Anadolu Psikiyatri Dergisi || ISSN: 1302-6631 || Volume 26; Number 3
https://doi.org/10.5281/zenodo.15322871

XGBoost meets Explainable AI: A Scalable Approach for Detection of ARP Spoofing

Nivedhidha M, RamKumar M P, Dharani M, Emil Selvan G S R

[*]Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai

***Abstract*— ARP is a critical component of network communication, yet its lack of inherent security measures makes it vulnerable to ARP spoofing attacks. These attacks exploit ARP's trust-based design, enabling adversaries to intercept, modify, or block network traffic, leading to severe consequences such as data exfiltration, denial of service, and MITM attacks. Traditional ARP spoofing detection methods often fall short due to their reliance on static rules and limited adaptability to dynamic network environments. ARP XGBoost Shield model presents an advanced methodology for detecting ARP spoofing attacks using a combination of ML and explainable AI techniques. The method combines PCA for dimensionality reduction, SHAP for feature interpretability, and the XGBoost classifier for robust anomaly detection. The system achieves high accuracy while minimizing false positive by balancing the dataset through preprocessing steps such as feature extraction, normalization, and oversampling. The proposed model is tested on a large IoT network intrusion dataset, has testing accuracy of 93%.**

Keywords—ARP Spoofing Detection, Dimensionality Reduction, Explainable AI, SHAP Values, XGBoost Classifier, Machine Learning.

1. INTRODUCTION

ARP is essential for mapping IP addresses to MAC addresses within a LAN, facilitating seamless communication between devices. However, ARP lacks security features, making it vulnerable to ARP Spoofing attacks, where an attacker manipulates ARP messages to intercept, modify, or block network traffic. Since ARP does not verify the authenticity of responses, attackers can exploit this weakness to disrupt network communications, posing a significant cybersecurity risk.

Traditional ARP spoofing detection methods, including heuristic-based approaches, ARP packet inspection, and static IP-MAC binding, have several limitations. Static methods are inflexible and prone to false positive due to valid network changes, while packet inspection struggles to scale in high-traffic environments. Modern detection techniques are needed for Software-Defined Networking.

ML and DL provides high precision, scalable and adaptive solutions for ARP spoofing detection. The models are capable of analyzing big data, identifying attack patterns and adapting to changing threats. DL methods help in improving real time attack detection with help of their hierarchical feature extraction process where as ML models helps in detecting accuracy and response times. Intelligent and automated threat detection in networks acquires greater use of AI based cyber security tools.

The proposed methodology, ARP XGBoost Shield uses a hybrid model by implying XGBoost for classification enhanced by preprocessing methods like PCA for reducing dimension and SHAP based feature identification for determining the salient attributes. ARP XGBoost model is scalable as it defects precision and minimizes false positive. With an improvement of SHAP analysis effective, efficient and scalable ARP spoofing detection can be done in such a way that it ensures network security.

2. RELATED WORKS

Husain Abdulla and others [1] (2020) proposed a method based on neural network for detection of ARP spoofing in IoT environments. The network traffic data such as TCP, UDP, and ARP packet percentages are preprocessed and hidden layer of three neurons is used along with neural network. The system is trained in such a way that it identifies network traffic anomalies. The use of neural network helped in detecting ARP spoofing with above 90% accuracy. This is comparatively more than the classic ARIMA statistical techniques. Finally, it is stated that the use of neural network in IoT environments showed enormous improvement than usual methods.

Mrinal Kumar et al. [2] used some of the popular ML algorithms for detecting and preventing ARP spoofing attacks. In this approach, ML models like RF, LSTM, CNN, SVM and Isolation Forest are combined with real time data analysis. From this it was noted that RF performed the best with 94% of accuracy, 92% of precision, and 95% of recall. Also it had low False positive rate (5%) and False negative rate (2%). The usage of CNN and LSTM were effective in which CNN performed better in spatial feature analysis and LSTM in temporal dependencies, while they used more computational power. RF was more accurate than SVM. Isolation Forest was efficient but had a low accuracy of 85% and high error rates.

Dharani et al. [3] (2024) suggested 3 ML models such as LSTM, BiLSTM, CNN for detecting ARP spoofing with the help of static thresholding. For balancing the dataset SMOTE was used. Use of ML models resulted good accuracy of more than 90% along with low false positive rate.

Yuwei Sun and colleagues (2021) [4] suggested unsupervised learning methods for identifying suspicious ARP activity. An autoencoder neural network was used for analysing ARP traffic and transforming it into latent feature vectors. These vectors were combined with k-means for identifying unique suspicious behaviour.

Ramkumar M.P et al. (2022) [5] designed an IDS especially for fog computing environments. These are vulnerable to cyberattacks because of their scattered nature. Ensemble classification model is created that is used for combining various ML algorithms and also optimization strategies were included for improving accuracy and efficiency. The developed system achieved high detection accuracy and minimized false positive rate. The specified technique showed effective real-time intrusion detection while using benchmark datasets.

Mehak Usmani and others (2022) [6] proposed ML-based model combining LSTM and DT classifiers for prediction of ARP spoofing attacks. Kitsune Network Attack Dataset was used for training the model. The models delivered 99% accuracy using LSTM network and 100% accuracy using DT classifier. From the above result it is clear that DT performed better than LSTM and proves to be efficient.

Prasana and colleagues [7] proposed IDS based on anomaly that are designed specially for ICS. This approach detected and prevented cyberattacks in industrial settings.

Lirim Ashiku et al. (2021) [8] proposed a model to detect network attacks. This is deep learning-based NIDS using CNN and MLP architectures. UNSW-NB15 dataset was used and it achieved 94.4% and 95.6% accuracy. This did not work better in some of the underrepresented attack classes.

Yanfang Fu and others [9] suggested hybrid approach combining XGBoost with RNN. The model initially started with preprocessing followed by dimensionality reduction done by XGBoost. Model training was done using NSL-KDD and UNSW-NB15 dataset. This hybrid model delivered 88.13% accuracy on NSL-KDD and 87.07% on UNSW-NB15 for binary classification.

Sydney Kasongo (2023) [10] developed an IDS using XGBoost-based feature selection and RNNs to enhance network security. The study applied XGBoost to select optimal features from NSL-KDD dataset (22 features) and UNSW-NB15 dataset (17 features), and implemented three RNN variants—LSTM, GRU, and Simple RNN for binary and multiclass classification. The combination of XGBoost and LSTM yielded notable performance for binary classification on NSL-KDD, with test and validation accuracy of 88.13% and 99.49%, respectively. XGBoost paired with Simple RNN achieved 87.07% test accuracy on UNSW-NB15. For multiclass classification, the XGBoost-LSTM and XGBoost-GRU models achieved accuracies of 86.93% and 78.40% on NSL-KDD and UNSW-NB15, respectively. While demonstrating superior performance, the system faced challenges with training time and detecting underrepresented classes, highlighting areas for future improvement.

Prasanna and others (2023) [11] present a method for detecting anomalies in IIoT protocols, which improves security in smart manufacturing and industrial networks.

Fatima Ezzahra Laghrissi [12] et al. created an IDS that uses LSTM networks to improve performance. The methodology begins with preprocessing the KDD99 dataset, which uses dimensionality reduction techniques such as PCA and MI. Three models were developed: LSTM without feature reduction, LSTM

with PCA, and LSTM with MI. The LSTM-PCA model was most effective at reducing irrelevant features and noise, achieving a high accuracy 99.49% for binary classification and 99.39% for multiclass classification. Despite its superiority in binary classification, the model's performance in multiclass classification was suboptimal due to the complexity of attack types.

Chao Liu 2021 [13] introduced a hybrid method that combines feature selection method like RF and LSTM for intrusion detection. Random Forest was used for extracting the most important features from the dataset, reducing dimensionality and improving the learning process. Once the best features have been extracted, LSTM model is used for capturing sequential dependencies in network traffic, as it is well-suited for time-series data such as network flows. The system was tested on NSL-KDD and achieved accuracy of 91.6%. However, LSTM has a long training time, particularly when combined with the feature selection process.

G. Logeswari et al. (2022) [14] proposed a Hybrid Feature Selection (HFS) methodology using CFS to remove irrelevant features and RF-RFE to refine the selection. NSL-KDD was used and achieved 98.72% accuracy. Classification was done with the help of LightGBM. The main disadvantage was increase in computational overhead.

Sams Aafiya Banu and colleagues [15] suggested model that used SMOTE for addressing imbalanced datasets in intrusion detection. This helped in improving performance of ML models.

Soosan Naderi Mighan [16] et al. (2020) suggested autoencoders and SVM for intrusion detection approach. To reduce dimensionality autoencoder was used which helped in anomaly detection. SVM was used for classifying compressed data. This combination was tested on CIC-IDS 2017 dataset and achieved 94.85% accuracy. Dependency of autoencoder on SVM limits the ability to identify non-linear attacks.

Mahalakshmi, M., and others [17] proposed a model for detecting intrusions in SCADA systems. It is cost-sensitive ML model assigned with SMOTE-SVM.

Soulaiman Moualla et al. (2021) [18] suggested a model for real-time intrusion detection based on ELM classifier. SMOTE was used for addressing class imbalance. Feature selection was done using the Extra Trees Classifier. This helps in reducing dimensionality by selecting the most important features with help of Gini Impurity criterion. For classification purposes ELM, a single-layer feedforward neural network was used. With the help of Adam optimizer, the model was optimized and tested on UNSW-NB15, which achieved 98.43% accuracy.

Muhammad Ashfaq et al. (2021) [19] suggested a hybrid DL model combining CNN with RNNs like LSTM and GRU. CNN and RNN capture spatial dependencies and temporal sequences respectively. This model is most effective for detecting patterns. For model optimization, Hyperparameter tuning was used and cross-validation for improving model reliability as well as to prevent overfitting. UNSW-NB15 dataset was used for model testing that resulted in 78.4% accuracy.

Paya, Antonio, et al [20] proposed a novel defense system called Apollon that helps in protecting IDS against AML attacks. Using Apollon, intrusions were identified. Each input required the Thompson sampling algorithm to select the ideal classifier or ensemble of classifiers from Multi-Armed Bandits (MAB) methodology. Apollon allows prevention of adversaries who develop adversarial samples that evade IDS detection by learning its behavior patterns.

The authors Arumalla Raja et al. (2024) [21] introduced an Efficient IDS for cloud computing environments through a HML classifier that unites SVM and ANN technologies. The researchers tested their developed system using CIC-IDS2018 dataset that contains multiple attack types within different types of network traffic information. The system applied two processing techniques to normalize data while selecting features that improved performance while decreasing calculation needs. The HML classifier demonstrated superior performance against Naïve Bayes and alternative traditional classifiers by attaining 96.7% F1-score with 96.25% recall and 97.6% precision as well as 96.54% accuracy.

Tongtong Su [22] suggested DL model combining convolutional layers with a BiLSTM layer for capturing spatial and temporal characteristics of network traffic. Before feature extraction one-hot encoding

and normalization techniques were used followed by an attention-based mechanism for better classification. The implemented model achieved 84.25% NSL-KDD dataset accuracy using Adam optimizer although it demonstrated weaknesses in detecting U2R attacks. The high computational demands of BLSTM together with attention layers prevent their use in real-time system applications.

Zhendong Wang et al. (2021) [23] introduced a new model that combines SDAE with ELM to enhance NIDS performance. Using SDAE enables better feature representation after the network traffic data undergoes noise reduction. The rapid classification process depends on the high-speed ELM learning algorithm. The dual implementation of SDAE and ELM allows the model to process extensive data quantities and deliver immediate detection capabilities. The evaluation utilized KDD Cup99 dataset alongside NSL-KDD dataset which yielded 97.83% and 98.12% accurate results through the use of SDAE-ELM model. Processing large datasets with denoising becomes time-consuming due to the extra complexity which renders this system ineffective in fast-response situations.

The authors of [24] introduced VPI which operates as virtualization-based tool to stop malicious network activities. The VPI framework runs natively within QEMU-KVM virtualization software thus it suits private cloud deployments effectively. VPI installed within QEMU-KVM avoids exposure to attacks that run through the kernel mode. The monitoring function of user-mode applications alongside network card observations lets VPI detect and stop communications from malicious code that operates at the kernel level.

The research paper [25] presents XIoT as a new explainable IoT attack detection model that addresses these challenges. XIoT operates on converted IoT network traffic data into spectrogram images to identify intricate attack signatures. XIoT delivers interpretability through explainable AI mechanisms since it implements these mechanisms as part of the system thus making cybersecurity analysts capable of understanding and trusting prediction results. Through XIoT users gain the ability to create decisive responses for cyber security threats. The model benefits from optical network characteristics which support efficient high-speed processing of IoT data streams in large scale for real-time detection across different IoT settings.

Traditional approaches, such as heuristic methods, suffer from high false positive and poor adaptability to dynamic networks. Existing ML/DL models, like CNN and LSTM, achieve high accuracy but are computationally intensive, limiting their practicality in real-time applications. ARP XGBoost Shield leverages SHAP-based feature selection to identify impactful features and PCA for dimensionality reduction, improving computational efficiency and reducing false positive. The XGBoost classifier ensures robust classification, handling imbalanced datasets effectively while scaling seamlessly to complex environments like IoT and SDN. The SHAP enhancement system provides vital information about feature importance that allows detection methods to be optimized. ARP XGBoost Shield delivers dependable and interpretable ARP spoofing detection through its solutions that resolve scalability dilemmas and efficiency as well as accuracy problems in dynamic network conditions. TABLE I shows the Summary of Acronyms.

TABLE I

SUMMARY OF ACRONYMS

Acronym	Term
AI	Artificial Intelligence
ARIMA	Autoregressive Integrated Moving Average
ARP	Address Resolution Protocol
CNN	Convolutional Neural Network
DoS	Denial of Service
DL	Deep Learning
F1 Score	F1 Score
IDS	Intrusion Detection System
IoT	Internet of Things
IP	Internet Protocol
LAN	Local Area Network
LSTM	Long Short-Term Memory
MAC	Media Access Control
ML	Machine Learning
MiTM	Man-in-the-Middle
PCA	Principal Component Analysis
PCAP	Packet Capture
RF	Random Forest
ROC-Curve	Receiver Operating Characteristic - Area Under Curve
SDN	Software-Defined Networking
SHAP	Shapley Additive Explanations
SVM	Support Vector Machine
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
XG BOOST	Extreme Gradient Boosting

3. PROPOSED METHODOLOGY

A. Overview

The ARP XGBoost Shield system uses Fig. 1 to integrate SHAP with autoencoders through a method that reduces dimensions before employing XGBoost as a classifier process. SHAP performs improved feature interpretation as it provides insights about model prediction reliance on individual features for refinement of the detection system. ARP XGBoost Shield model not only addresses the challenges of detecting ARP spoofing attacks, but it also produces reliable and interpretable results.

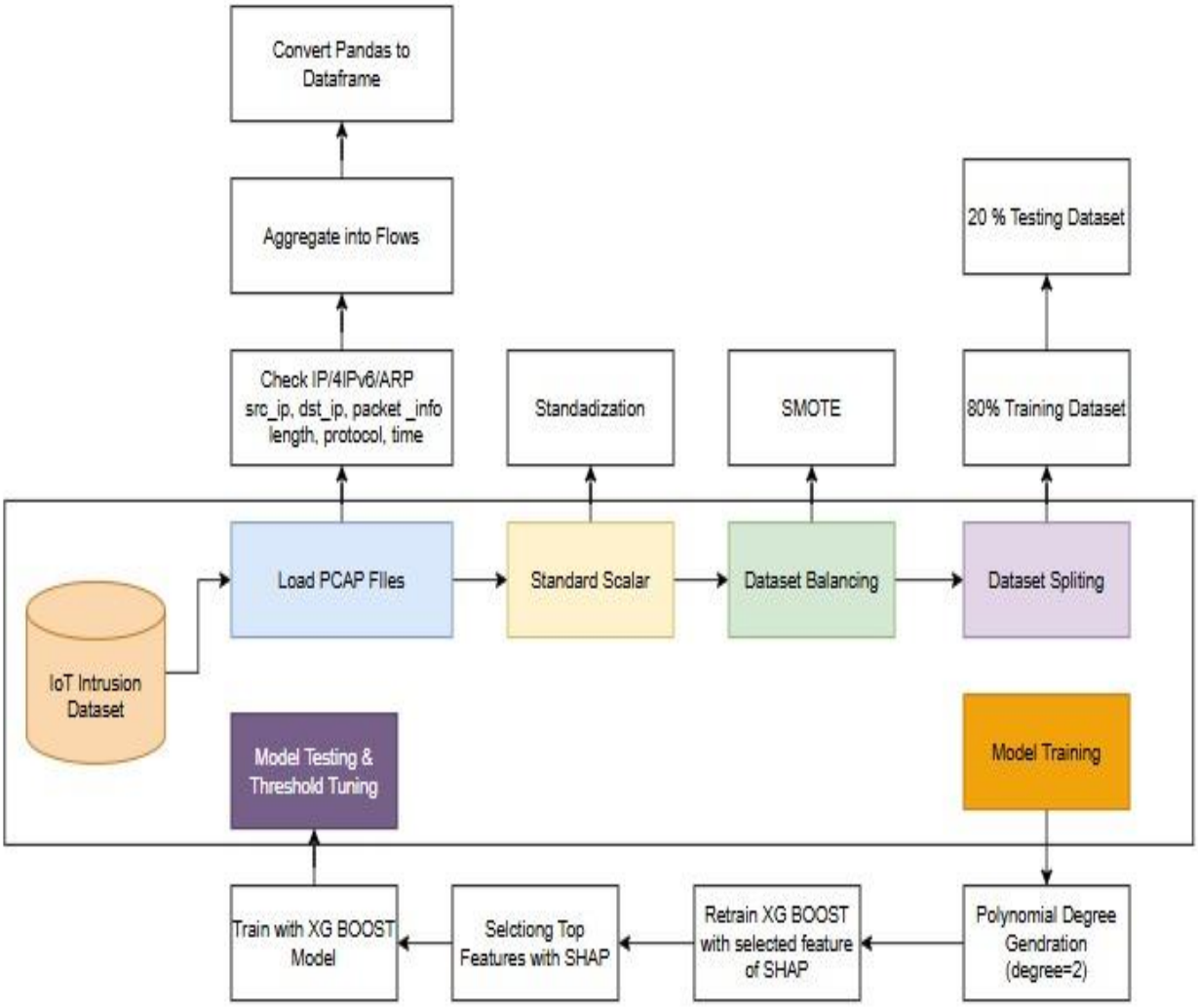

Fig. 1 System Architecture of ARP XGBoost Shield model

B. Data Preprocessing Layers

In initial state of methodology, PCAP are processed to extract critical features from the ARP, IP, and IPv6 layers. Packets are grouped into flows based on their source IP, destination IP, and protocol information. Statistical characteristics like total length, average length, total packets, and flow duration are computed for every flow. The dataset is built around these extracted features. Handling dataset imbalances is critical; therefore, techniques such as Random Oversampling are used to ensure balanced class distribution. Furthermore, textual features such as protocol information are tokenized and converted to numerical representations for seamless integration with subsequent modeling steps. Through preprocessing pipeline, dataset undergoes rigorous cleaning, balancing, and refinement, resulting in a high-quality dataset optimized for training and free from unnecessary noise and data.

C. Standardization

Standardization is process of applying uniform scaling of data because it does not allow any feature with higher numerical range to influence the model. This pre-processing method transforms features to have a standard deviation of one and a mean of zero so that each feature does not produce results disproportionately.

Through this method, it avoids any one feature with a greater numerical range from affecting results disproportionately.

ARP XGBoost Shield model employs standardization via the Standard Scaler from scikit-learn. The fit_transform method is first used to standardize training data, ensuring that each feature in the dataset has zero mean and one standard deviation. The transform method is then applied to the test data, with same scaler as on training data. This ensures that training and testing data are consistent, and that the model evaluates both datasets using the same standardized conditions. This step improvesmodel's training stability and overall predictive accuracy.

D. SMOTE

In SMOTE the minority class gets over-sampled through synthetic sample creation instead of traditional over-sampling operations using original data. A method created by researchers who utilized this strategy effectively for handwritten character recognition (Ha& Bunke, 1997) developed it. The approach resulted in additional training data because researchers only conducted few operations on real data. The minority class received additional training data through synthetic generation along the straight lines that connect its k nearest class members [28]. Random selection happens from k nearest neighbours in order to achieve the necessary over-sampling quantity.

Fig. 2 Visualization Chart - SMOTE

E. PCA

PCA transforms complex, correlated data into a simplified set of orthogonal principal components, prioritized by their ability to capture data variability. Through the retention of the most important components, PCA reduces the dataset while maintaining the bulk of its essential information. This dimensionality reduction improves computational efficiency, diminishes overfitting, and enhances model robustness. Under the recommended methodology, PCA is used to compress data dimensionality after normalization. The PCA method selects and retains the most important features while reducing the overall number of features. The parameter n_components=10 specifies the number of principal components to retain, but this can be changed based on the explained variance to ensure that only the most informative components are kept. The fit_transform function is applied to the scaled training data to fit the PCA model while also

transforming it into a lower-dimensional space. Similarly, the scaled test data is transformed to reduce its dimensionality using the fitted PCA model, ensuring that the training and test data are consistent. PCA reduces data complexity, advances training, and increases generalization performance by filtering out noisy and redundant features.

F. XGBoost model

The ARP XGBoost Shield model depends on the XGBoost classifier because this model demonstrates impressive capabilities in imbalanced data management and protects against overfitting. XGBoost supports as a sophisticated ML tool that combines DT and Gradient Boosting algorithms to develop an flawless and robust predictive model. The performance quality of XGBoost increases steadily through successive elimination of previous prediction mistakes particularly when dealing with unbalanced datasets and preventing overfitting scenarios. XGBoost provides great versatility and effectiveness when analyzing structured and tabular data through its parallel processing mechanism as well as regularization and early stopping features. XGBoost delivers exceptional performance and flexibility through three main capabilities that benefit structured or tabular data analyses.

G. SHAP

SHAP enhances the ARP XGBoost Shield model through its capability to show feature variables responsible for prediction outcomes. The main purpose of SHAP is to enhance models by improving their performance and robustness and transparency [26]. The importance of individual features to detect legitimate versus spoofed ARP packets can be determined through SHAP value evaluation. The model becomes more precise because the data selection focuses on main features which eliminates less essential elements. Threshold adjustment serves as a vital second step in the methodology. The model checks numerous threshold points to achieve balanced sensitivity and specificity performance which fulfills requirements of real-world detection scenarios fighting ARP spoofing attacks. Based on threshold adjustment and SHAP combined methods, trustworthy system for ARP spoofing attack detection with comprehensible results can be achieved.

H. Model Training

PCA-generated polynomial features serve as inputs for training the XGBoost model while its parameters are initialized with them. XGBoost model receives the PCA-generated polynomial features before the model receives fine-tuning through optimized hyperparameters for better performance. The training process utilizes 0.05 learning rate together with six tree depth levels and runs for 500 boosting rounds according to the n_estimators parameter. A scale_pos_weight calculation is performed to achieve balanced predictive modelling of minority class events (spoofed ARP packets). Model was trained on the transformed training data (X_train_poly) and related target labels (y_train). SHAP was used to explain the relevance of features in the model predictions. SHAP Tree Explainer calculates SHAP values, which measure the contribution of every feature towards the model's ultimate decision. To graphically represent feature importance and display the most impactful features towards ARP spoofing detection, a SHAP summary plot is created. Using the SHAP values, the most important ten features are chosen to re-train the model, allowing it to concentrate on the most important features and possibly enhance generalization. The XGBoost model is then retrained with these most important features. The model's performance is assessed by adjusting classification thresholds (0.4 and 0.45), and probability predictions (test_probs) are thresholded to yield binary predictions (test_pred). These predictions are compared to the actual labels (y_test) to determine the model's accuracy and performance. Fig. 3 explains the training workflow of XGBoost.

Fig. 3 XGBoost Training Workflow

4. EVALUATION

A. Benchmark Dataset

IoT Network Intrusion Dataset used as the basis of the intrusion detection approach is proposed. The dataset presents a wide set of network traffic data, which includes 1,264,000 samples of benign and attack activities, each characterized by 47 features. Dataset contains a wide variety of real-world attacks, including DoS/DDoS, MitM, Malware, and Scanning and probing attacks, thus presenting a strong and comprehensive dataset for the assessment of intrusion detection models' effectiveness.

B. Evaluation Metrics

Three evaluation metrics including Accuracy and Precision and Recall enable proper classification of benign and malicious instances when assessing performance in the system. The metrics are evaluated through calculating the following formulas:

1) *Accuracy*: Accuracy is described by Equation (1) as the amount of correct forecasts to the number of instances, which is overall efficiency of the model.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

2) *Precision:* Equation (2) characterizes Precision as proportion of model's capacity to properly classify positive cases, determined by the number of True positive predictions divided by the total number of positive predictions of model.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

3) *Recall:* Equation (3) defines the measure of identifying all positive instances by calculating ratio of correct positive predictions to total positive cases.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

C. Experimental Setup

The experimental setup utilizes TensorFlow with Keras as the backend, running on a Windows 11 system powered by AMD Ryzen 3 3200U processor, Radeon Vega Mobile GPU, and 8GB RAM. To optimize the model's hyperparameters, we performed 100 training iterations on the IoT network intrusion dataset, which comprises 1,264,000 instances.

D. Performance Analysis

In the performance measurement, the threshold values were taken as 0.4 and 0.45, respectively, with the following outcomes. For threshold 0.4, testing accuracy was 93%. Precision and recall for class 1 (malicious) were 0.64 and 0.77, respectively, with an F1 score of 0.70 and the classification report showed high precision (0.94) and recall (0.96) for class 0 (benign), but lower for class 1, as shown by the confusion matrix. At threshold 0.45, the performance was similar, with model having testing accuracy of 94%. accuracy for class 1 reduced to 0.65, and recall dipped to 0.73, resulting in an F1 measure of 0.69.

SHAP Output: Fig. 4 explains that SHAP summary plot gives a good understanding of the recognition of legitimate and spoofed ARP packets in XGBoost model and how various features affect the results. SHAP determines the contribution of each feature to model output and is represented on a horizontal axis. Features' positive SHAP values highlight a packet as spoofed, while negative ones indicate vice versa. The vertical axis ranks features by importance, with the most influential features, for instance, x0 and x7 appearing at the top. The scatterplot shows various dots which reveal the distribution of data with regard to SHAP value of each feature. The color map, which changes from blue (low feature values) to red (high feature values), facilitates visualization of how feature values affect predictions. The red shaded area represents values of x0 that strongly contribute to spoofed packet prediction while small x0 values remain insignificant. The ability to detect spoofed packets depends significantly on the chosen value settings for x0. However, small values of x0 create minimal changes to the detection outcome. Interaction terms, like x3 × x9.

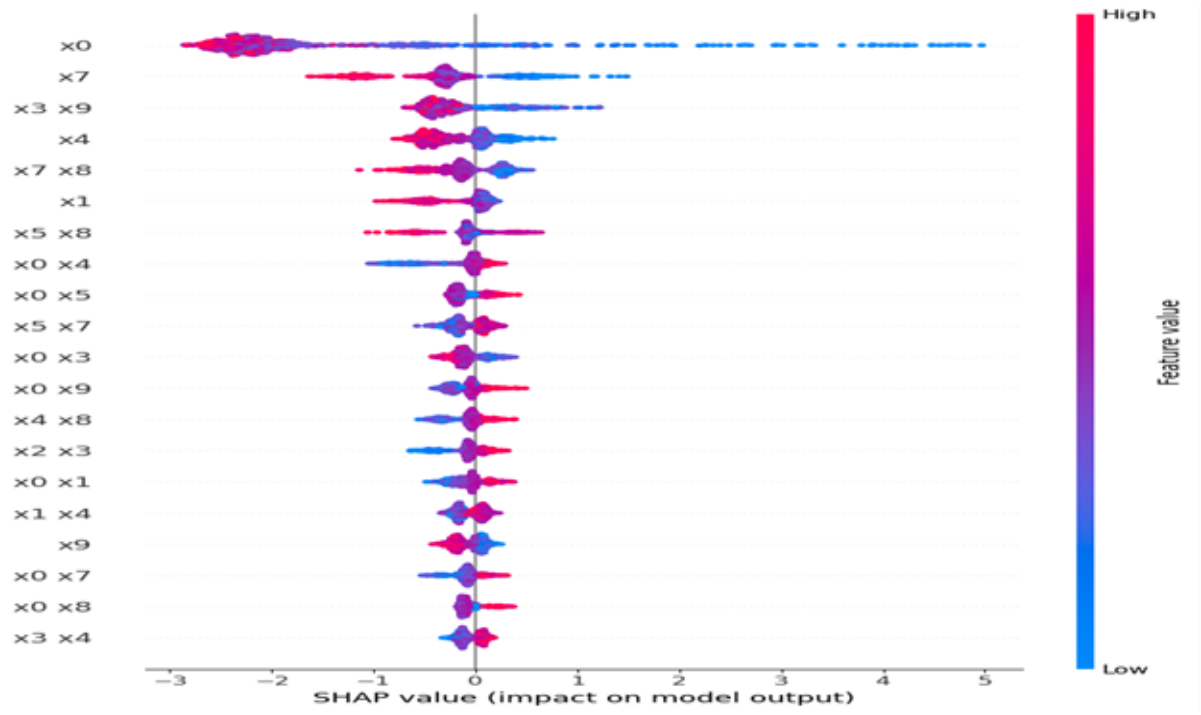

Fig. 4 SHAP Value

4) *Confusion Matrix:* Fig. 5(a) and 5(b) shows confusion matrices for an XGBoost model tested at two classification thresholds, 0.45 and 0.4. Each matrix plots the predicted label versus true label and misclassifies results into four bins: True negative, False positive, False negative, and True positive. With a threshold of 0.45, model correctly classifies 259 negative cases and 12 positive cases and misclassifies 11 negative cases as positive and 18 positive cases as negative.

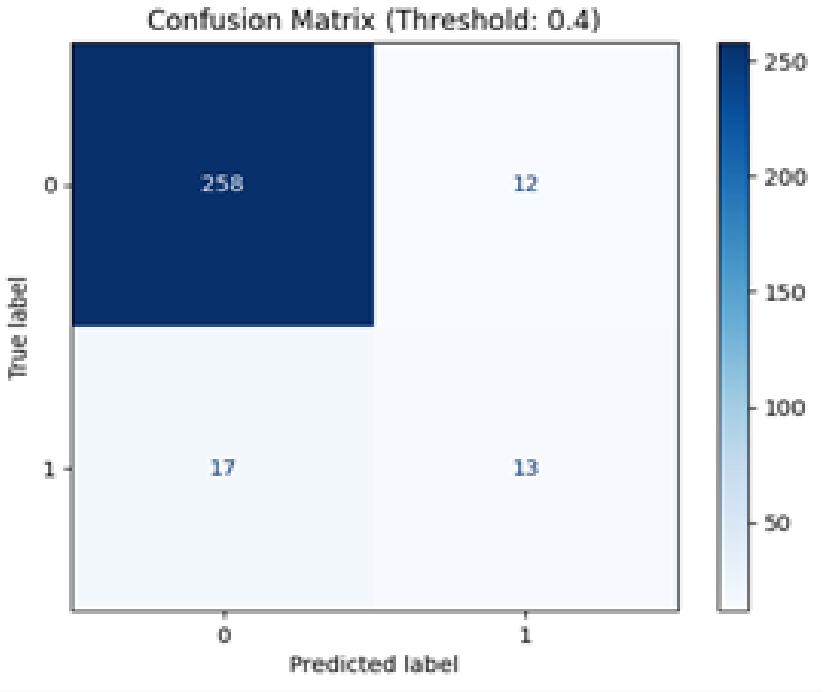

Fig. 5(a) Confusion Matrix of Threshold of 0.4

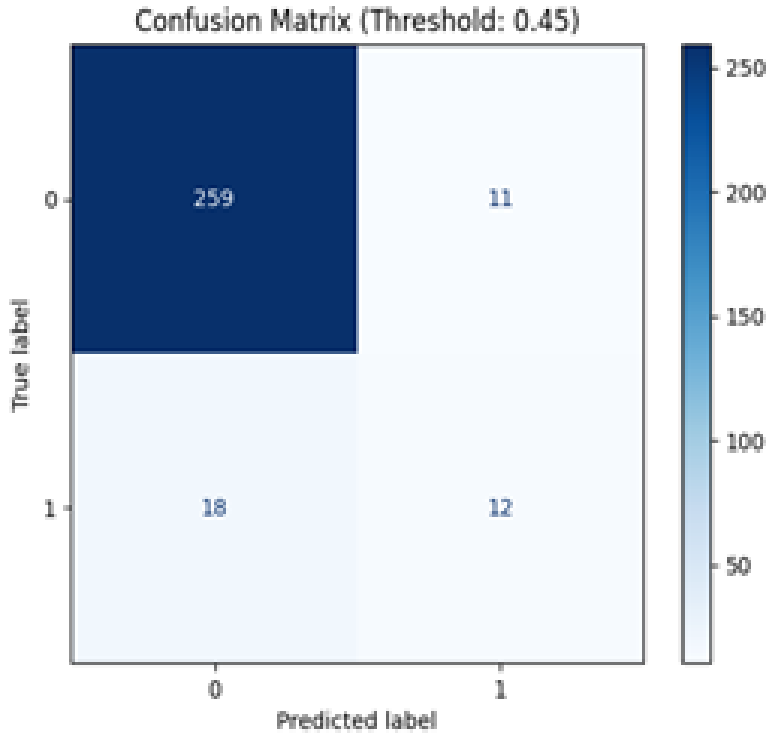

Fig. 5(b). Confusion Matrix of Threshold of 0.45

E. Comparison State of Art

TABLE II illustrates that conventional techniques, including those based on ARIMA statistical processing or basic neural network, meet with reasonable accuracy but are faced with high rates of error and low scalability. For example, methods based on neural network deliver above 90% accuracy but fail to provide

interpretability as well as strength against dynamic IoT scenarios. In contrast, the current approach incorporates PCA for feature reduction, XGBoost for resilient classification, and SHAP for interpretability of features. This blend not only provides greater precision (testing precision of 93%) but also minimizes false positive considerably by concentrating on the most effective features as shown in Fig. 6. Moreover, the efficiency of XGBoost in dealing with imbalanced data and intricate attack patterns makes the system scalable and flexible for real-world network scenarios such as IoT and SDN. In general, the suggested methodology provides a more accurate, interpretable, and scalable solution, overcoming the shortcomings of earlier methods.

TABLE III

PERFORMANCE METRICS OF ARP XGBOOST SHIELD

METRICS	***THRESHOLD 0.4***	***THRESHOLD 0.45***
Accuracy	93	94
Precision	0.62	0.70
Recall	0.80	0.70
F1-Score	0.70	0.70

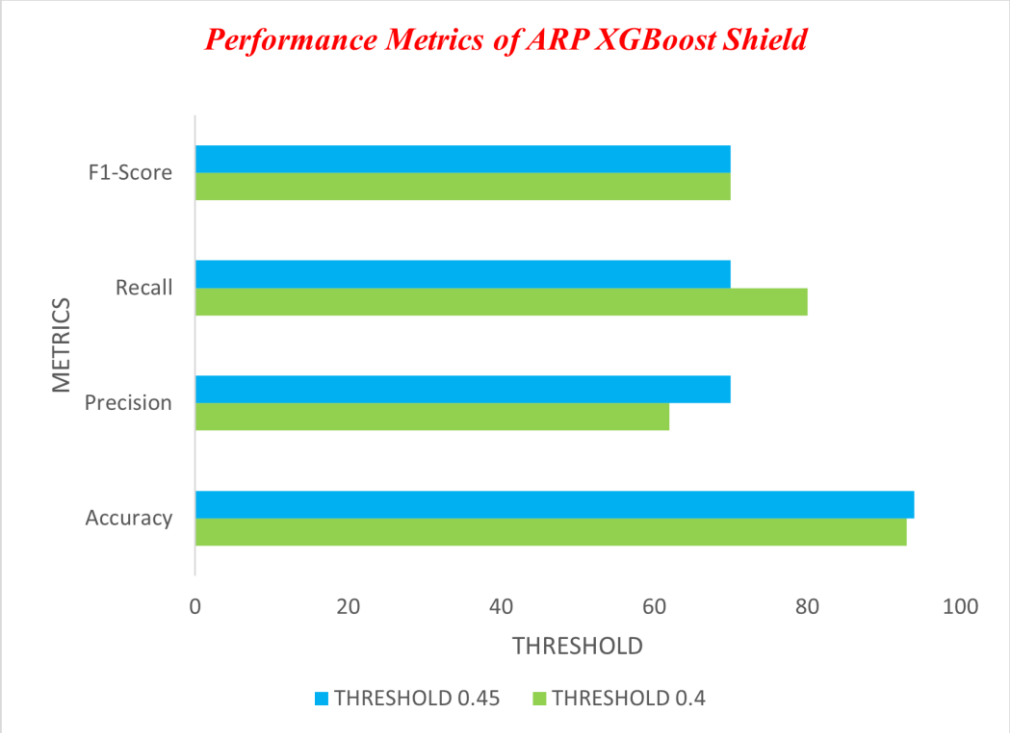

Fig. 6 Performance Comparison of ARP XGBoost Shield at different Thresholds

5. CONCLUSION & FUTUREWORK

The ARP XGBoost Shield model introduced state-of-the-art ARP spoofing detection methodology using the collective strength of PCA, XGBoost, and SHAP to perform feature selection and explainability. With the use of these methods, the system was able to realize strong classification performance. It effectively addressed challenges such as data imbalance and feature importance across diverse network conditions. The experimental results demonstrated strong accuracy in identifying ARP spoofing attacks, particularly in distinguishing between benign and malicious activities. In spite of certain drawbacks in identifying intricate malicious patterns, the approach presented has substantial potential compared to existing techniques. The future plans include optimizing model sensitivity towards sophisticated attack situations and investigating real-time deployment with regard to comprehensive network security solutions.

ACKNOWLEDGMENT

The authors extend their due thanks to the Thiagarajar College of Engineering management, Madurai, India for their extensive research facilities and the financial backing from Thiagarajar Research Fellowship (TRF) scheme (File.no: TCE/RD/TRF/2024/15 dated 09-02-2024) is gratefully acknowledged.

REFERENCES

[1] Husain Abdulla, et al. "*ARP spoofing detection for IoT networks using neural networks*." SSRN Electronic Journal,2020, https://doi.org/10.2139/ssrn.3659129. Accessed 19 Nov.2020..

[2] J. Mrinal kumar and Chandra Sekhar Dash. "*Detecting and preventing ARP spoofing attacks using real-time data analysis and machine learning.*" International Journal of Innovative Research in Computer Science and Technology, vol. 12, no. 5,Sept. 2024, pp. 47–55, https://doi.org/10.55524/ijircst.2024.12.5.7.Accessed 6 Nov. 2024.

[3] Dharani, M., Nivedhidha, M., Sangeetha, A., Saravanan, V.,Ramkumar, M. P., & GSR, E. S. (2024, October). "*Detection of ARP spoofing with optimized false alarm using deep learning based absolute thresholding*" in 2024 4th International Conference on Sustainable Expert Systems (ICSES) (pp.1650-1657). IEEE

[4] Yuwei Sun , et al. "*Suspicious ARP activity detection andclustering based on autoencoder neural networks*." 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), 8 Jan.2022,https://doi.org/10.1109/ccnc49033.2022.9700697.

[5] Ramkumar, M. P., Daniya, T., Paul, P. M., & Rajakumar, S. (2022). *Intrusion detection using optimized ensemble classification in fog computing paradigm.* Knowledge-Based Systems, 252, 109364.

[6] Mehak Usmani, et al. *Predicting ARP spoofing with machine learning*. 23 Sept. 2022, https://doi.org/10.1109/icetst55735.2022.9922925

[7] Prasanna, S. S., Selvan, G. E., & Ramkumar, M. P. (2023, July). *Anomaly-based intrusion detection system for ICS*. In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-4). IEEE

[8] Lirim Ashiku, and Cihan Dagli. "*Network intrusion detection system using deep learning*." Procedia Computer Science, vol. 185, 2021, pp. 239–247, https://doi.org/10.1016/j.procs.2021.05.025.

[9] Yanfang Fu, et al. "*A deep learning model for network intrusion detection with imbalanced data*." Electronics, vol. 11, no. 6, 14 Mar. 2022, p. 898,https://doi.org/10.3390/electronics11060898.

[10] Sydney Kasongo, Mambwe. "*A deep learning technique for intrusion detection system using a recurrent neural networks based framework*."Computer Communications, Dec.2022, https://doi.org/10.1016/j.comcom.2022.12.010.

[11] Prasanna, S. S., Emil Selvan, G. S. R., & Ramkumar, M. P. (2023, March). *Protocol anomaly detection in IIoT*. In International Conference on Advanced Computing, Machine Learning, Robotics and Internet Technologies (pp. 37-46). Cham: Springer Nature Switzerland.

[12] Fatima Ezzahra Laghrissi et al. "*Intrusion detection systems using long short-term memory (LSTM).*" Journal of Big Data, vol. 8, no. 1, 7 May 2021, https://doi.org/10.1186/s40537-021-00448-4.

[13] Chao Liu, et al. "*A hybrid intrusion detection system based on scalable k-means+ random forest and deep learning*." IEEE Access, vol. 9, 2021, pp. 75729–75740, https://doi.org/10.1109/access.2021.3082147.

[14] Logeswari, G., et al. "*An intrusion detection system for SDN using machine learning.*" Intelligent Automation & Soft Computing, vol. 35, no. 1, 2023, pp. 867–880, https://doi.org/10.32604/iasc.2023.026769.

[15] Sams Aafiya Banu, S., Gopika, B., Esakki Rajan, E., Ramkumar, M. P., Mahalakshmi, M., & Emil Selvan, G. S. R. (2022, March*). Smote variants for data balancing in intrusion detection system using machine learning*. In International Conference on Machine Intelligence and Signal Processing (pp. 317-330). Singapore: Springer Nature Singapore.

[16] Mighan, Soosan Naderi, and Mohsen Kahani. "*A novel scalable intrusion detection system based on deep learning*." International Journal of Information Security, 15 June 2020, https://doi.org/10.1007/s10207- 020-00508-5.
[17] Mahalakshmi, M., Ramkumar, M. P., & GSR, Emil. Selvan. (2022, December). *Scada intrusion detection system using cost sensitive machine learning and smote-svm*. In 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N) (pp. 332-337). IEEE.
[18] Soulaiman Moualla , et al. "*Improving the performance of machine learning-based network intrusion detection systems on the UNSW-NB15 dataset*." Computational Intelligence and Neuroscience, vol. 2021, 15 June 2021, pp. 1–13, https://doi.org/10.1155/2021/5557577.
[19] Muhammad Ashfaq Khan. "*HCRNNIDS: Hybrid Convolutional Recurrent Neural Network-Based Network Intrusion Detection System*." Processes, vol. 9, no. 5, 10 May 2021, p. 834, https://doi.org/10.3390/pr9050834.
[20] Paya, Antonio, et al. "*Apollon: A robust defense system against adversarial machine learning attacks in intrusion detection systems*." Computers & Security, vol. 136, 1 Jan. 2024, p. 103546, www.sciencedirect.com/science/article/pii/S016740482300456X? via%3Dihub, https://doi.org/10.1016/j.cose.2023.103546.
[21] Arumalla Raja , et al. " *A novel efficient intrusion detectionsystem in Cloud Using Hybrid Machine Learning Classifier*."
[22] Tongtong Su, et al. "*BAT: deep learning methods on network intrusion detection using NSL-KDD dataset*." IEEE Access, vol. 8, 2020, pp. 29575–29585, https://doi.org/10.1109/access.2020.2972627.
[23] Zhendong Wang, et al. "*Intrusion detection methods based on integrated deep learning model*." Computers & Security, Jan. 2021, p. 102177, https://doi.org/10.1016/j.cose.2021.102177.
[24] Erez Shlingbaum, Raz Ben Yehuda, Michael Kiperberg, Nezer Jacob Zaidenberg. *Virtualized network packet inspection*, Computer Networks, 2024
[25] Nouman Imtiaz, Abdul Wahid, Syed Zain Ul Abideen, Mian Muhammad Kamal et al. "*A deep learning-based approach for the detection of various internet of things intrusion attacks through optical networks*" , Photonics, 2025
[26] Usman Ahmed, Zheng Jiangbin, Ahmad Almogren, Muhammad Sadiq, Ateeq Ur Rehman, M. T. Sadiq, Jaeyoung Choi. "*Hybrid bagging and boosting with SHAP based feature selection for enhanced predictive modeling in intrusion detection systems*" , Scientific Reports, 2024
[27] V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar,V. Vennila. "*Challenges in information, communication and computing technology*" , CRC Press, 2024.
[28] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *SMOTE: Synthetic minority over-sampling technique*. The Journal of Artificial Intelligence Research, *16*, 321–357. https://doi.org/10.1613/jair.953