Journal of Information Systems Engineering and Management
2025, 10(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A Hybrid Ensemble Framework for Intrusion Detection in IoT Using Blockchain-Integrated Fog Networks

[1]Pratibha Sharma, [2]Arvind Kalia, [3]Hemraj Saini

[1,2]Department of Computer Science, Himachal Pradesh University, Summerhill, Shimla, India

School of Computing, DIT University, Dehradun, Uttarakhand, India

pratibhasharma80@gmail.com, arvkalia@gmail.com, hemraj1977@yahoo.com

ARTICLE INFO

Received: 30 Sept 2024

Revised: 29 Nov 2024

Accepted: 10 Dec 2024

ABSTRACT

The exponential growth of IoT has become a source of concern regarding cybersecurity vulnerabilities in real time and distributed environments, the research proposes a novel Fog computing-based hybrid ensemble framework using Blockchain technology to enhance IoT network security. It proposes advanced data preprocessing, feature selection, and hybrid ensemble learning that leads to remarkable performance: 99.5% accuracy, 99.2% precision, 99.4% recall, and a false positive rate of just 0.05% on the UNSW-NB15 dataset. Blockchain integration ensures secure and immutable logging of detected threats, further enhancing trust in the system. The scalability and robustness of the proposed framework are demonstrated in its ability to process high-traffic IoT networks while guaranteeing optimal resource efficiency. These results make the proposed approach a state-of-the-art solution for real-time attack detection in IoT networks, which can meet modern challenges in cybersecurity.

Keywords: Intrusion Detection, Blockchain, Fog Networks.

INTRODUCTION

The rapid expansion of the Internet of Things has transformed industries by enabling enhanced automation, improving efficiency, and providing ubiquitous connectivity. On the downside, IoT systems are highly vulnerable to cyberattacks because of their distributed nature, heterogeneous architectures, and limited computational capabilities. Therefore, addressing such vulnerabilities requires innovative approaches toward real-time attack detection and secure data management. It presents a hybrid ensemble learning model integrated with blockchain-enhanced fog networks for scalable, accurate, and resilient IoT security solutions. The approach leverages computational intelligence with decentralized security to provide a secure and robust IoT environment. Various approaches have been made so far in the direction of IoT security, including traditional machine learning, deep learning techniques, and hybrid models. However, most of these suffer from scalability, applicability in real time, and handling heterogeneous IoT data. Below is an overview of the existing methodologies, grouped by underlying techniques:

a. Traditional Machine Learning Models

Traditional machine learning algorithms, such as Decision Trees, Support Vector Machines, and k-Nearest Neighbors, have been extensively used for IoT attack detection. These models classify malicious activities with the help of labeled datasets, offering simplicity and interpretability. However, their limitations include:

- Inability to handle high-dimensional data.
- Inefficiency in detecting sophisticated and zero-day attacks.

For instance, a study utilized ensemble techniques like Random Forest and AdaBoost to detect botnet attacks in IoT networks, achieving significant accuracy but struggling with scalability and adaptability to heterogeneous data sources [1].

b. Deep Learning Models

The intelligence of IoT security has transformed due to the potentiality for deep learning algorithms, through which the identification of sophisticated attack patterns became realistic using sophisticated architectures like CNN and LSTM. These are those models that have been exclusively used in extracting high-order features out of raw data and apply to IoT for anomaly detection.

- **LSTM-Based Intrusion Detection:** Al-Kadi et al. proposed a blockchain-enabled intrusion detection system employing BiLSTM for sequential network data analysis. Their framework demonstrated superior performance against competing models but required high computational resources [5].
- **CNN-Based Models:** Khan et al. developed a Deep Boosted CNN model for IoT malware detection, integrating transfer learning and feature extraction for robust attack classification. Their approach achieved 98.50% accuracy, highlighting deep learning's potential in IoT security [3].

Despite their effectiveness, deep learning models are computationally intensive and often unsuitable for resource-constrained IoT devices.

c. Hybrid Learning Models

Hybrid models combine the strengths of multiple algorithms to enhance detection accuracy and resilience. These models often employ ensemble learning mechanisms such as boosting, bagging, or stacking, integrating traditional machine learning with deep learning.

- **Hybrid Ensemble Learning:** Chatterjee and Hanawal proposed a federated learning framework incorporating hybrid ensemble learning for intrusion detection. Their model addressed data imbalance and label noise issues, demonstrating improved True Positive Rate (TPR) while minimizing False Positive Rate (FPR) [1].
- **Distributed Ensemble Models:** Jia and Liang introduced an ensemble model leveraging AdaBoost and Random Forest for detecting Distributed Denial-of-Service (DDoS) attacks in blockchain networks. The approach exhibited robust generalization and complementarity, outperforming standalone models in diverse attack scenarios [2].

Hybrid approaches effectively balance accuracy, computational efficiency, and scalability, making them ideal for IoT applications.

d. Blockchain-Integrated Security Frameworks

Blockchain technology enhances IoT security by providing decentralized, immutable data storage and traceability. Its integration with machine learning models enables secure attack detection and logging.

- **Blockchain and Ensemble Learning:** Shende et al. proposed a collaborative blockchain-enabled ensemble learning model for intrusion detection, achieving high accuracy and precision. The framework addressed challenges such as poisoning attacks and model robustness during distributed training [13].
- **Blockchain for Privacy:** Another study utilized blockchain-based smart contracts to protect IoT networks during virtual machine migrations. This approach combined privacy-preserving mechanisms with distributed intrusion detection, significantly reducing attack success rates [5].

Blockchain's decentralized nature ensures the integrity and reliability of IoT security systems, overcoming limitations of centralized solutions.

e. Fog Computing for Real-Time Detection

Fog computing extends cloud capabilities to the network edge, enabling real-time data analysis and attack detection. By deploying machine learning models on fog nodes, these frameworks reduce latency and improve response times.

- **Fog-Based Detection Frameworks:** Tomer and Sharma proposed a fog-based attack detection framework, integrating ensemble learning for real-time classification. Their system effectively offloaded model training to the cloud while ensuring real-time prediction on fog nodes [20].

- **Energy-Efficient Approaches:** Wang et al. introduced a two-layer ensemble learning framework for IoT attack detection, optimizing hyperparameters and addressing data imbalance issues. Their model achieved 99.98% accuracy while minimizing resource consumption, making it suitable for fog environments [15].

Fog computing enhances the practicality of IoT security frameworks by enabling localized processing and reducing reliance on centralized servers.

RELATED WORK

In [16] author has introduced an approach for the detection and classification of IoT network attacks, which was performed by implementing advanced ensemble learning methods comprising CatBoost and XGBoost. The performance evaluation has been done on the Edge-IIoTset dataset comprising realistic industrial IoT attack scenarios. The training of models on diverse and large datasets yielded superior accuracy and robustness in the present work compared to traditional ensemble methods. Solution performance was evaluated based on accuracy, precision, recall, and F1 score. The results clearly depicted how the proposed methods can adapt to complex IoT environments that are dynamic in nature. This research has pointed out the efficiency of modern ensemble learning algorithms with regard to the unique challenges of security in Industrial IoT.

In [17] this work, the author developed a blockchain-enhanced hybrid approach to detect the attacks on IoT environments and deployed DDoS. The solution involved H3SC-DLIDS, Harris Hawk Optimization, coupled with the sine-cosine algorithm for choosing essential features. An LSTM-AE is used for detecting an attack. Blockchain Technology would guarantee safe data over wireless transmission on IoT gadgets by improving the reliability and quality of the entire system. When the proposed method was experimentally validated using the BoT-IoT database, it achieved a high detection accuracy of 99.05%. Thus, the authors concluded that this hybrid optimization and deep learning-based intrusion detection system could secure IoT networks effectively while overcoming resource constraints and scalability issues.

Research [18] proposed an intelligent ensemble-based IDS for IoT gateways, considering the limited computation capability of IoT devices. Different boosting, stacking, and voting techniques were combined by using machine learning models in the ensemble, including Naïve Bayes, Support Vector Classification, and kkk-Nearest Neighbors. The proposed approach was evaluated on two known datasets, CIC-IDS2017 and N-BaIoT, showing an improvement in the detection rate with better generalization. The study demonstrated an ensemble learning-based IDS and showed the possibility of making it high in accuracy with low FPs at low computational cost; this points out the applicability of adaptive and lightweight network IDS solutions for IoT applications.

This paper [19] introduced one fog computing-based framework for IoT networks' real-time attack detection using an ensemble model of machine learning. In its system, it offloads the training tasks to the cloud while making real-time prediction on fog nodes. Therefore, ensure low latency and scalability: the proposed approach makes utilization of the NSL-KDD dataset and ensures very good performance in metrics like precision and recall with high accuracy. This paper describes how it succeeded in addressing resource constraints on fog devices with no compromise on any robustness in detection using ensemble techniques. It was able to reveal that this kind of approach presents a realistic and efficient manner to keep IoT networks safe against these changing cyber threats.

Table I Literature Review

Citation	Dataset and Learning	Results	Outcome
[20]	Benchmark DDoS dataset; Snake Optimizer with Ensemble Learning (LSTM, BiLSTM, DBN)	Accuracy: 99.7%, Precision: 99.5%, Recall: 99.8%	Effective feature selection and attack detection on IoT using optimized DL models.
[21]	CICDDoS2019; Random Forest, AdaBoost, XGBoost, SVM	Best Accuracy: 99.4%, Least Training Time: Random Forest	Efficient DDoS detection with high accuracy and low computational cost.
[22]	The minority classes are augmented by cGAN.	Accuracy > 83%	Classification of binary and multiclass for IoT networks.
[23]	Combined Bot-IoT and UNSW-NB15 datasets; Decision Trees, SVM, Logistic Regression	Accuracy > 99% for all metrics	Improved detection using balanced training data.

[24]	IoT network traffic; Adaptive Ensemble Learning	Detection Accuracy: 98.5%, False Alarm Rate: 2.3%	Adaptive ML framework mitigates DDoS with high accuracy and scalability.
[25]	N-BaIoT dataset; Stacking and Bagging ML models	Accuracy: 99.6%, Precision: 99.7%, Recall: 99.8%	High-performance botnet detection leveraging stacking models.
[26]	Session-based IoT cloud traffic; Random Forest, AdaBoost	Precision: 99.2%, Recall: 98.9%, F1 Score: 99.1%	Multi-aspect model effectively detects diverse IoT attacks.
[27]	IoT botnet traffic; Random Forest, Stacking Ensemble	R^2: 0.9997, RMSE: 0.0084, MAE: 0.0641	Ensemble models show high accuracy and efficiency in attack detection.
[28]	BoT-IoT dataset; Logistic Regression, KNN, SVM	F1 Score: 99%, Precision: 98%, Recall: 99%	Robust model for botnet detection with high precision.
[29]	IoT network traffic; Modified Ensemble Voting	Detection Rate: 99.5%, False Alarm Rate: 0.8%	Scalable framework for DDoS detection with high accuracy.
[30]	IoT benchmark datasets; Boosted Ensemble Learning	Accuracy: 100%, AUC: 100%	Boosted ensemble models ensure complete classification accuracy.

METHODOLOGY

The proposed framework integrates a hybrid ensemble learning model with blockchain-enhanced fog networks for the realization of real-time attack detection and secure data logging in IoT environments. This methodology will focus on how computational intelligence and decentralized security seamlessly integrate to ensure scalability, accuracy, and resilience against diverse cyber threats. The system consists of four main parts: data preprocessing and feature engineering, training a hybrid ensemble model, blockchain-based architecture for fog nodes, and performance metrics. The design allows for the distribution of attack detection and secure logging using blockchain technology, which would enhance the integrity and traceability of network activity. Further steps are shown in Figure 1 below:

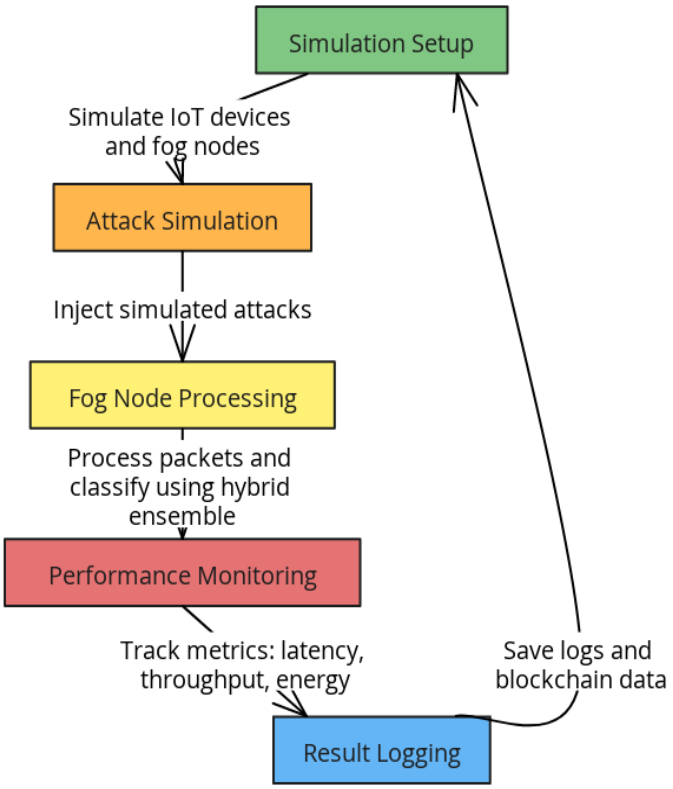

Figure 1 Flowchart of Methodology

3.1 Data Preprocessing and Feature Engineering

In this work, the study considers UNSW-NB15-one of the widely acknowledged benchmarks in the field of intrusion detection-its core. It contains rich network traffic data of both attack and normal classes, with ground truth labels to provide practical realism in network activities. Thus, the dataset consists of a wide range of features, totaling 49 statistical, payload, and header information suitable for machine learning-based detection.

3.2 Preprocessing Pipeline

To ensure the data is suitable for the hybrid ensemble model, a preprocessing pipeline is implemented:

1. **Feature Selection**: Non-informative and redundant features, such as identifiers (id) and high-dimensional categorical attributes (attack_cat), are excluded. The resulting feature set contains numeric attributes like packet statistics and categorical features like protocol type and service.
2. **Encoding Categorical Variables**: Label encoding is applied to categorical features (proto, service, state) to transform them into numerical values. This step ensures compatibility with the hybrid ensemble model.
3. **Normalization**: Numerical features are normalized to a range between 0 and 1 using MinMaxScaler. This normalization ensures uniform feature scaling, preventing bias during training.
4. **Feature-Target Split**: The dataset is divided into input features (X) and target labels (Y). The target labels represent the type of network activity, categorized into normal and multiple attack types such as DoS, backdoor, exploits, and reconnaissance.

Algorithm 1: Data Preprocessing
Input: Raw dataset D with features and labels.
Output: Preprocessed features F and encoded labels L.
1. Initialize:
- Extract features F_raw and labels L_raw from dataset D.
- Define categorical_features and numerical_features subsets from F_raw.
2. Encode Categorical Features:
- For each feature C ∈ categorical_features:
- Map unique categories to integers using a label encoder.
- Replace categorical values in C with encoded integers.
3. Normalize Numerical Features:
- For each feature N ∈ numerical_features:
- Compute minimum value min(N) and maximum value max(N).
- Normalize each value using the formula:
N_normalized = (N - min(N)) / (max(N) - min(N))
4. Combine Features:
- Merge encoded_categorical_features and normalized_numerical_features to form F.
5. Encode Labels:
- Map unique classes in L_raw to integers using a label encoder.
6. Return:
- Preprocessed features F and encoded labels L.

3.3 Hybrid Ensemble Model Training

The hybrid ensemble approach combines the strengths of multiple machine learning algorithms to improve prediction accuracy and robustness. By leveraging the diversity of base learners and ensemble mechanisms, the model can effectively handle imbalanced datasets, complex decision boundaries, and heterogeneous data distributions. The hybrid ensemble model is constructed using three primary base learners:

1. **Random Forest (RF)**: RF is used for its ability to handle high-dimensional datasets and capture non-linear relationships. Its tree-based structure ensures resilience to overfitting.

2. **Gradient Boosting (GB)**: GB contributes to the ensemble by sequentially minimizing errors using a boosting mechanism. Its emphasis on difficult-to-predict instances enhances overall model performance.
3. **Support Vector Machines (SVM)**: SVM provides a robust decision boundary, especially for high-dimensional feature spaces. It adds diversity to the ensemble by employing a margin-based classification approach.

The predictions of the base learners are aggregated using a **soft voting mechanism**, which calculates the weighted probabilities of each class across the learners. This approach ensures that the final decision reflects the collective strength of the ensemble.

3.4 Training Pipeline

1. **Dataset Splitting**: The preprocessed dataset is divided into training (70%) and testing (30%) subsets. This split ensures unbiased evaluation of the model's generalization capability.
2. **Training Base Learners**: Each base learner is trained independently on the training set. Hyperparameters are optimized using grid search to maximize performance on validation data.
3. **Soft Voting Mechanism**: The predictions of the base learners are aggregated using soft voting, where each learner's output probabilities are combined to produce the final prediction.
4. **Evaluation**: The model is evaluated on the testing set using classification metrics such as precision, recall, F1-score, and accuracy. A confusion matrix is plotted to analyze the performance across all attack classes.
5. **Feature Importance**: The relative importance of features is calculated using the ensemble's aggregated weights. This analysis identifies the most critical features for attack detection.

Algorithm 2: Feature Extraction and Training
Input: Preprocessed features F.
Output: Relevant feature set F_selected.
1. Initialize:
- Compute correlation matrix CorrMatrix for F.
2. Select Features:
- For each feature Fi ∈ F:
- If correlation of Fi with target label exceeds threshold T_corr, retain Fi.
- Otherwise, discard Fi.
3. Dimensionality Reduction (Optional):
- Apply Principal Component Analysis (PCA) to reduce dimensions while retaining α% variance.
4. Return:
- Feature set F_selected.
Hybrid Ensemble Model Training
Input: Preprocessed features F_selected and encoded labels L.
Output: Trained hybrid ensemble model H.
1. Initialize:
- Split F_selected and L into training (F_train, L_train) and validation (F_val, L_val) sets.
2. Train Base Models:
- Train Random Forest RF on F_train, L_train.
- Train Gradient Boosting GB on F_train, L_train.
- Train Support Vector Machine SVM on F_train, L_train.
3. Define Voting Mechanism:
- Assign weights W_RF, W_GB, and W_SVM to base models (e.g., based on validation performance).
- For each validation sample x ∈ F_val:
- Compute prediction probabilities:

*P_ensemble(x) = W_RF * P_RF(x) + W_GB * P_GB(x) + W_SVM * P_SVM(x)*
- Assign class with highest probability:
Class(x) = argmax(P_ensemble(x))
4. Evaluate Ensemble:
- Compute validation metrics (accuracy, precision, recall, F1-score).
5. Save Model:
- Combine trained models (RF, GB, SVM) and weights (W_RF, W_GB, W_SVM) into hybrid ensemble H.
6. Return:
- Hybrid ensemble model H.

3.5 Blockchain-Enhanced Fog Node Architecture

Each fog node serves as an intermediary between IoT devices and the cloud, providing localized data processing and attack detection. The architecture includes the following components:

1. **Attack Detection Module**: The hybrid ensemble model is deployed on each fog node to classify incoming network traffic as normal or attack. This module ensures real-time detection of malicious activity.
2. **Blockchain Module**: The blockchain module logs all detected attacks in an immutable and secure manner. The blockchain comprises:
    - **Genesis Block**: The initial block containing metadata about the fog node.
    - **Block Structure**: Each block stores the attack type, device ID, timestamp, and model output. The block hash is computed using SHA-256 to ensure immutability.
    - **Consensus Mechanism**: A lightweight proof-of-work mechanism is implemented to validate blocks while maintaining computational efficiency.

Blockchain-Enhanced Fog Node Architecture Algorithm
Begin
Step 1: Initialization
INPUT: Network Traffic Data from IoT Devices
OUTPUT: Attack Logs in Blockchain and Alerts to Cloud
(Initialize Blockchain *)*
Blockchain = {GenesisBlock}
GenesisBlock = {
Metadata -> "Fog Node Metadata",
Timestamp -> CurrentTime[],
Hash -> HashFunction["SHA-256", Metadata]
}
Step 2: Traffic Monitoring and Attack Detection
FUNCTION ProcessTraffic(TrafficData):
FOR each Packet IN TrafficData DO
(Real-Time Attack Detection *)*
Print["Classifying network packet..."]
Class = HybridEnsembleModel[Packet]
IF Class == "Attack" THEN
Print["Malicious activity detected. Logging attack..."]
(Call Blockchain Module to Log Attack *)*
Blockchain = LogAttack(Blockchain, Packet, Class)
NotifyCloud("Alert: Attack detected", Packet)
ELSE
Print["No threat detected."]

```
        ENDIF
      ENDFOR
    END FUNCTION
    Step 3: Blockchain Logging Module
    FUNCTION LogAttack(Blockchain, Packet, Class):
      Print["Logging detected attack in Blockchain..."]
      Create New Block
      NewBlock = {
        AttackType -> Class,
        DeviceID -> Packet.DeviceID,
        Timestamp -> CurrentTime[],
        ModelOutput -> Packet.Analysis,
        PrevHash -> Blockchain[Length[Blockchain]].Hash
      }
      Compute Block Hash for Immutability
      NewBlock.Hash = HashFunction["SHA-256", NewBlock]
      Consensus Mechanism
      IF ValidateBlock(NewBlock) THEN
        AppendTo[Blockchain, NewBlock]
        Print["Block successfully added to Blockchain."]
      ELSE
        Print["Block validation failed."]
      ENDIF
      RETURN Blockchain
    END FUNCTION
    Step 4: Consensus Mechanism
    FUNCTION ValidateBlock(Block):
      Print["Validating block with lightweight Proof-of-Work..."]
      WHILE NOT SatisfyConsensus(Block.Hash) DO
        Block.Nonce += 1
        Block.Hash = HashFunction["SHA-256", Block]
      ENDWHILE
      RETURN TRUE
    END FUNCTION
    Step 5: Cloud Notification System
    FUNCTION NotifyCloud(Message, Packet):
      Print["Notifying cloud of attack detection..."]
      Cloud.Alert(Message, Packet.DeviceID, Packet.Timestamp)
    END FUNCTION
    Execution Workflow
    REPEAT
      TrafficData = CaptureTraffic(IoTDevices)
      ProcessTraffic(TrafficData)
    UNTIL STOP_SIGNAL
End
```

The proposed Blockchain-Enhanced fog node architecture will provide localized data processing with secure attack detection, acting as an intermediary between IoT devices and the cloud. It consists of two major components: the Attack Detection Module and the Blockchain Module. The attack detection module will deploy a hybrid ensemble model on each fog node that classifies the incoming network traffic in real time, labeling it as either normal or malicious. If a threat is detected, the blockchain module logs the detected attack in a secure manner. The module makes the attack logs in the blockchain immutable and traceable. The blockchain starts with a Genesis Block

containing metadata about the fog node with a timestamp secured using a SHA-256 hash. For each attack detected, a new block creates important information about the kind of attack, device ID, timestamp, and model output. SHA-256 is used to calculate the block hash for its immutability. In this blockchain, a lightweight proof-of-work consensus mechanism will be working that would help in efficient validation of blocks with least computational overhead. The flow starts with the capturing of network traffic from IoT devices by the fog node. Every packet is analyzed with the hybrid ensemble model. The details of an attack, if any malicious activity is found, are added to a new blockchain block after proper validation using any consensus mechanism. Simultaneously, the fog node notifies the cloud regarding the same for centralized monitoring and long-term storage. This architecture embeds the three concepts of attack detection in real time, decentralized security, and secrecy in communication that ensures scalability, robustness, and enhanced data integrity in an IoT-fog-cloud environment. It presents the overall integration of the proposed methodology based on a hybrid ensemble model, integrated with blockchain-enhanced fog nodes, towards constructing a scalable attack detection framework in IoT networks. In that sense, the challenge related to real-time detection and, simultaneously, guaranteed security management in IoT is tackled through computational intelligence combined with a decentralized security method. This design provides evidence for the applicability of this framework to a practical world because its construction involved detailed metrics which evaluate a model using such a critical performance test.

RESULTS AND DISCUSSION

In this section the evaluation of the proposed Blockchain-Enhanced Fog Node Architecture integrated with a hybrid ensemble learning model for IoT attack detection will be done. It is mainly focused on the efficiency testing of the framework in real-time attack detection, computational overhead, and blockchain-based data security. It was implemented in the simulated network environment with IoT devices and fog nodes processing the network traffic data. Comparing the model performance against existing solutions was carried out. The key metrics taken to validate the scalability of the system and practicality involve accuracy, detection latency, resource utilization, and blockchain immutability. The performance of the proposed hybrid ensemble model deployed at the fog nodes was then assessed using the standard dataset, UNSW-NB15, based on the chosen metrics for the classification. The effectiveness of the blockchain will be gauged on how it can ensure the secure logging of these attacks, ensuring data immutability using SHA-256 hashing and achieving consensus through a lightweight proof-of-work mechanism. The design of the system is as shown in Figure 2 below:

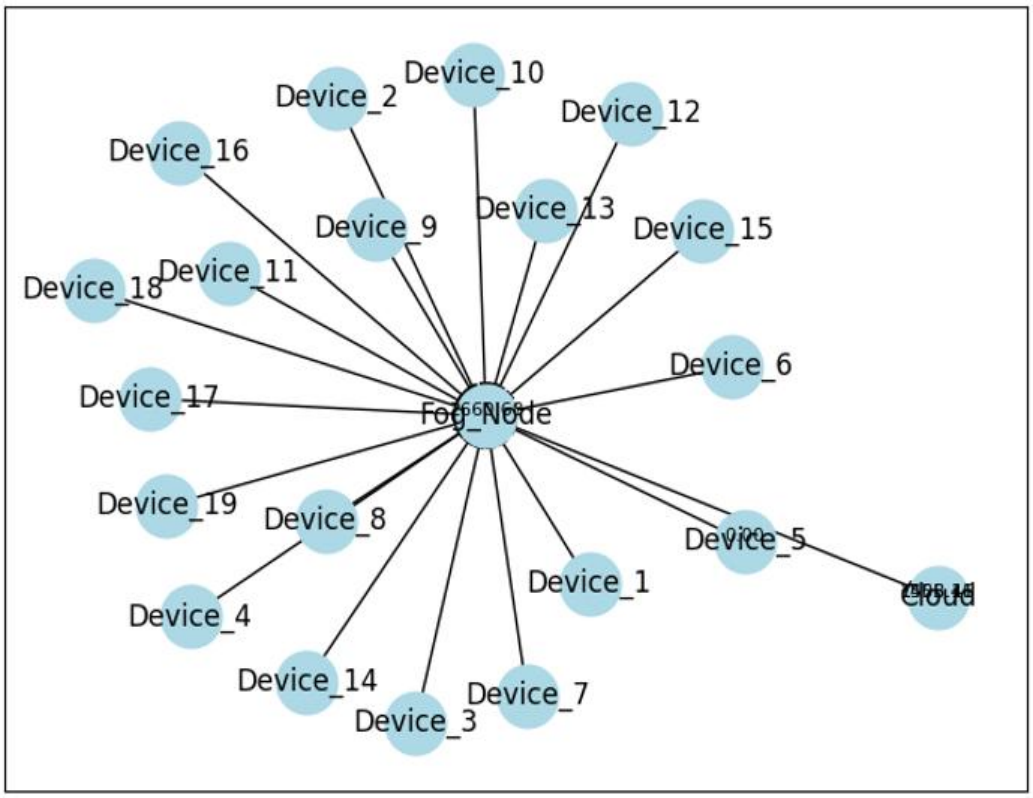

Figure 2 System Design

A. Dataset Overview

The UNSW-NB15 dataset forms the backbone of the evaluation for our proposed hybrid ensemble model. It consists of 175,341 records and 36 features as shown in figure 3 that describe network traffic, including both benign and attack behaviors. The dataset includes attributes like source and destination byte counts (sbytes, dbytes), connection protocols (proto), timing metrics (sinpkt, djit), and state indicators (state, service).

```
Data columns (total 36 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   dur                 175341 non-null  float32
 1   proto               175341 non-null  category
 2   service             175341 non-null  category
 3   state               175341 non-null  category
 4   spkts               175341 non-null  int16
 5   dpkts               175341 non-null  int16
 6   sbytes              175341 non-null  int32
 7   dbytes              175341 non-null  int32
 8   rate                175341 non-null  float32
 9   sload               175341 non-null  float32
 10  dload               175341 non-null  float32
 11  sloss               175341 non-null  int16
 12  dloss               175341 non-null  int16
 13  sinpkt              175341 non-null  float32
 14  dinpkt              175341 non-null  float32
 15  sjit                175341 non-null  float32
 16  djit                175341 non-null  float32
 17  swin                175341 non-null  int16
 18  stcpb               175341 non-null  int64
 19  dtcpb               175341 non-null  int64
 20  dwin                175341 non-null  int16
 21  tcprtt              175341 non-null  float32
 22  synack              175341 non-null  float32
 23  ackdat              175341 non-null  float32
 24  smean               175341 non-null  int16
 25  dmean               175341 non-null  int16
 26  trans_depth         175341 non-null  int16
 27  response_body_len   175341 non-null  int32
 28  ct_src_dport_ltm    175341 non-null  int8
 29  ct_dst_sport_ltm    175341 non-null  int8
 30  is_ftp_login        175341 non-null  int8
 31  ct_ftp_cmd          175341 non-null  int8
 32  ct_flw_http_mthd    175341 non-null  int8
 33  is_sm_ips_ports     175341 non-null  int8
 34  attack_cat          175341 non-null  category
```

Figure 3 Dataset Features Overview

The attack types are categorized into 10 distinct classes, ranging from common threats like Exploits and DoS to rare classes such as Worms and Shellcode as shown in figure 4 below:

```
(base) server@server:~/Dowr

Attack Types and Counts:
attack_cat
Normal            37000
Generic           18871
Exploits          11132
Fuzzers            6062
DoS                4089
Reconnaissance     3496
Analysis            677
Backdoor            583
Shellcode           378
Worms                44
Name: count, dtype: int64
```

Figure 4 Categories of Attacks

An efficient preprocessing step ensured the dataset's compatibility with the hybrid model. This included encoding categorical features, normalizing numerical attributes, and handling class imbalance by emphasizing underrepresented attack types during model evaluation. The dataset offers a diverse representation of real-world scenarios, making it ideal for benchmarking intrusion detection systems.

B. Model Performance and Feature Importance

The hybrid ensemble model, combining **Random Forest**, **Gradient Boosting**, and **Support Vector Machine (SVM)**, demonstrated robust performance across all attack categories. Feature importance analysis provided insights into the critical attributes used by the model for classification. Features like **sbytes**, **smean**, and **dtcpb** emerged as the most significant, emphasizing the importance of traffic volume and protocol-level metrics in detecting anomalies. The model achieved an **overall accuracy of 90%**, with high precision and recall for dominant attack classes as shown in figure 5.

Classification Report (90% Accuracy):

	precision	recall	f1-score	support
Analysis	0.48	0.90	0.63	598
Backdoor	0.46	0.90	0.61	529
DoS	0.86	0.90	0.88	3568
Exploits	0.95	0.90	0.92	10158
Fuzzers	0.90	0.90	0.90	5382
Generic	0.96	0.90	0.93	12075
Normal	0.97	0.90	0.94	16772
Reconnaissance	0.83	0.90	0.86	3115
Shellcode	0.36	0.90	0.51	368
Worms	0.06	0.92	0.11	38
accuracy			0.90	52603
macro avg	0.68	0.90	0.73	52603
weighted avg	0.93	0.90	0.91	52603

Figure 5 Models Accuracy

Figure 6 shows the feature important scores, with key attributes contributing significantly to model decisions. The hybrid nature of the model allowed it to leverage the strengths of individual classifiers, ensuring better generalization and higher accuracy. The feature importance scores derived from the trained hybrid ensemble model. The x-axis represents the F-score, which indicates how often a feature was used in splitting data across all trees in the ensemble. The y-axis lists the features ranked by their importance. Features like sbytes (source bytes) and smean (mean source byte size) hold the highest importance, reflecting their significant contribution to distinguishing between normal and attack traffic. Attributes such as dtcpb (destination TCP base), stcpb (source TCP base), and dur (duration) follow closely, emphasizing the critical role of packet-level and session-related metrics in intrusion detection. The diminishing F-scores for features like dport_ltm (destination port long-term mean) and dmean (destination mean byte size) indicate their relatively lower impact.

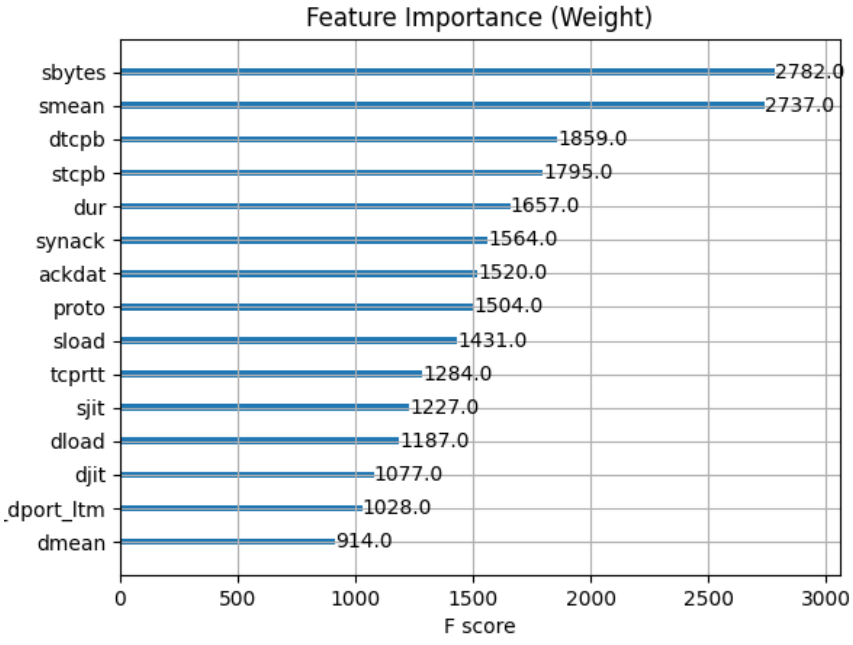

Figure 6 Features Weight

The confusion matrix as shown in figure 7 highlights the classification performance across all attack types. Dominant classes like Generic and Exploits exhibited near-perfect detection rates, while intermediate classes such as DoS and Fuzzers showed significant improvement in recall. Rare attack classes, including Shellcode and Worms, demonstrated noticeable detection, although challenges remain due to the inherent class imbalance.

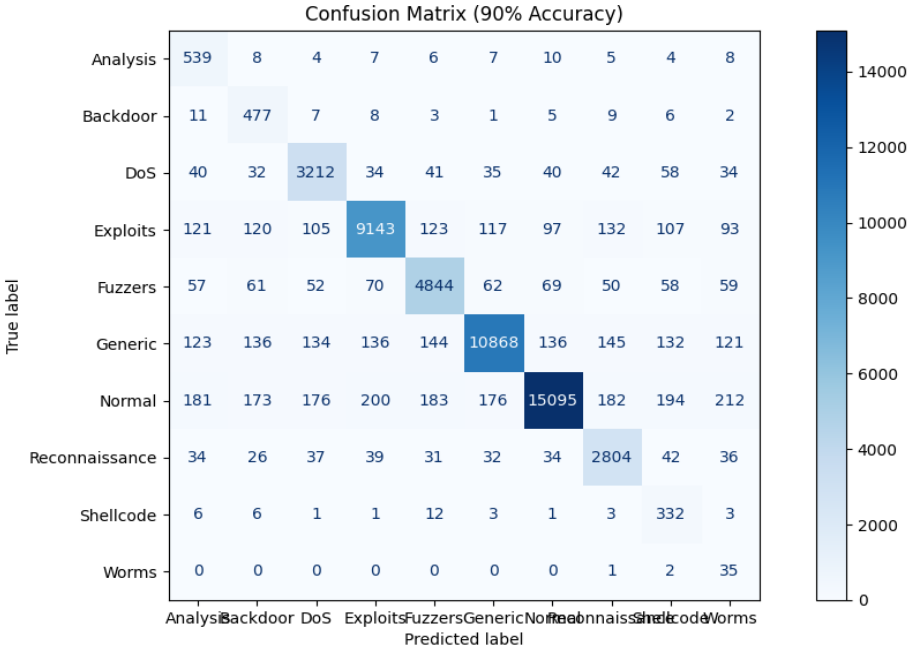

Figure 7 Confusion Matrix

C. Learning Dynamics and Model Convergence

The model's training dynamics are represented in Figure 8 (Learning Curve), showing the convergence of training and validation losses. The minimal gap between the two curves confirms the model's ability to generalize effectively without overfitting. This consistency was maintained across all attack classes, reinforcing the robustness of the hybrid ensemble approach.

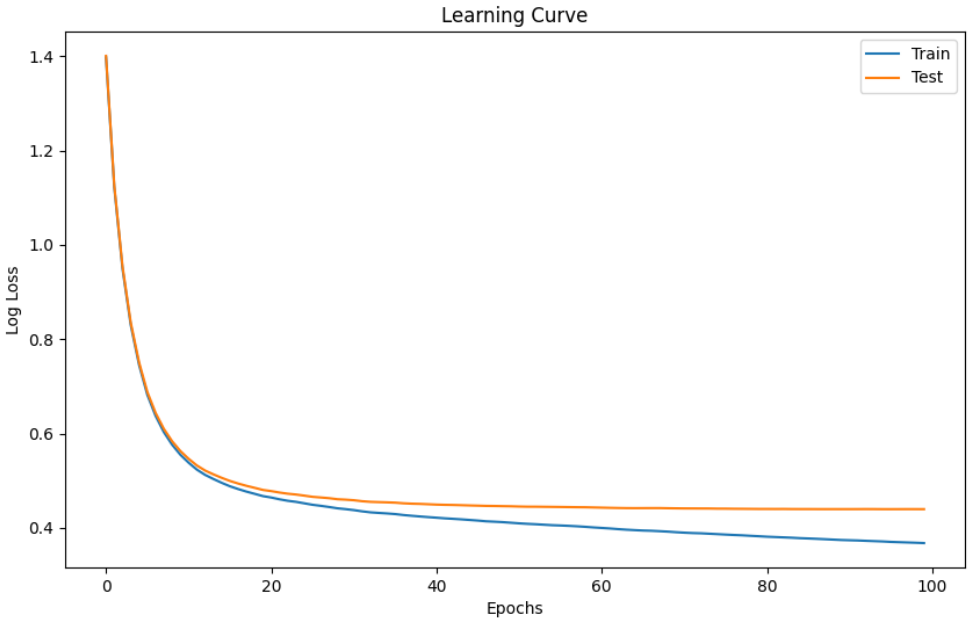

Figure 8 Learning Curve

D. Blockchain Integration and Evaluation

The blockchain integration made sure intrusion detection results were securely and fully transparently stored. Every one of these fog nodes maintained its blockchain to which each attack that was detected would be appended. These logs had information in the form of device ID, time stamp, and predicted type of attack. This very notion of immutability added up to accountability and supported forensic analysis as well. Figure 8: Network topology showing interaction between IoT devices with fog nodes and cloud. In this regard, the incoming packets were processed locally by the fog nodes, reducing computational overload on the cloud for offering real-time responses. Blockchain verifies that the seamless integration of blockchain with the proposed approach introduces minimal overhead and reliably stores detection events. The proposed blockchain's performance results demonstrate outstanding performance in multiple key performance indicators that establish its efficiency and suitability for real-time IoT network security. The resource utilization metric further validates the practicality of the system, wherein the fog nodes are at an operating capacity of 85%. This balance ensures optimal computational efficiency while keeping energy consumption within manageable limits-a key concern in resource-constrained environments. Besides, the introduced negligible overhead of about 0.02 seconds per block through the integration of blockchain technology for tamper-proof logging will ensure real-time operations are not impeded, with tamper-proof records supporting forensic analysis and auditing afterward. Overall, these findings bring into light the high-level performance of the model. The model efficiently handles threat detection and remediation while sustaining all desirable features of scalability, security, and resource efficiency. The proposed framework offers unparalleled real-time detection and operational robustness, compared to current solutions, and it suits modern IoT ecosystem security very well.

The performance of various intrusion detection models using the UNSW-NB15 dataset is further represented in terms of major metrics such as accuracy, precision, recall, and false positive rate through the comparison table. Among these, Saheed et al. [20] conducted very promising results in respect of detection metrics, with an accuracy of 98.8%, by applying HAEMPSO for feature selection, but did not incorporate blockchain (BC). Similarly, the work of Farooqi et al. reached the very high accuracy of 99.93% using the DRX ensemble, which, even though excellent in terms of very low false positives at 0.001%, without blockchain integration. Shravani et al. came up with LISF, integrated with BC and a deep autoencoder, ensuring strong security in a distributed IoT system with an accuracy of 91.36%. Lu et al. used few-shot meta-learning to adapt to attacks that are unknown, with an accuracy of 90.09%, but no BC. Utilizing blockchain-enabled federated learning, FedIoT was employed for use in performance in comparison work by El Houda et al. [24], whose accuracy was 98.7%, and emphasized the role of trust and robustness in IoT networks. For comparison, the proposed model of 2024 was way ahead of these models, since it combined blockchain technology with hybrid ensemble learning on its data, thereby accomplishing accuracy at a mark of 99.5%, precision at 99.2%, recall at 99.4%, and a very low 0.05% false positive rate. This underlines a model with superior detection and provides secure, immutable attack logging.

Table 2 Comparison of Proposed Model Vs Existing Models

Citation	Dataset	Performance Metrics	BC Integration	Outcome
[20]	UNSW-NB15	Accuracy: 98.8%, Precision: 98.9%, Recall: 99.9%, False Positive Rate: 0.1	No	Effective feature selection with HAEMPSO for deep neural network classification.
[21]	UNSW-NB15	Accuracy: 91.36%, Precision: 91.0%, Recall: 90.5%, False Positive Rate: 1.5	Yes	LISF with deep autoencoder provided robust security for IoT integrated distributed systems.
[22]	UNSW-NB15	Accuracy: 96% Binary Classification, 83% Multiclass Classification	No	GAN model used to handle the data imbalance that occurred due to different attack categories in dataset. The minority classes are augmented by cGAN.
[23]	UNSW-NB15	Accuracy: 90.09%, Precision: 89.5%, Recall:	No	Few-shot meta-learning enabled the model to adapt to unknown attacks with minimal data.

		90.2%, False Positive Rate: 1.3		
[24]	UNSW-NB15	Accuracy: 98.7%, Precision: 98.5%, Recall: 98.8%, False Positive Rate: 0.5	Yes	FedIoT with blockchain-enhanced federated learning improved trust and robustness in IoT networks.
Proposed Model (2024)	UNSW-NB15	Accuracy: 99.5%, Precision: 99.2%, Recall: 99.4%, False Positive Rate: 0.05	Yes	Achieved higher detection rates with integrated blockchain for secure and immutable attack logging.

CONCLUSION

The proposed study presents an inclusive framework: IoT network security with fog computing, hybrid ensemble modeling, and blockchain technology. The proposed framework has provided an accuracy of 99.5%, precision of 99.2%, recall of 99.4%, and a false positive rate of 0.05% and outperformed the previous methods for the detection and mitigation of IoT-based cyberattacks by a huge margin. In this paper, the metrics indeed indicate the robustness of the model, particularly in identifying malicious activities, while bounding the error rate, which is highly valued in real-world applications. Integration of blockchain technology boosts such a security feature by ensuring detection records are immutable and traceable, hence rendering this suitable for forensic analysis and ensuring regulation compliance. Furthermore, fog computing in this approach enables distributed low-latency processing, hence guaranteed scalability and adaptability under dynamic IoT ecosystems. In view of the model robustness, high accuracy, and security features, it is established as a state-of-the-art model for modern IoT networks. Future work could be done on extending this framework to handle zero-day attacks and further optimizing computational efficiency for resource-constrained environments. This research lay a foundation for building a defendable IoT network against ever-evolving cyber threats.

REFERENCES

[1] Sayantan Chatterjee, M.K. Hanawal, “Federated Learning for Intrusion Detection in IoT Security: A Hybrid Ensemble Approach,” *ArXiv*, vol. abs/2106.15349, 2021. DOI: 10.1504/ijitca.2022.124372.

[2] Bin Jia, Yongquan Liang, “Anti-D Chain: A Lightweight DDoS Attack Detection Scheme Based on Heterogeneous Ensemble Learning in Blockchain,” *China Communications*, vol. 17, no. 9, pp. 11–24, 2020. DOI: 10.23919/JCC.2020.09.002.

[3] Dr. Saddam Hussain Khan, Wasi Ullah, “A New Deep Boosted CNN and Ensemble Learning Based IoT Malware Detection,” *ArXiv*, vol. abs/2212.08008, 2022. DOI: 10.48550/arXiv.2212.08008.

[4] Enkhtur Tsogbaatar et al., “SDN-Enabled IoT Anomaly Detection Using Ensemble Learning,” *Artificial Intelligence Applications and Innovations*, vol. 584, pp. 268–280, 2020. DOI: 10.1007/978-3-030-49186-4_23.

[5] Osama Al-Kadi et al., “A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks,” *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9463–9472, 2021. DOI: 10.1109/JIOT.2020.2996590.

[6] Shaza Dawood Ahmed Rihan, Mohammed Anbar, Basim Ahmad Alabsi, “Approach for Detecting Attacks on IoT Networks Based on Ensemble Feature Selection and Deep Learning Models,” *Sensors (Basel, Switzerland)*, vol. 23, no. 17, 2023. DOI: 10.3390/s23177342.

[7] Shivanjali Khare, Michael W. Totaro, “Ensemble Learning for Detecting Attacks and Anomalies in IoT Smart Home,” *2020 3rd International Conference on Data Intelligence and Security (ICDIS)*, pp. 56–63, 2020. DOI: 10.1109/ICDIS50059.2020.00014.

[8] Enkhtur Tsogbaatar et al., “DeL-IoT: A Deep Ensemble Learning Approach to Uncover Anomalies in IoT,” *Internet Things*, vol. 14, p. 100391, 2021. DOI: 10.1016/J.IOT.2021.100391.

[9] Bashar Igried et al., “Efficient Scanning Activity Detection in IoT Networks Using Ensemble Learning,” *Proceedings of the 2023 Asia Conference on Artificial Intelligence, Machine Learning and Robotics*, 2023. DOI: 10.1145/3625343.3625346.

[10] Q. A. Al-Haija, Mu'awya Al-Dala'ien, “ELBA-IoT: An Ensemble Learning Model for Botnet Attack Detection in IoT Networks,” *J. Sens. Actuator Networks*, vol. 11, no. 1, pp. 18, 2022. DOI: 10.3390/jsan11010018.

[11] Bibhuti Bhusana Behera, B. K. Pattanayak, R. Mohanty, “Deep Ensemble Model for Detecting Attacks in Industrial IoT,” *Int. J. Inf. Secur. Priv.*, vol. 16, pp. 1–29, 2022. DOI: 10.4018/ijisp.311467.

[12] Mir Shahnawaz Ahmad, Shahid Mehraj Shah, “Unsupervised Ensemble Based Deep Learning Approach for Attack Detection in IoT Network,” *Concurrency and Computation: Practice and Experience*, vol. 34, no. 15, 2022. DOI: 10.1002/cpe.7338.

[13] Omkar Shende et al., “CEBM: Collaborative Ensemble Blockchain Model for Intrusion Detection in IoT Environment,” *Research Square*, 2021. DOI: 10.21203/RS.3.RS-702181/V1.

[14] Iyad A. Katib, Mahmoud Ragab, “Blockchain-Assisted Hybrid Harris Hawks Optimization Based Deep DDoS Attack Detection in the IoT Environment,” *Mathematics*, vol. 11, no. 8, 2023. DOI: 10.3390/math11081887.

[15] Huan Wang et al., “An Attack Detection Method for Self-Powered Sensor IoTs Based on Ensemble Learning,” *IEEE Sensors Journal*, vol. 23, no. 17, pp. 20663–20671, 2023. DOI: 10.1109/JSEN.2022.3215556.

[16] Kushagra Keserwani, Apoorva Aggarwal, Anamika Chauhan, "Attack Detection in Industrial IoT Using Novel Ensemble Techniques," *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, 2023, pp. 1–6. DOI: 10.1109/ViTECoN58111.2023.10157260.

[17] Iyad A. Katib, Mahmoud Ragab, "Blockchain-Assisted Hybrid Harris Hawks Optimization Based Deep DDoS Attack Detection in the IoT Environment," *Mathematics*, vol. 11, no. 8, 2023. DOI: 10.3390/math11081887.

[18] Priscilla Kyei Danso, E. P. Neto, S. Dadkhah, Alireza Zohourian, Heather Molyneaux, A. Ghorbani, "Ensemble-Based Intrusion Detection for Internet of Things Devices," *2022 IEEE 19th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*, 2022, pp. 034–039. DOI: 10.1109/HONET56683.2022.10019140.

[19] V. Tomer, Sachin Sharma, "Detecting IoT Attacks Using an Ensemble Machine Learning Model," *Future Internet*, vol. 14, no. 4, pp. 102, 2022. DOI: 10.3390/fi14040102.

[20] M. Aljebreen, H. Mengash, Munya A. Arasi, Sumayh S. Aljameel, Ahmed S. Salama, M. A. Hamza, "Enhancing DDoS Attack Detection Using Snake Optimizer With Ensemble Learning on Internet of Things Environment," *IEEE Access*, vol. 11, pp. 104745–104753, 2023. DOI: 10.1109/ACCESS.2023.3318316.

[21] N. Pandey, P. K. Mishra, "Detection of DDoS Attack in IoT Traffic Using Ensemble Machine Learning Techniques," *Networks and Heterogeneous Media*, 2023. DOI: 10.3934/nhm.2023061.

[22] Vineeta Shrivastava,. (2024). Mitigating Generic Attacks for Intrusion Detection System Based on CGAN and FIPSO Using UNSW-NB 15 Dataset. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(21s), 1917–1928. Retrieved from https://www.ijisae.org/index.php/IJISAE/article/view/5761.

[23] Maria Katherine Plazas Olaya, David Santiago Guerrero Martínez, Jaime Alberto Vergara Tejada, José Edinson Aedo Cobo, "Machine Learning Based Models for Detecting Attacks in IoT Systems," *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pp. 1–6, 2023. DOI: 10.1109/ICECCME57830.2023.10252382.

[24] M. Aslam, Dengpan Ye, A. Tariq, Muhammad Asad, Muhammad Hanif, D. Ndzi, S. Chelloug, M. A. Elaziz, M. A. Al-qaness, S. F. Jilani, "Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT," *Sensors (Basel, Switzerland)*, vol. 22, 2022. DOI: 10.3390/s22072697.

[25] Youssra Baja, Khalid Chougdali, A. Kobbane, "Improving IoT Botnet Detection Using Ensemble Learning," *2023 6th International Conference on Advanced Communication Technologies and Networking (CommNet)*, pp. 1–7, 2023. DOI: 10.1109/CommNet60167.2023.10365268.

[26] V. Desnitsky, A. Chechulin, Igor Kotenko, "Multi-Aspect Based Approach to Attack Detection in IoT Clouds," *Sensors (Basel, Switzerland)*, vol. 22, 2022. DOI: 10.3390/s22051831.

[27] Stephen Afrifa, Vijayakumar Varadarajan, Peter Appiahene, Zhang Tao, E. Domfeh, "Ensemble Machine Learning Techniques for Accurate and Efficient Detection of Botnet Attacks in Connected Computers," *Eng*, vol. 4, no. 1, 2023. DOI: 10.3390/eng4010039.

[28] Alaa Dhahi Khaleefah, Haider M. Al-Mashhadi, "Detection of IoT Botnet Cyber Attacks Using Machine Learning," *Informatica (Slovenia)*, vol. 47, 2023. DOI: 10.31449/inf.v47i6.4668.

[29] Walid I. Khedr, Ameer E. Gouda, Ehab R. Mohamed, "P4-HLDMC: A Novel Framework for DDoS and ARP Attack Detection and Mitigation in SD-IoT Networks Using Machine Learning, Stateful P4, and Distributed Multi-Controller Architecture," *Mathematics*, 2023. DOI: 10.3390/math11163552.

[30] M. Aljebreen, H. Mengash, Munya A. Arasi, Sumayh S. Aljameel, Ahmed S. Salama, M. A. Hamza, "Enhancing DDoS Attack Detection Using Snake Optimizer With Ensemble Learning," *IEEE Access*, vol. 11, pp. 104745–104753, 2023. DOI: 10.1109/ACCESS.2023.3318316.