interpretability to the detection process. For communication security, a Paillier-enhanced AES (P-AES) mechanism is integrated, combining Paillier homomorphic encryption with AES and dynamic key generation to ensure authenticated and leakage-free data transfer. The Python-based framework was evaluated on metrics including encryption/decryption time, throughput, accuracy, F-measure, and false discovery rate. **Findings:** The proposed approach outperforms existing techniques, achieving 98.9% accuracy, 98.89% F1-score, and 1.68 ms decryption time. **Novelty:** The framework demonstrates its effectiveness, robustness, and practicality in detecting ARP spoofing attacks and securing SDN communication; thus making it a reliable solution for mitigating MITM intrusions in SDN environments.

Keywords: Encryption; Decryption; Spoofing Attacks; Artificial Intelligence; Cryptography; SDN users; Key Generation; Recurrent Neural Network

1 Introduction

Software-defined networking (SDN) has been developed as an important networking approach through the decoupling of the control and data planes to ensure centralized control, programmability, and flexibility in traffic orchestration^{(1), (2), (3)}. The above strengths have resulted in rapid growth of SDN in areas such as cloud computing, IoT, and next-generation communication networks. Nevertheless, the centralized control

process is also vulnerable to various security risks, making SDNs vulnerable to cyberattacks^{(4), (5)}. Among all the threats, ARP spoofing is an important threat, which is created because of the lack of ARP authentication, making it easy for the attacker to change the IP-MAC association for the purpose of conducting MITM attacks^{(6), (7), (8)}. Several approaches for machine learning, deep learning, and XAI-based approaches have been proposed for securing SDN networks. Machine learning-based ARP spoofing detection has enhanced attack detection by analyzing network traffic but has been limited by dataset dependency and adaptability to changing environments^{(9), (10)}. Game theory-based and verification-based techniques provided better protection against ARP attacks, but at the cost of introducing certain theoretical assumptions and verifying the overheads⁽¹¹⁾. The deep learning models using federated learning, GNNs, and hybrid systems have offered better intrusion detection, but computational overheads and lack of interpretability pose a serious problem^{(12), (13), (14)}. Likewise, SD-IoT security systems with a multi-layered approach and ensemble learning techniques provided effective attack prevention, but their assessment was limited to certain attack types or involved significant computational overheads^{(15), (16)}. Similarly, some other SDN-based approaches for intrusion detection based on supervised learning and explainable models have produced better results, but there is still a challenge of providing accurate and real-time protection against evolving attacks^{(17), (18)}.

In addition to that, the traditional techniques of intrusion detection and encryption also suffer from several constraints, including the reliance on manually selected features, inability to detect zero-day attacks, and opaque modeling. While the technique of deep learning enhances the representation and accuracy of features, the black-box approach of decision-making decreases the level of confidence of the administrator in this security technique. Thus, using a combination of secure encryption mechanisms and explainable models of deep learning can be considered an effective strategy to develop SDN security systems. With this background, the current study proposes an Adaptive Encryption-Assisted Explainable Mish-Activated Recurrent Neural Network (MARNN) model for detection of ARP Spoofing in SDN. The framework combines encryption-based secure communication, recurrent deep learning-based attack detection, and SHAP-driven interpretability to achieve accurate, reliable, and transparent intrusion detection while addressing the limitations of existing approaches.

The major contributions of the manuscript are as follows:

- A Var-SS normalization technique to improve the quality and reliability of the dataset to provide more stable and consistent inputs for downstream modeling.
- The MARNN model is proposed to detect and classify the occurrence or non-occurrence of ARP spoofing attacks in SDN networks efficiently, by taking advantage of the Mish activation function for enhanced learning and generalization.
- Explainable AI (XAI) methods, such as SHAP and PDP-ICE, are combined for retaining interpretability of the model's decisions and presenting enhanced visual insights into intrusion detection results.
- An enhanced P-AES system is applied that aligns the Paillier cryptosystem's homomorphic behavior with AES, coupled with dynamic key generation, to improve resistance against cryptography attacks and provide end-to-end secure communication in SDN environments.

The rest of the manuscript is presented as follows: Section 2 discusses related works, Section 3 outlines the suggested methodology in detail, Section 4 represents the experimental results and their associated analysis, and Section 5 concludes the manuscript and presents possible future directions.

2 Methodology

2.1 Proposed Methodology

The proposed method presents an adaptive encryption-assisted, explainable intrusion detection model for SDN networks, as shown in Figure 1. The framework begins with preprocessing the original data samples from the ARP Spoofing-Based MITM Attack Dataset using the Var-SS normalization technique, which improves data quality and dependability. Next, a MARNN is used to precisely detect and classify ARP spoofing attacks, thereby facilitating accurate intrusion detection. To enhance interpretability and deliver more comprehensive insights into the decision-making process of the model, explainable AI methods like SHAP and PDP-ICE are incorporated, providing detailed visualizations of detection results. To provide secure communication in the SDN framework, a P-AES mechanism is used, fusing the homomorphic nature of the Paillier cryptosystem with AES and dynamic key generation. The encryption mechanism provides authentic, dependable, and leakage-free data sharing among SDN users while enhancing strong intrusion detection, thereby merging security, interpretability, and detection efficiency into a single framework.

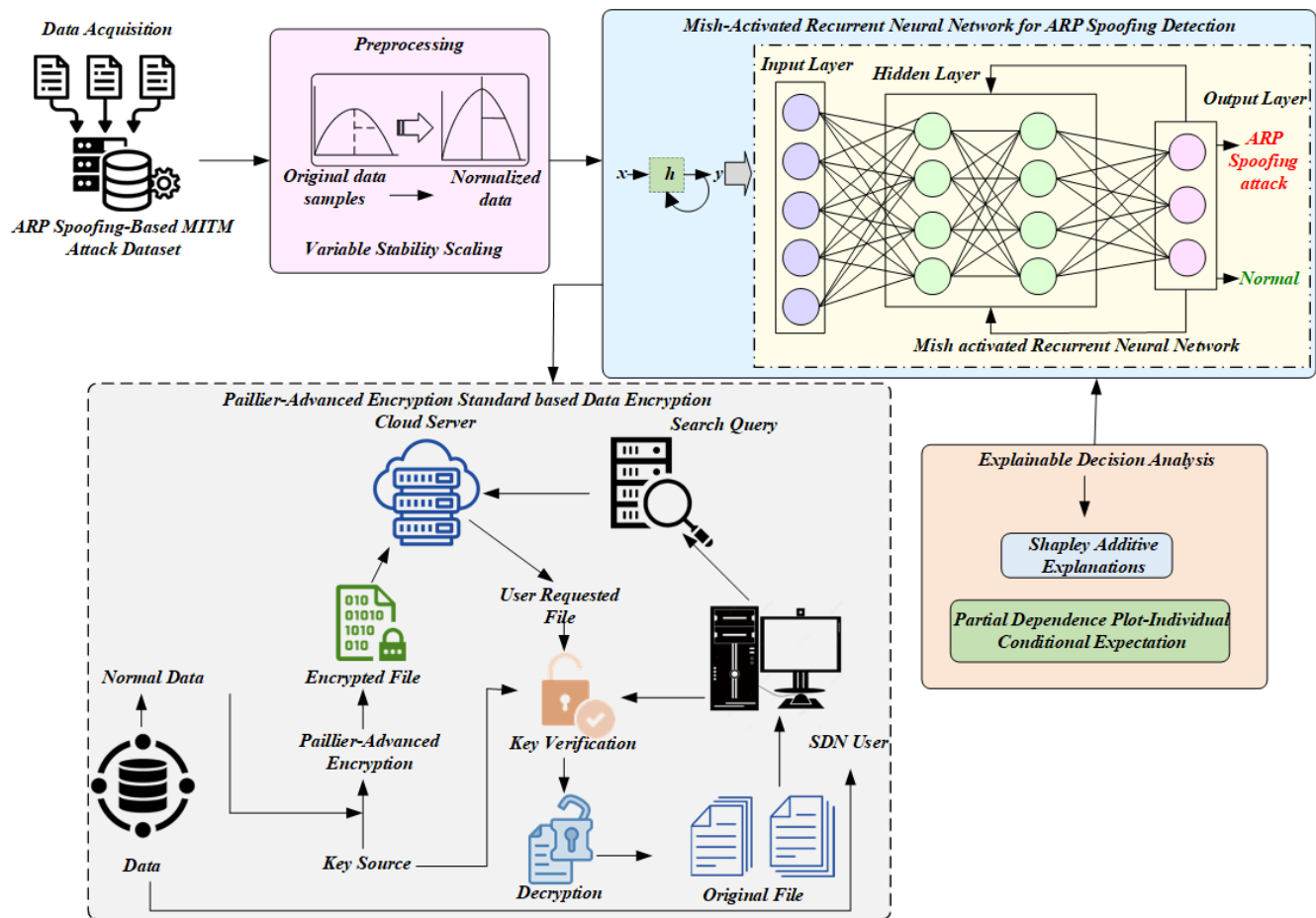


Fig 1. Block Diagram of the Proposed Framework for ARP Spoofing Detection

2.2 Data Acquisition

The raw data samples were accessed from the ARP Spoofing-Based MITM Attack Dataset⁽¹⁹⁾ hosted on Kaggle and created by mizanunswcyber. The dataset comprises several CSV files of total size around 9 MB, including informative records of network events of ARP spoofing attacks, upon which model training and testing are based. It contains spoofed and benign ARP traffic samples, giving labeled instances that mark normal and anomalous behavior. This structured format makes the dataset extremely good for training and testing deep learning-based intrusion detection models.

2.3 Preprocessing with Variable Stability Scaling

During preprocessing, the input traffic data is scaled through Var-SS to improve the reliability and quality of the data. Var-SS is an enhancement of the conventional z-score normalization, where the Coefficient of Variation (CV) is used as a scaling factor to assign more importance to features with low standard deviations while minimizing the impact of features with high variability. The normalized feature value is computed by equation (1)

$$y'_{j,k} = \frac{(y_{j,k} - \mu_j)}{\sigma_j} \cdot \frac{\mu_j}{\sigma_j} \quad (1)$$

Where $y_{j,k}$ represents the original value of the j^{th} feature, μ_j , and σ_j are the mean and standard deviation of the j^{th} feature, respectively. $\frac{\mu_j}{\sigma_j}$ is a scale factor, which enables the preprocessing to give more weight to low-variability features.

2.4 Mish-Activated Recurrent Neural Network for ARP Spoofing Detection

After preprocessing, the normalized network traffic data is fed into the MARNN, which is particularly designed to identify temporal dependencies in sequential SDN traffic through its recurrent hidden layers⁽²⁰⁾. The architecture adheres to the configuration of a simple recurrent neural network (SiRNN), namely, an input layer, an internal layer with recurrent connections, and an output layer. For an input sequence $A = \{a_0, a_1, a_2, \dots, a_D\}$, where each $a_d \in \mathcal{Y}^M$, the model generates predictions $b_d \in \mathcal{Y}^P$ for the output sequence $C = \{c_0, c_1, c_2, \dots, c_D\}$, with each $c_d \in \mathcal{Y}^P$. The hidden layer consists of T units that retain temporal information from previous time steps ($d' < d$), and the hidden and output states of the model were computed iteratively by equations (3) and (4)

$$g_d = f(r_d) \quad (3)$$

$$n_d = U_{bg}g_d + h_b \quad (4)$$

Where $r_d = U_{ga}a_d + U_{gg}g_{d-1} + h_g$, and $b_d = e(n_d)$. U_{ga} , U_{gg} and U_{bg} are weight matrices, h_g and h_b are bias vectors, f is the Mish activation, and e is a softmax function or a linear function. In ARP spoofing detection, MARNN does binary classification for identifying the occurrence of spoofed traffic, trained on cross-entropy loss on all time steps for best accuracy. It builds upon SiRNN, motivated by IRNN, with two primary changes: substituting ReLU with Mish for better gradient flow and improved representation learning. Second, the recurrent weight matrix U_{gg} is an initialized positive definite matrix with one spectral radius, and biases are initialized to zero. These enhancements stabilize the hidden dynamics and avoid gradient-related problems like vanishing or exploding gradients during backpropagation through time (BPTT). From the dynamic systems point of view, when U_{gg} is positive definite, the evolution of the hidden state is given by the equation (5)

$$g'_d = \begin{cases} \Gamma g'_{d-1} \text{ if } g'_{d-1} > 0 \\ 0 \text{ otherwise} \end{cases} \quad (5)$$

Where $\Gamma = Q^{-1}U_{gg}Q$ is a diagonal matrix, Q is a matrix composed of eigenvectors of U_{gg} and $g'_d = Q^{-1}g_d$. To maintain stability during Mish activation, U_{gg} is initialized with the equation (6)

$$U_{gg} = \frac{J + I}{x} \quad (6)$$

Where $I = \frac{1}{V} \langle W^D, W \rangle$, and $x = \max(\lambda(I + J))$. W is a standard normal matrix, $\langle \rangle$ represents the dot product, $\lambda(\cdot)$ refers to the set of all eigenvalue matrices, V is the number of hidden units in the recurrent layer, x is the largest eigenvalue, J is the identity matrix, and I is the normalized positive matrix. This initialization ensures that U_{gg} remains positive definite with controlled spectral radius, allowing the MARNN to learn SDN traffic temporal dependencies robustly without unstable dynamics. The Mish activation function, replacing the default ReLU, is defined mathematically by equation (7)

$$f(a) = a \cdot \tanh(\ln(1 + e^a)) \quad (7)$$

Where a is the input, $\tanh$ is the hyperbolic tangent function, and $\ln$ refers to the natural logarithm. Mish is non-monotonic and continuously differentiable, offering smoother gradient flow and reducing the gradient saturation issue commonly encountered with ReLU, especially for $a \leq 0$. Empirical results demonstrate that Mish is always superior to both ReLU and Swish in sequential modeling problems.

2.5 Enhancing Interpretability Using Explainable AI

To enhance interpretability, two Explainable AI (XAI) techniques are incorporated. SHAP is utilized to estimate the contribution of each input feature, i.e., packet-level features or flow statistics, thus emphasizing the most impactful indicators of spoofed traffic. Simultaneously, the PDP-ICE approach gives localized, instance-based explanations by visualizing the specific features' changes that impact the class label for benign and malicious traffic. Meanwhile, SHAP and PDP-ICE fortify the IDS's transparency and credibility, allowing for enhanced trust and comprehension of the IDS's decision-making process in identifying the presence and absence of ARP spoofing attacks.

2.5.1. Shapley Additive Explanations

Having interpretable predictive models is vital in creating a reliable IDS because most deep learning-based intrusion detection approaches tend to act like black boxes. In response to this, SHAP, which is based on game theory, quantifies the contribution of every input feature (e.g., packet size, inter-arrival time, traffic volume, or protocol utilization) to the overall classification, hence gaining feature-level transparency. In this setting, every feature is treated as a player of a cooperative game, and its Shapley value is the incremental contribution to the outcome of the prediction. The explanation model is defined by equation (8)

$$h(a') = \gamma_0 + \sum_{k=1}^M \gamma_k a'_k \quad (8)$$

Where $h(a')$ is the IDS model prediction, M is the number of input features, and γ is the feature attribution score. Feature attribution is calculated by equation (9)

$$\gamma_k = \sum_{J \subseteq N \setminus \{k\}} \frac{|J|!(M - |J| - 1)!}{M!} [I_b(J \cup \{k\}) - I_b(J)] \quad (9)$$

Where J is the subset of the feature set N , and $I_b(J)$ is the expected model prediction based solely on the subset J . The expected value is given by equation (10)

$$I_b(J) = D[I(b) | b_J] \quad (10)$$

Where $D[I(b) | b_J]$ is the expected prediction from the subset J . By using SHAP in the advocated IDS, the specific contributions of network traffic features to labeling data as benign or spoofed (ARP attack) are made transparent.

2.5.2. Partial Dependence Plot-Individual Conditional Expectation

PDPs were utilized in the designed IDS as a model-agnostic visual tool to identify the average impact of chosen input features (e.g., packet size, traffic interval, ARP request frequency) on model predictions. Let $A \subset \{1, \dots, q\}$, the features of interest (e.g., ARP-related traffic fields), and B the complement set of remaining features is the two subsets of the predictor. The partial dependence function of the IDS model f on the feature subset y_A is given by equation (11)

$$f_A = \mathbb{E}_{y_B} [f(y_A, y_B)] = \int f(y_A, y_B) dQ(y_B) \quad (11)$$

Since the true data distribution $Q(y_B)$ is unknown, it is estimated by averaging over the observed training samples and is shown in equation (12)

$$\hat{f}_A = \frac{1}{M} \sum_{j=1}^M \hat{f}(y_A, y_{Bj}) \quad (12)$$

Where y_{Bj} are the target values of the complementary features in the training set. For each subset of predictors A has its own partial dependence function f_A . PDPs reveal the global average effects of features on model predictions but could hide heterogeneous effects whenever there are strong interactions between features. To overcome this, ICE plots are used. ICE plots break down the PDP by showing instance-level conditional relationships and thus how predictions differ for individual traffic samples and providing a finer view of the model's behavior. For each observation $\{(y_{Aj}, y_{Bj})\}_{j=1}^M$ the ICE curve is as shown in equation (13)

$$f_A^{(j)}(y_A) = f(y_A, y_{Bj}) \quad (13)$$

Where y_{Bj} is fixed at the j^{th} observation, and y_A is varied. Plotting $f_A^{(j)}(y_A)$ for each j provides M conditional expectation curves. The PDP is then merely the average of all ICE curves and is shown in equation (14)

$$\hat{f}_A(y_A) = \frac{1}{M} \sum_{j=1}^M f_A^{(j)}(y_A) \quad (14)$$

In ARP spoofing detection, ICE plots enable us to see how particular network sessions react to variations in traffic features, unveiling conditional heterogeneity that PDP alone could miss. When ICE curves have large intercept shifts, it can be

challenging to identify interaction effects. To enhance interpretability, Centered ICE (c-ICE) plots are used. For a selected reference point y^* in the range of y_A , the centered ICE transformation of each ICE curve is given by the equation (15)

$$\hat{f}_{center}^{(j)} = \hat{f}^{(j)} - 1\hat{f}(y^*, y_{Bj}) \quad (15)$$

Where $\hat{f}_{center}^{(j)}$ is the centered ICE curve $(y^*, \hat{f}(y^*, y_{Bj}))$ acts as a base case for each curve. In IDS, centered ICE plots enable us to visualize more effectively the relative impact of ARP traffic features between sessions, emphasizing cumulative effects and dependencies. To investigate interaction effects further, derivative ICE plots are used. Rather than plotting function values, derivative ICE deals with the partial derivative of $\hat{f}$ with respect to y_A . The $\hat{f}$ can be given by the equation (16)

$$\hat{f}(y) = \hat{f}(y_A, y_B) = k(y_A) + l(y_B) \quad (16)$$

So, the derivative of $\hat{f}$ can be given by the equation (17)

$$\frac{\partial \hat{f}(y)}{\partial y_A} = k'(y_A) \quad (17)$$

If no interaction occurs, all ICE curves have a common derivative and thus the same shape. With interactions, the derivatives are different between observations, yielding heterogeneous patterns. Through the application of PDP-ICE in the recommended IDS, the localized impacts of each network traffic characteristic on the classification are visualized, enabling the model to present how predictions change over various instances. PDP-ICE separates the overall feature impacts and captures instance-specific variation in labeling traffic as benign or ARP spoofed, thus enhancing interpretability by having more tangible conditional associations and enhancing the IDS's transparency, trustworthiness, and reliability in real-time identification. Subsequently, traffic features like MAC/IP addresses, ARP packets, timestamps, and node identity are securely encrypted using Paillier-based AES encryption.

2.6 Paillier-boosted Advanced Encryption Standard

ARP spoofing traffic is initially intercepted, and the MITM attack is prevented by the process of MARNN, after which only legitimate traffic information is protected. The usual traffic characteristics like source and destination MAC/IP addresses, ARP request/response packets, timestamps, and node identity are encrypted by the Paillier-based AES (P-AES) technique^(?). This method encrypts sensitive traffic data from plaintext to ciphertext so that attackers are not able to inject spoofed packets back into the network. The authorized parties with the proper private key alone can decrypt the secured data, thus preserving confidentiality and integrity. In addition, the additive homomorphism nature of the Paillier scheme facilitates secure aggregation of encrypted traffic features without reducing accuracy when decrypted. Hence, the three stages of P-AES: key pair generation, encryption, and decryption, as a whole, maintain normal traffic security after ARP spoofing prevention.

2.6.1. Key Generation Process

The process generates cryptographic keys to use in encrypting and decrypting ARP traffic features.

Step 1: Choose two large primes s and t independently, as shown in equation (18)

$$\gcd = (st, (s-1)(t-1)) = 1 \quad (18)$$

Where s and t are two large random prime numbers, gcd is the greatest common divisor of integers.

Step 2: Calculate $m = s \times t$ and get lcm which is shown in equation (19) to obtain γ

$$\gamma = lcm = ((s-1)(t-1)) \quad (19)$$

Where m is a part of the public key, lcm is the least common multiple of integers, γ is a private exponent used in decryption.

Step 3: Generate a random integer $u \in X m^2$

Where u is an element of the multiplicative group modulo m^2 .

Step 4: Ensure the order of u is a multiple of m in $X * m^2$

Step 5: Public key is (m, u)

Step 6: Private key is (γ, ν)

Where ν is used in decryption to recover the plaintext. These generated keys are the basis of the P-AES encryption process so that only authorized entities can decrypt and encrypt ARP traffic attributes. Through securely sharing the public key and safeguarding the private key, the system assures confidentiality and integrity of the network communications.

2.6.2. Encryption Process

P-AES is used to encrypt network traffic features by encoding them before processing. In this process, a corresponding public and private key pair is created in the encryption process, such that only authorized parties with the private key can obtain the original information. After being encrypted, the data traffic that is protected is then sent securely to the IDS or cloud server, thus making it impossible for attackers to inject or manipulate ARP-related packets during intrusion detection.

Step 1: Let p be the ARP traffic feature such that $0 \leq p < m$

Step 2: Select a random integer v such that $0 < v < m$, with $v \in X * m^2$

Step 3: Calculate the ciphertext as shown in equation (20)

$$c = up * v \bmod m^2 \quad (20)$$

Where p is the plaintext message, v is the random integer chosen between $(0, m)$, u is the public key generator, m is the part of the public key, and c is the ciphertext.

This ciphertext is then securely transmitted, shielding attackers from inferring or tampering with the original ARP traffic characteristics. Consequently, the encrypted data is protected during communication and storage within the IDS system.

2.6.3. Decryption Process

After the encrypted ARP traffic is received, only the legal detector with the private key can extract the original values for spoofing analysis.

Step 1: Let c be the ciphertext corresponding to the ARP traffic feature, where $c \in X * m^2$

Step 2: Recover the plaintext ARP feature, which is shown in equation (21)

$$plaintext = L(c^\gamma \bmod m^2) \quad (21)$$

Where c is the ciphertext, γ is the private key parameter, and $\bmod m^2$ is the modulo m^2 . With the use of the P-AES encryption method, only legitimate detectors can restore original ARP traffic characteristics for maintaining integrity and confidentiality in spoofing detection. Legitimate traffic is encrypted securely once attacks have been thwarted, thus avoiding any unauthorized access or tampering. With the use of Paillier's homomorphic qualities and AES robustness, the IDS gets strong confidentiality, integrity, and secure communication, thus making it more reliable and safeguarded against malicious behavior.

3 Results and Discussion

This section summarizes the experimental results of the proposed Adaptive Encryption-Assisted Explainable Mish-Activated Recurrent Neural Network for precise detection and safe communication. The model is implemented in Python and evaluated on a system equipped with an Intel Core i5-5200U CPU @2.20GHz, 4 GB RAM, and a 64-bit operating system. The dataset is split in an 8:1:1 ratio for training, testing, and validation, respectively. Comparative analysis is conducted against existing techniques, including TS-RBDM⁽⁶⁾, LSTM-CNN⁽⁷⁾, BSHNN-LEA⁽⁸⁾, and Gbell PRF-MAC-CGRU⁽²¹⁾.

3.1 Confusion Matrix

Figure 2 shows the confusion matrix, which points to the classification accuracy of the proposed method against ARP spoofing attacks and normal cases.

Figure 2 shows the confusion matrix analysis of the proposed method. It demonstrates how accurately the model separated "Normal" and "Attack" classes. Among 6,852 real normal samples, the model accurately predicted 6,775 as normal and mispredicted 77 as attacks. In the same way, among 8,017 attack samples, 7,930 were accurately predicted as attacks, but just 87 were mispredicted as normal. These findings emphasize that the model is highly accurate in classification and produces very few false negatives and false positives, thus being extremely reliable for attack detection while minimizing misclassifications of normality.

3.2 Simulation analysis of the Proposed Method over the Existing Method

In this section, the performance achieved by the proposed method over the existing schemes is deliberated via graphical illustrations. Several existing methods, like TS-RBDM, LSTM-CNN, BSHNN-LEA, and Gbell PRF-MAC-CGRU techniques, were compared with the proposed framework. The comprehensive scrutiny of the attained effectiveness is depicted below.

Figure 3 shows the model's training and validation performance over 100 epochs. The training accuracy sharply increases within the initial 10 epochs and remains steady at nearly 99%, while the validation accuracy trends similarly, slightly fluctuating

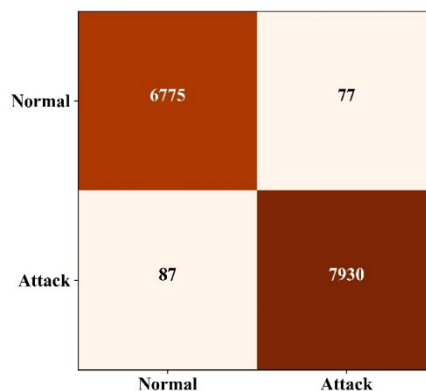


Fig 2. Confusion matrix analysis of the proposed method

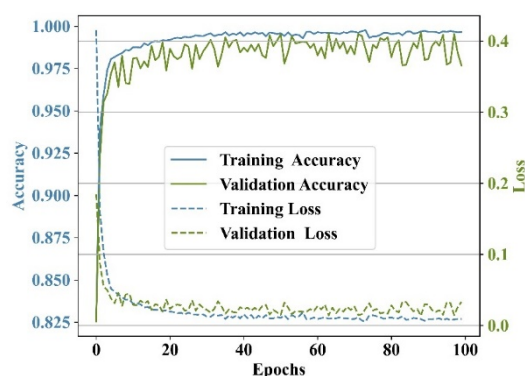


Fig 3. Training and validation analysis of accuracy and loss

but continuously ranging around 98–99%. In contrast, the training and validation loss decline very sharply in the first few epochs and reduce to very low values after around 15 epochs, and stabilize for the remainder of the training. This is an indicator that the model is realizing fast convergence, high accuracy, and robust generalization over the entire 100 epochs without notable evidence of overfitting.

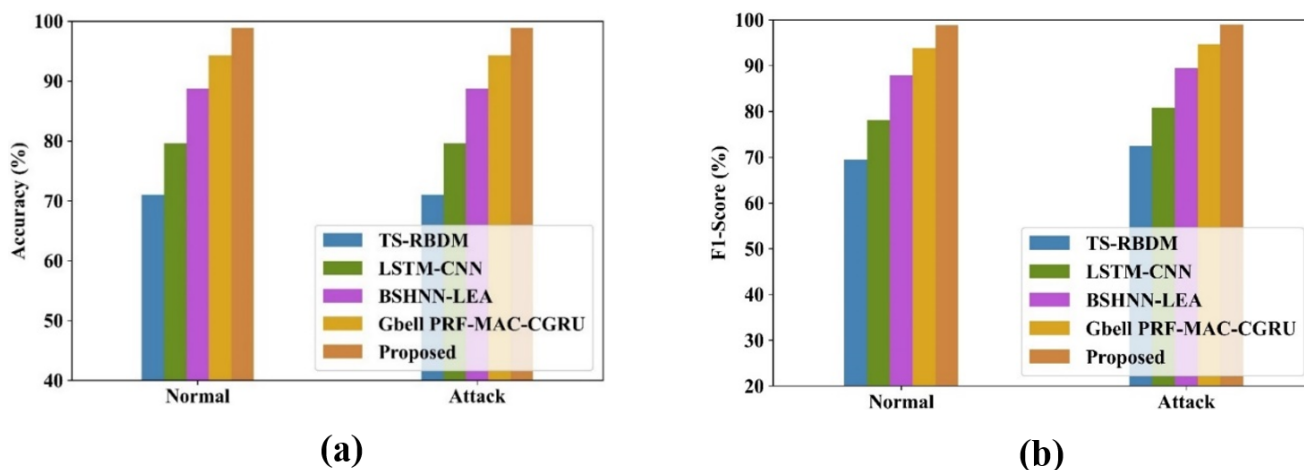


Fig 4. Comparative analysis of (a) Accuracy and (b) F-score

Figure 4 shows the comparative analysis of accuracy. The performance comparison indicates that the TS-RBDM technique achieved 71.07% accuracy for both normal and attack detection. The LSTM-CNN model has an enhanced accuracy of 79.60%, while the BSHNN-LEA technique improved it to 88.71%. The Gbell PRF-MAC-CGRU method registered an even higher accuracy of 94.32%, with remarkable improvement compared to the previous methods. Lastly, the proposed approach surpassed all existing methods, with a remarkable 98.89% accuracy for normal and attack detection. The outstanding performance proves that the proposed approach has better feature representation, greater precision, and stronger generalization; hence, it is more reliable and effective than the previous methods. Figure 7 shows the comparative analysis of the F1-score. For the Normal class, the F1-score for TS-RBDM was 69.47%, which rose to 78.15% for LSTM-CNN, further to 87.86% for BSHNN-LEA, and stood at 93.87% for Gbell PRF-MAC-CGRU, whereas the proposed method surpassed them all with 98.80%. For the Attack class, TS-RBDM scored an F1-score of 72.51%, LSTM-CNN scored 80.88%, BSHNN-LEA scored 89.45%, and Gbell PRF-MAC-CGRU scored 94.71%, while the proposed method scored the highest with 98.97%.

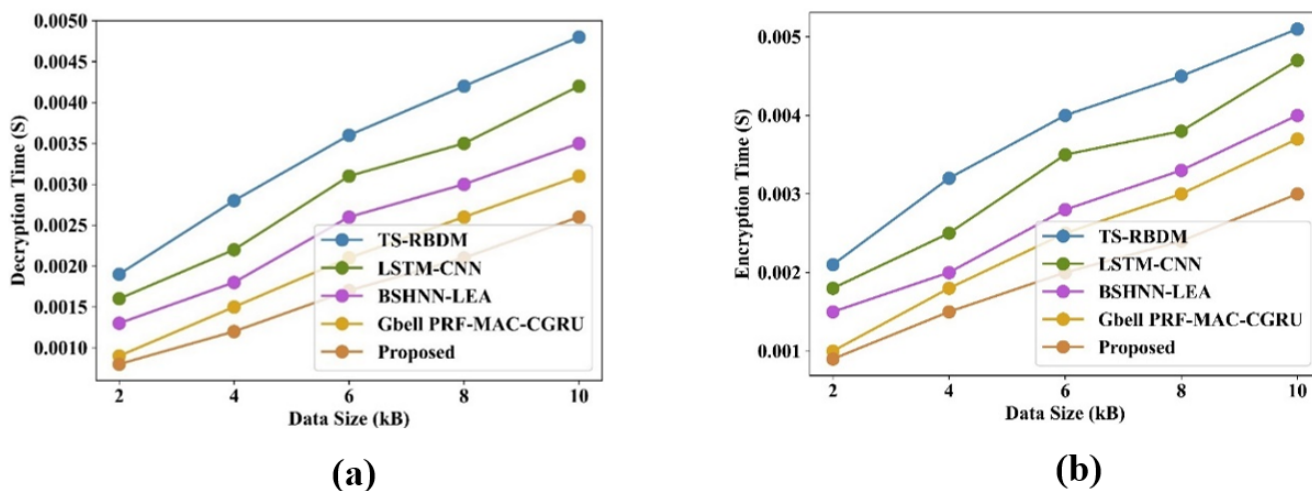


Fig 5. Comparative analysis of (a) decryption time and (b) encryption time

Figure 5(a) compares the decryption time of different methods for data sizes ranging from 2 kB to 10 kB. As the data size increased, the decryption time increased for all methods. However, the proposed approach consistently achieved the lowest decryption time, ranging from 0.0008 s (2 kB) to 0.0026 s (10 kB), outperforming TS-RBDM, LSTM-CNN, BSHNN-LEA, and Gbell PRF-MAC-CGRU. Similarly, Figure 5(b) presents the encryption time comparison. Although encryption time increased with data size, the proposed method consistently required the least time, varying from 0.0009 s (2 kB) to 0.0030 s (10 kB), demonstrating superior computational efficiency over the competing approaches.

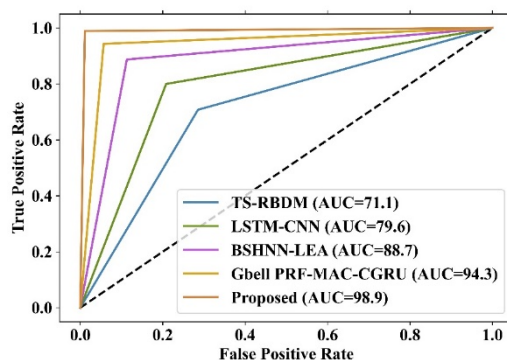


Fig 6. Comparative analysis of AUC-ROC

Figure 6 presents a comparative analysis of the AUC-ROC. The ROC curve is used for comparing models' performance based on how well they classify between classes. The TS-RBDM model has an AUC of 71.1, which is a comparatively poor

classification performance. LSTM-CNN does better with an AUC value of 79.6. The BSHNN-LEA model does even better with an AUC of 88.7, clearly indicating a better balance between sensitivity and specificity. The Gbell PRF-MAC-CGRU has an even better AUC value of 94.3, indicating strong predictive power.

3.3 Explainable AI Analysis

In this section, the obtained explanations are illustrated for the proposed model using Shapley additive explanations (SHAP) and Partial Dependence Plot–Individual Conditional Expectation (PDP-ICE) model-agnostic plots, based on the ARP Spoofing-Based MITM Attack Dataset. A brief analysis of the obtained XAI plots is deliberated below:

3.3.1. SHAP Analysis

SHAP analysis showcases the relative significance of characteristics like ARP request rate, MAC address modification, switch port activity, and network traffic patterns in shaping the model's predictions. This allows for open interpretation of the detection decisions, solidifying the trustworthiness and resilience of the SDN's defense against ARP spoofing attacks.

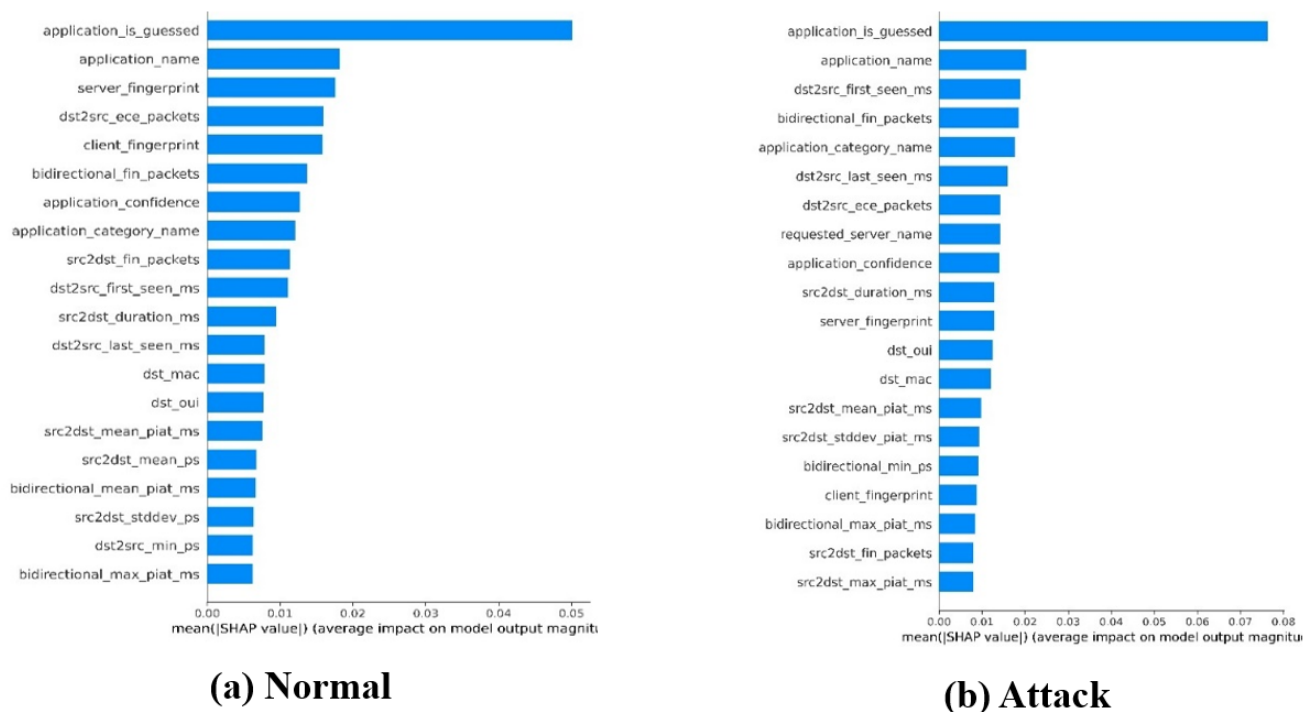


Fig 7. SHAP Feature Importance Analysis for ARP Spoofing-Based MITM Attack Dataset

Figure 7 shows the SHAP feature importance plot, which depicts unique patterns in normal and attack traffic. For the normal scenario, `application_is_guessed` is the most impactful feature, followed by `application_name`, `server_fingerprint`, `dst2src_ece_packets`, and `client_fingerprint`, indicating that stable application-level attributes and persistent packet exchanges are the primary factors in identifying normal flows. Conversely, `application_is_guessed` plays a significantly greater role during attacks, capturing uncertainty in application signatures typical of malicious traffic, while `application_name`, `dst2src_first_seen_ms`, `bidirectional_fin_packets`, and `application_category_name` also take on greater significance. In addition, `duration_ms`, `piat_ms`, and `stddev_piat_ms`, as well as identifiers `dst_mac` and `dst_oui`, place greater weight in the attack case, since attacks tend to tamper with session timing, packet rates, and device identifiers.

3.3.2. PDP-ICE Analysis

PDP-ICE analysis illustrates the impact of changes in ARP request rate, MAC address modification, switch port activity, and traffic patterns on the predictions made by the model. This global and instance-level duality in behavior improves interpretability and enhances the SDN's resilience to ARP spoofing attacks.

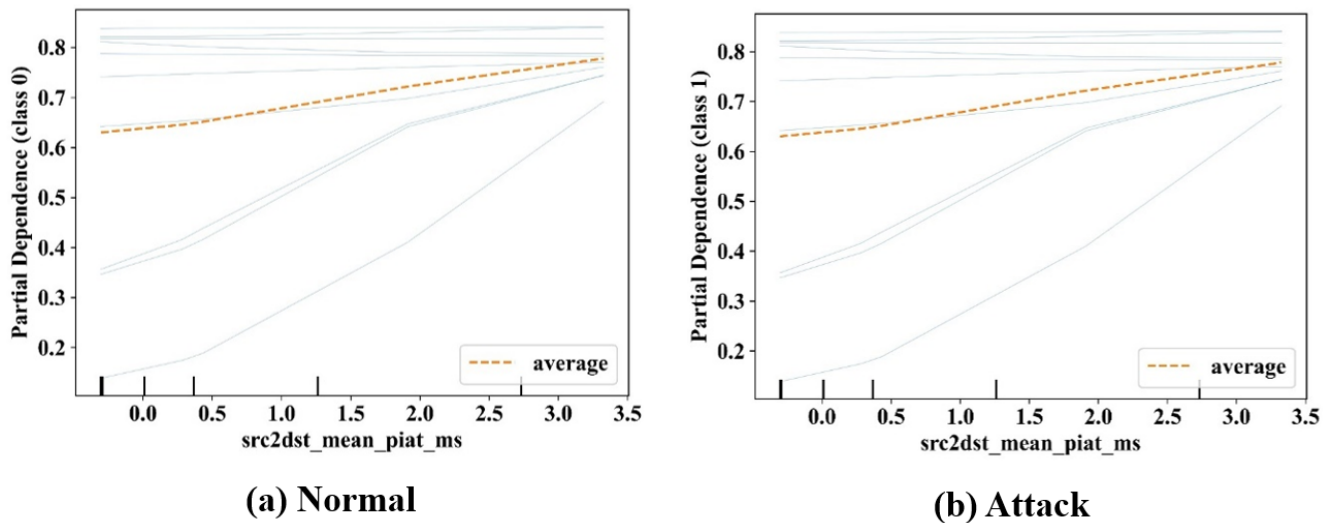


Fig 8. PDP-ICE Feature Importance Analysis for ARP Spoofing-Based MITM Attack Dataset

Figure 8 shows the PDP-ICE of `src2dst_mean_piat_ms`, showing its effect on model predictions for both normal traffic and attack traffic. In the usual case (a), as the mean inter-arrival time of packets between source and destination grows larger, the chance of being labeled as normal traffic (class 0) continually increases, with the average curve having a general upward tendency, which suggests that solid and slightly increased inter-arrival times are tightly coupled with normal communication behaviors. In case (b) of an attack, the same rising pattern is found, but the reliance is marginally in the direction of class 1 (attack), indicating that, as well as larger inter-arrival times affecting malicious traffic, the model discriminates based on the fine differences and irregularities in timing patterns. In general, the plots affirm that `src2dst_mean_piat_ms` is a fundamental temporal feature in which smoother variations coincide with normal flows and anomalies, or atypical timing behaviors give rise to attack classification, thus improving the model's interpretability in differentiating benign and malicious traffic.

4 Conclusion

The proposed Adaptive Encryption-Assisted Explainable Mish-Activated Recurrent Neural Network (MARNN) framework effectively integrates reliable intrusion detection with safe communication in SDN networks. The reliability of data is enhanced by Var-SS normalization, and reliable ARP spoofing-based attack detection is offered by the MARNN model supported by SHAP and PDP-ICE explanations for a clear and informative understanding of results. The communications security in the proposed framework is enhanced through an AES mechanism fortified with Paillier, providing support for homomorphic encryption as well as dynamic key generation to provide immunity against cryptographic attacks. Results from experiments prove the efficacy of the framework, which gains a 19.91% improvement in accuracy and a 90.25% reduction in the rate of false discovery when compared with other methods, proving its better intrusion detection ability. In the future, the architecture could be scaled to enable collective intrusion detection based on federated learning across distributed SDN domains so that controllers in different geographical regions can learn from one another without exchanging raw data. Scalability would be enhanced while offering privacy-preserving global threat intelligence against DDoS and routing attacks at scale. In addition, implementing digital twin technology in SDN environments could also facilitate real-time simulation, prediction, and anticipation of prospective attacking scenarios, further improving adaptability and resilience in large-scale network deployments.

5 Acknowledgements

NIL

Author details: V. Mangaiyarkarasi is a Research Scholar; Dr. S. Malathi is the Research Supervisor & serving Assistant Professor.

References

- 1) Majumder S, Barma MKD, Saha A. ARP spoofing detection using machine learning classifiers: an experimental study. *Knowledge and Information Systems*. 2025;67(1):727–766. <https://doi.org/10.1007/s10115-024-02219-y>.
- 2) Nivedhidha M, RamKumar MP, Dharani M, Selvan GSRE. XGBoost meets Explainable AI: A Scalable Approach for Detection of ARP Spoofing. *Zenodo*. 2025. <https://doi.org/10.5281/zenodo.15322871>.
- 3) Malathi S, Begum SR. Enhancing trustworthiness among IoT network nodes with ensemble deep learning-based cyber-attack detection. *Expert Systems with Applications*. 2024;255:124528. <https://doi.org/10.1016/j.eswa.2024.124528>.
- 4) Rahman MA, Hossain MS. A deep learning-assisted software-defined security architecture for 6G wireless networks: IIoT perspective. *IEEE Wireless Communications*. 2022;29(2):52–59. <https://doi.org/10.1109/MWC.001.2100426>.
- 5) Jain P, Rathour A, Sharma A, Chhabra GSC. Bridging Explainability and Security: An XAI-Enhanced Hybrid Deep Learning Framework for IoT Device Identification and Attack Detection. *IEEE Access*. 2025. <https://doi.org/10.1109/ACCESS.2025.3562725>.
- 6) Dandotiya M, Makwana RRS. Secured DDoS attack detection in SDN using TS-RBDM with MDPP-Streebog-based user authentication. *Transactions on Emerging Telecommunications Technologies*. 2025;36(2):e70052. <https://doi.org/10.1002/ett.70052>.
- 7) Prasad P, Tahir M, Isoaho J. Enhancing explainability of artificial intelligence for threat detection in SDN-based multicast systems. *Procedia Computer Science*. 2025;257:569–574. <https://doi.org/10.1016/j.procs.2025.06.070>.
- 8) Nayak NKS, Bhattacharyya B. An intrusion detection system for 5G SDN network utilizing binarized deep spiking capsule Fire Hawk neural networks and blockchain technology. *Future Internet*. 2024;16(10):359. <https://doi.org/10.3390/fi16100359>.
- 9) Hnamte, Vanlalruata, Hussain J. Enhancing security in software-defined networks: An approach to efficient ARP spoofing attacks detection and mitigation. *Telematics and Informatics Reports*. 2024;14:100129. <https://doi.org/10.1016/j.teler.2024.100129>.
- 10) Majumder S, Barma MKD, Saha A. ARP spoofing detection using machine learning classifiers: an experimental study. *Knowledge and Information Systems*. 2025;67(1):727–766. <https://doi.org/10.1007/s10115-024-02219-y>.
- 11) Mvuh F, Tchendji VK, Djamegni CT, Anwar AH, Tosh DK, Kamhoua C. Countering ARP spoofing attacks in software-defined networks using a game-theoretic approach. *Computers & Security*. 2024;139:103696. <https://doi.org/10.1016/j.cose.2024.103696>.
- 12) Rohith S, Logeswari G, Tamilarasi K, Sudhakaran G. A federated deep learning approach for SDN security with quantum-optimized feature selection and hybrid MSDC net architecture. *Scientific Reports*. 2026;16(1):8038. <https://doi.org/10.1038/s41598-026-37289-1>.
- 13) Zhao J, Zhang D, He Q, Lin M, Yang Y. DMSTG-AD: an SDN intrusion detection method based on dynamic multi-scale spatio-temporal graph neural network. *Scientific Reports*. 2026;16:14528. <https://doi.org/10.1038/s41598-026-44360-4>.
- 14) Elshewey AM, Abbas S, Osman AM, Aldakheel EA, Fouad Y. DDoS classification of network traffic in software-defined networking using a hybrid convolutional and gated recurrent neural network. *Scientific Reports*. 2025;15(1):29122. <https://doi.org/10.1038/s41598-025-13754-1>.
- 15) El-Sayed A, Ramadan H, Mohamed ER, Elkomy OM. SFARP: a multi-layered real-time security framework for hybrid ARP and DDoS attack defense in SD-IoT networks. *Scientific Reports*. 2025;15:43479. <https://doi.org/10.1038/s41598-025-28830-9>.
- 16) Akpokodje E, Tang T, Rajapaksha P. Enhancing UAVs security: SDN and ML-based automated ARP spoofing detection and defense. *Proceedings of SPIE - Autonomous Systems for Security and Defence II*. 2025;13680. <https://doi.org/10.1117/12.3071823>.
- 17) Wang S, Balarezo JF, Chavez KG, Al-Hourani A, Kandeepan S, Asghar MR, et al. Detecting flooding DDoS attacks in software-defined networks using supervised learning techniques. *Engineering Science and Technology, an International Journal*. 2022;35:101176. <https://doi.org/10.1016/j.jestech.2022.101176>.
- 18) Michael I, Oluwasogo J, Isreal O. Adaptive Ensemble Learning for Zero-Day Attack Detection in Software-Defined Vehicular Networks. 2025. Available from: https://www.researchgate.net/publication/392078899_Adaptive_Ensemble_Learning_for_Zero-Day_Attack_Detection_in_Software-Defined_Vehicular_Networks.
- 19) Kaggle. ARP Spoofing based MITM Attack Dataset. . Available from: <https://www.kaggle.com/datasets/mizanunswcyber/arp-spoofing-based-mitm-attack-dataset>.
- 20) Abbas S, Alsubai S, Ojo S, Sampedro GA, Almadhor A, Hejaili AA, et al. An efficient deep recurrent neural network for detection of cyberattacks in realistic IoT environment. *The Journal of Supercomputing*. 2024;80(10):13557–13575. <https://doi.org/10.1007/s11227-024-05988-7>.
- 21) Grandhi SH, Basani DKR, Gudivaka RK, Gudivaka BR, Gudivaka RL, Adewole KS. Secure SDN-IoT framework with adaptive Gbell PRF-MAC and convolutional GRU for intrusion detection systems. *IETE Journal of Research*. 2025;p. 1–15. <https://doi.org/10.1080/03772063.2025.2474206>.