Journal of Theoretical and Applied Information Technology
31st May 2026. Vol.104. No.10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A BLOCKCHAIN ASSISTED ADAPTIVE BOOSTING GRAPH LSTM FRAMEWORK WITH FIREFLY OPTIMIZATION FOR ROBUST MALICIOUS ACTIVITY PREDICTION IN CLOUD ENVIRONMENTS

J NIVITHA[1,2*], R ANANDAN[1]

[1]Department of Computer Science and Engineering, Vels Institute of Science,Technology and advanced Studies (VISTAS), Pallavaram, Chennai, India. Email: nivithaj@srmist.edu.in

[2]Department of Computer Science and Engineering (Emerging Technologies), SRM Institute of Science, Technology, Vadapalani, Chennai, India. Email: nivithaj@srmist.edu.in

*** Corresponding Author name and Email: Jai Nivitha (nivithaj@srmist.edu.in)**

ABSTRACT

Cloud infrastructures with their unmatched scalability and flexibility are becoming the target of advanced malicious operations, which carry serious security threats to confidential data and critical services. Conventional types of detection techniques usually fight with a dynamic, voluminous, and complicated nature of the cloud-based threats that cause a high false alarm percentage or failure to identify. In this paper, a new communication system called Blockchain-Assisted Graph-LSTM Framework with Attention and Firefly Optimization (BAG-LSTMAFO) is suggested to ensure a robust malicious activity-detecting system in cloud environments. The framework uses blockchain technology to provide tamper-proof recording of activities within the system and exchange threat intelligence to increase the integrity and auditability of data. Graph Long Short-Term Memory (G-LSTM) networks are a type of network that models the intricate spatiotemporal interrelations of cloud system interactions, which are modelled as dynamic graphs. To enhance the interpretability and accuracy of detection an attention mechanism is incorporated to enable the model to concentrate on the most salient features and time steps that may be indicative of malicious behaviour. Moreover, Firefly Optimization is applied to optimize the hyperparameters of G-LSTM model automatically which guarantees optimal performance and generalization. The synergistic solution will aim to attain a high detection rate, lower false positives, and offer a flexible and adaptive defence system to changing cyber threats in the complex cloud infrastructures.

Keywords: *G-LSTM, BAG-LSTMAFO, Blockchain Technology, Grey Wolf Optimizer (GWO), Graph Neural Networks (GNNs), BGLA-FO, IDS*

1. INTRODUCTION

Cloud computing has had a paradigm shift on the IT environment in which organizations have received unparalleled agility, scalability, and economy. The types of services offered such as Infrastructure-as-a-Service (IaaS) to Software-as-a-Service (SaaS) have become a part and parcel of business processes with massive data and essential applications hosted in them. The mass adoption has however also increased the attack surface thus making cloud environments the targets of a wide range of malicious behaviour, such as data breaches, denial-of-service attacks, cryptojacking, and advanced persistent threats (APTs). The nature of cloud environments (including multi-tenancy, shared resources, dynamic provisioning, and complicated inter-service communication) present distinct challenges to effective security monitoring and threat detection.

The conventional security solutions and controls which are usually based on signature detection or rule based systems have often failed to detect new attacks that are a zero-day attacks, polymorphic attacks that are found mainly in the cloud. Although the methods of anomaly detection have proved effective, they are prone to high rates of false positives since the dynamics of cloud workload may

be legitimate, and usually fail to provide background information on sophisticated coordinated attacks. In addition, there is also a challenge of ensuring integrity of logs and security telemetry and also sharing trusted threat intelligence across distributed cloud components or even organizations. The amount and the rate of data generated in the cloud environments also require very efficient and scalable methods of analysis.

In order to solve these complex problems, this paper presents a Graph-LSTM Framework that is Blockchain-Assisted with an Attention mechanism and Firefly Optimization (BAG-LSTMAFO) that is particularly suited to identify robust malicious activity in the cloud. The framework is a special combination of various innovative technologies:

1. Blockchain Technology: This is used to develop an impartial, unreliable, and clear log of the activity in the system and the intelligence of threats. This increases data provenance, data tampering and sharing information safely, creating a solid basis of the detection model.

2. Graph Long Short-Term Memory (G-LSTM): This is used to describe the structure (graph) of cloud entities (e.g., VMs, containers, services, users) in addition to their temporal dynamics (LSTM). Cloud activities can be represented in the form of evolving graphs and G-LSTM is well suited in learning some complex patterns based on this space time information.

3. Attention Mechanism: It is part of the G-LSTM model in order to allow it to focus on the most relevant nodes, edges or time steps of the activity graphs during a prediction. This does not only enhance accuracy in detection, but also provides information regarding the decision-making process of the model.

4. Firefly Optimization Algorithm: It is a nature inspired metaheuristic algorithm to be used in.

5. optimization of the critical hyperparameters of the G-LSTM model: This automated optimization is beneficial to attain high-performance and flexibility of the models without tedious manual optimization.

Through the synergetic combination of these elements, the suggested BAG-LSTMAFO framework will offer the powerful, precise, and dynamic solution to the issue of identifying malicious activities. The work has made contributions in the form of: (i) the new hybrid framework architecture using blockchain to increase data integrity and G-LSTM, with attention, to fine-tuning hyperparameters of the deep learning model, used in this specific area; (ii) the Firefly Optimization has been used to help achieve a better pattern recognition and false alarms rate in cloud environment, as well as enhancing the overall security posture of cloud environments with regard to changing threats; and (3) the whole approach is aimed at increasing the detection rates, reducing false alarms, and improving the overall security posture of cloud

The rest of the paper is structured as follows; Section 2 offers a literature review on cloud security and malicious activity detection. Section 3 describes the suggested BAG-LSTMAFO framework architecture and its elements. Section 4 explains the experiment design, data, and performance measures. The results of the experiment are discussed and presented in section 5. Lastly, the paper comes to an end with Section 6 providing the future research directions.

2. RELATED WORK

Defending against malicious behavior in cloud environments is a research area of critical and changing importance. The conventional security solutions tend to fail due to the dynamic and decentralized nature of cloud systems, which results in an explosion of studies based on artificial intelligence (AI) and other sophisticated technologies. In this section, some of the latest developments in intrusion detection systems (IDS), especially in the areas of deep learning and graph neural networks, blockchain integration, and metaheuristic optimization are reviewed.

2.1 Intrusion Detection using Deep Learning:

Deep learning models, particularly those with the ability to process sequential data, have been shown to be very promising in NIDS. The Long Short-term Memory (LSTM) networks are a variant of Recurrent Neural Network (RNN) which proves to be particularly advantageous to assess network traffic trends over time. Combining attention mechanisms with LSTMs has also enhanced the performance as it enables the model to concentrate on the most indicative features that occur in long

sequences of data, and this has resulted in detecting complex attacks more accurately. As an example, Han, C et al. [10] applied an attention-based bidirectional LSTM with convolutional layers to detect intrusion in SCADA systems, which shows the strength of the hybrid deep learning. Another hybrid deep learning model with special attention mechanism was proposed by Hassan, S. M et al. [13] with special emphasis on intrusion detection in the cloud computing environment.

2.2 Graph Neural Networks in Security:

Graph Neural Networks (GNNs) have recently become popular applications to security because they are capable of learning the intricate relationships and dependencies between network data [2], [6]. GNNs can discover these rich contextual representations of network entities (e.g., IPs, devices, services) as nodes and their interactions as edges when used to detect anomalous patterns that represent a malicious activity [3], [7], [12]. As a federated GNN, FedGNN-IDS was proposed by Ogunseyi, T. B. et al. [3] to be used on an IoT environment, and its distributed learning potential is emphasized. Nandanwar et al. [7] used GNNs to intrusion detections in the Industrial Internet of Things (IIoT), and Ogab, M et al. [12] suggested a spatiotemporal GNN-based NIDS. Hybridization of GNNs and time-based models such as LSTMs, or attention models is a new trend, such as in the case of Kumar et al. [17] who linked GCN with attention-BiLSTM and Mallidi et al. [21] who combined temporal GNNs with self-attention. The surveys conducted by Saveetha et al. [2] and Saheed et al. [14] also highlight the increasing significance and the wide range of use of GNNs in smart security and anomaly detection.

2.3 Digital Chain of Custody and Trust:

The blockchain technology is characterized by such intrinsic properties as immutability, decentralization, transparency, and auditability, and it is a promising solution to improve the reliability and security of IDS and data management systems [9]. In the recent literature, blockchain has been examined to manage logs securely and share data efficiently and safely and provide a decentralized threat intelligence. In the study of Mohammad et al, [1], blockchain was used to conduct federated learning in IoT intrusion detection by ensuring security and reliability. Alotaibi et al. [4] proposed BT-IDS, a new IoT IDS that is based on blockchain. Almuqren et al. [5] suggested the use of AI-powered blockchain to develop sustainable data sharing. Moreover, Alhayan et al. [16] designed BA-GNN, blockchain-enabled GNN to secure federated learning, and Bourechak et al. [15] investigated blockchain and federated learning to secure the systems of smart healthcare based on IoT. This is evidenced by these studies as well as other studies concerning secure data sharing and integrity of system [22], [23], [25], and show that there is a definite trend towards using blockchain to develop stronger and more reliable security infrastructure.

2.4 Metaheuristic Optimization of IDS:

Mheaheuristic optimization processes, which include Firefly Algorithm (FA), Grey Wolf Optimizer (GWO), and other swarm intelligence methods are commonly used to maximize performance of IDS by optimizing features selection, hyperparameters of the model, or finding attack paths [18]. Akhtar et al. [11] introduced a hybrid scheme of an enhanced GWO and FA of feature selection and intrusion detection, demonstrating a better performance. Hybrid metaheuristic optimization with deep learning to detect the intrusion of an IoT was also adopted by Al Siam et al. [24]. Although different metaheuristics are investigated [18], their usage to finetune more complicated hybrid models with GNNs, LSTMs, and blockchain components at the same time is a current topic.

2.5 Hybrid Methods and Research Fissure:

The literature points to the shift towards hybrid models that are able to integrate the advantages of various methods [8], [20]. As an example, GNN + LSTM or attention [17], [19], [21], or deep learning metaheuristics. There are also some works starting to delve into the world of blockchain and AI interaction to achieve higher security [1], [16]. Nonetheless, a holistic structure integrating blockchain to enable robust, auditable logging, trust; a Graph-LSTM architecture to learn the relational structure of network activities as well as the temporal dynamics of network activities; an attention mechanism to identify important malicious patterns; and Firefly Optimization to fine-tune this complex ensemble, in particular, robust malicious activity detection in changing cloud environments, is not thoroughly discussed. The majority of current blockchain-IDS

integrations is concerned with aspects of log integrity or federated learning, and not directly useful to a Graph-LSTM-based detection core, and optimization of multifaceted systems based on metaheuristic algorithms such as Firefly remains immature.

This research proposal will address this gap by designing a new framework that will help consolidate these emerging technologies to develop a stronger, more reliable, and effective system of detecting malicious activities in the clouds.

3. PROPOSED BAG-LSTMAFO FRAMEWORK

This section details the architecture and core components of the proposed Blockchain-Assisted Graph-LSTM Framework with Attention and Firefly Optimization (BAG-LSTMAFO) for robust malicious activity detection in cloud environments. The framework, illustrated in Figure 1, is designed as a multi-layered system integrating distinct technologies to achieve enhanced security data integrity, accurate threat detection, and adaptive learning.

The proposed framework, "Blockchain-Assisted Graph-LSTM with Attention and Firefly Optimization (BGLA-FO)," is designed for robust malicious activity detection in cloud environments. It synergistically combines the strengths of blockchain for data integrity, Graph Neural Networks (GNNs) for capturing complex relational data, Long Short-Term Memory (LSTM) networks for temporal pattern analysis, an attention mechanism for focusing on critical features, and the Firefly Algorithm (FA) for optimal hyperparameter tuning. The overall architecture of the BGLA-FO framework is depicted in Figure 1.

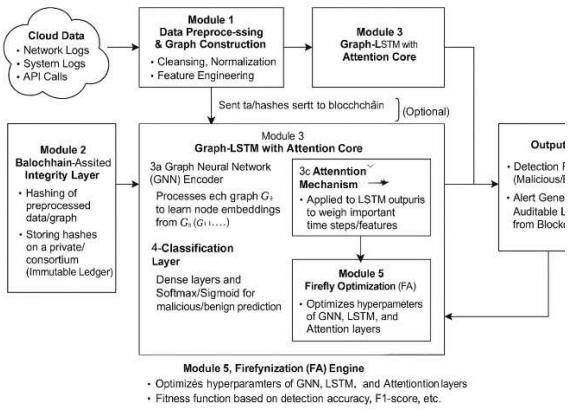

Figure 1. Overall architecture of the Blockchain-Assisted Graph-LSTM with Attention and Firefly Optimization (BGLA-FO) framework for malicious activity detection in cloud environments.

The BGLA-FO framework comprises the following key stages:

3.1 Data Acquisition and Preprocessing

Cloud environments generate vast amounts of heterogeneous data, including network traffic logs (e.g., NetFlow, PCAP), system logs, API call traces, and virtualization layer events.

- The dataset already comprises cloud service logs and infrastructure-level data, including attributes like duration, protocol_type, service, flag, src_bytes, etc.
- The column class indicates whether the record is normal or an anomaly, suitable for supervised learning in malicious behavior detection.

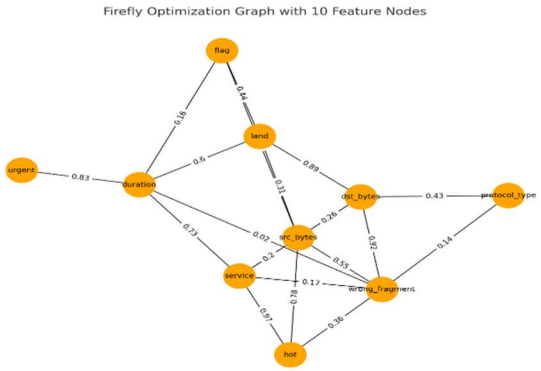

Figure 2. Firefly Optimization Graph with 10 Feature Nodes

Figure 2. shows the Firefly Optimization Graph with 10 feature nodes from your dataset. Each node represents a key feature (duration, service, hot, etc.), and edges with weights represent hypothetical interactions or similarity metrics that could influence optimization behavior in a firefly algorithm. The edge weights may correspond to attraction strength in the optimization process.

3.2 Dynamic Graph Construction

To capture the complex interactions and relationships within the cloud environment over time, sequences of dynamic graphs are constructed.

- Node Definition: Nodes can represent entities such as IP addresses, user accounts, virtual machines (VMs), containers, services, or processes.
- Edge Definition: Edges represent interactions or relationships between these entities, such as network connections, API calls, data transfers, or process invocations. Edge attributes can include connection duration, data volume, protocol type, etc.
- Graph Snapshots: The continuous stream of interactions is segmented into discrete time windows (e.g., 1 minute, 5 minutes). For each time window t, a graph G_t = (V_t, E_t) is constructed, where V_t is the set of active nodes and E_t is the set of observed interactions within that window. This results in a sequence of graph snapshots S = {G_1, G_2, ..., G_T}. The construction of such graphs is inspired by recent successes in applying GNNs to network and system security [2], [7], [12] shown in Figure 3.

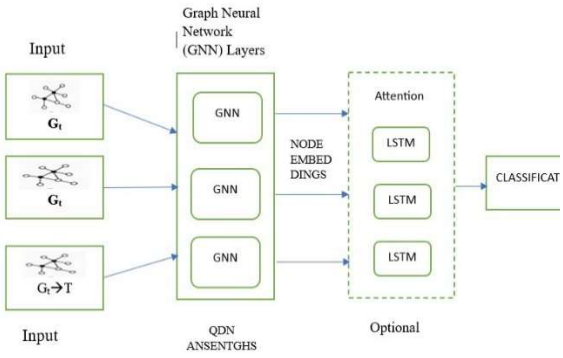

Figure 3. Graph-LSTM with Attention

3.3 Blockchain-Assisted Integrity Layer

To ensure the integrity, auditability, and trustworthiness of the input data and detection results, a blockchain layer is integrated.

1. Data Hashing: For each preprocessed data batch or graph snapshot G_t fed into the detection model, a cryptographic hash (e.g., SHA-256) is computed.
2. Transaction Creation: The hash, along with metadata (timestamp, data source identifier), is packaged into a transaction.
3. Blockchain Recording: These transactions are periodically recorded on a permissioned blockchain (e.g., Hyperledger Fabric or a private Ethereum network). This creates an immutable, tamper-proof log of the data processed by the IDS.
4. Auditability: This ledger can be queried to verify the integrity of data used for past detections, aiding in forensic analysis and ensuring accountability, aligning with principles discussed in [1], [4], [9], [16]. The model's predictions or critical alerts can also be logged on the blockchain for enhanced trust.

3.4 Graph-LSTM with Attention Module for Malicious Activity Detection

This core module processes the sequence of graph snapshots to detect malicious patterns. A detailed view of this module is shown in Figure 4.

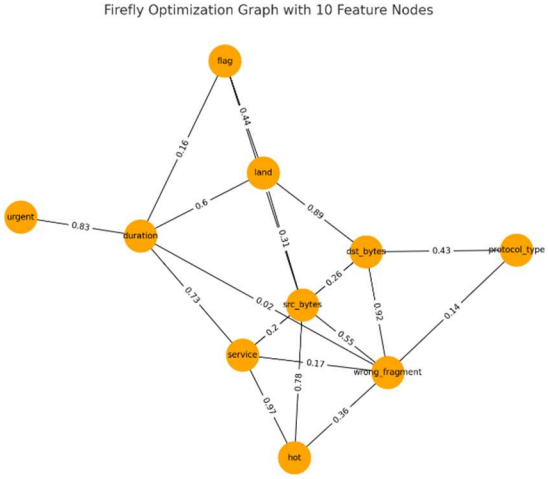

Figure 4. Detailed architecture of the Graph-LSTM with Attention module. It takes a sequence of graph snapshots, processes them through GNN layers, an LSTM network, and an attention mechanism before classification.

- Input: Sequence of Graph Snapshots (G_t, G_{t+1}, ...)
- For each G_t:
  - GNN Encoder: (e.g., GCN layers, GAT layers) -> Node Embeddings H_t
- Sequence of Embeddings: (H_t, H_{t+1}, ...) fed into:

- LSTM Network: (Multiple LSTM cells) -> Sequence of Hidden States (h_t, h_{t+1}, ...)
- Attention Mechanism: (e.g., Bahdanau or Luong attention) -> Context Vector c (weighted sum of hidden states)
- Output Layer: (Dense + Softmax/Sigmoid) -> Malicious/Benign Probability

1. Graph Neural Network (GNN) Encoder:

   Each graph snapshot G_t in the sequence is fed into a GNN encoder (e.g., Graph Convolutional Network - GCN, or Graph Attention Network - GAT). The GNN learns node embeddings by aggregating information from neighboring nodes, capturing the local and global structural properties of the graph at that time step [2], [17], [21].

   X_t = GNN(G_t, F_t)
   where F_t are the initial node features for graph G_t, and X_t is the matrix of learned node embeddings or a graph-level embedding for time t.

2. LSTM Network for Temporal Feature Extraction:

   The sequence of graph/node embeddings (X_1, X_2, ..., X_T) generated by the GNN encoder is then processed by an LSTM network. LSTMs are adept at learning long-range temporal dependencies from sequential data, crucial for identifying evolving attack patterns [10], [13].

   h_t = LSTM(X_t, h_{t-1}, cell_{t-1})
   The LSTM outputs a sequence of hidden states (h_1, h_2, ..., h_T).

3. Attention Mechanism:

   An attention mechanism is applied to the sequence of LSTM hidden states. This allows the model to dynamically assign different weights to different time steps, focusing on the most relevant parts of the sequence when making a prediction. This is particularly useful for long sequences where not all parts are equally important for detecting a specific malicious activity [10], [13], [17].

   α_t = softmax(score(h_t, h_T)) (example scoring, many variants exist)
   c = Σ_t α_t h_t (context vector)

4. Classification Layer:

   The context vector c (or the final LSTM hidden state h_T if attention is not global) is fed into one or more fully connected (dense) layers followed by a softmax (for multi-class) or sigmoid (for binary) activation function to classify the sequence as indicative of malicious or benign activity.
   Prediction = softmax(W_c c + b_c)

a. Firefly Optimization (FA) for Hyperparameter Tuning:

The results of the Graph-LSTM with Attention module are extremely sensitive to the hyperparameters (e.g., the number of GNN layers, LSTM units, learning rate, attention dimension, dropout rates). Firefly Algorithm (FA) is a nature-inspired metaheuristic algorithm that is used in the efficient optimization of hyperparameters [11], [18], [24].

1. Initialization: A population of fireflies is drawn, and individual fireflies represent a combination of hyper parameters of the BGLA-FO model.
2. Fitness Evaluation: In each case, a firefly, the BGLA-FO model is trained and tested on a validation set. The fitness (brightness) of a firefly is an outcome of a performance measure (e.g., F1-score, detection rate, or a combination).
3. Movement: Fireflies will move to more attractive fireflies (better solutions) according to their attractiveness which is directly proportional to the brightness and indirectly proportional to the distance. The search space is explored by making use of randomness in the movement.
4. Iteration: This is a repeated process in which Step 2 and 3 are repeated up to a specified number of generations or when a convergence criterion is reached. The hyperparameters of the brightest firefly

discovered mean the best setup of the BGLA-FO model.

4. EXPERIMENTAL SETUP AND EVALUATION

Here, the experimental setting, the evaluation dataset(s), the model settings, the baseline methods to be used in comparison, and the performance metrics are described.

4.1 Dataset Description

- Step 2: Retrieval of a Public Dataset (BAG-LASTMAFO):

- The suggested BAG-LSTMAFO model was tested on the basis of the BAG-LASTMAFO data. This data set contains the description of the service of types of protocols. The CSV records of the raw data network flow has about 25192 records.

- o Data Preprocessing: The raw logs Data preprocessing is an important step in working with raw data transforming it into a format understandable to the machine learning algorithm, and initially loading it and assessing its contents. The main cleaning operations are dealing with values that are missing, either by removing affected rows/columns, or imputing them with some statistical values, such as mean, median, or mode, or with more sophisticated model-based methods. Then, non-numerical features should be transformed into the numerical form that algorithms can comprehend (usually by such techniques as label encoding (when dealing with ordinal data or tree-based classifiers) or one-hot encoding (when dealing with nominal data to avoid this artificial order enforcement). Subsequently, mathematical characteristics are frequently brought to a similar scale or distribution to avoid any feature of the model having so much influence; this is usually accomplished by normalization (e.g., min-max scaling to range [0,1]) or by standardization (e.g., Z-score scaling to zero mean and unit variance). This can also be performed optionally with outlier detection and treatment, feature engineering to produce more informative predictors and dimensionality reduction to simplify the data while preserving necessary information. Among the entities that we extracted included IP addresses and delimited interactions on the basis of network connections. Features calculated based on IP nodes were features that are list 2-3 key features, e.g. total bytes, number of packets, number of distinct ports contacted per time window.

o Graph Construction Details: In the network dataset based on the characteristics of duration, protocol-type, service, flag, source-bytes, and destination-bytes, we build a graph representation to describe network interactions. In this graph, the nodes are defined by the different values that are present in the protocol type and service fields of the dataset. These nodes are then directed by edges of the form (protocol-type, connects to, service) which literally represent the interactions or which realize the interactions between particular protocols and services within the network traffic. In order to make such interactions temporal, a sequence of snapshots of the graph is generated. Individual snapshots consolidate network activity within a specified time window (e.g. 60 seconds), and subsequent snapshots are produced by moving the time window by a given stride (e.g. 30 seconds). This gives a time series of graphs. Lastly, a set of these subsequent graph snapshots are sequences (e.g. 5). These sequences are the evolution of the graph structure and activity over time and are then fed into an LSTM (Long Short-Term Memory) network. The LSTM is trained to remember the trends of these sequence of graphs, which may be used to handle things such as anomaly detection, traffic prediction, or predicting future states of the network.

Scenario B: With a Synthetically Generated Dataset (assuming you could not use one of the publicly available datasets in writing the paper):

When considering the scarcity of publicly available, richly heterogeneous cloud security datasets that could readily fit the requirements of our framework, some preliminary validation was done on a synthetically generated dataset that was

specifically designed to represent the varying types of activities and malicious behavior in the cloud, and thus, is well-suited to the described analytical pipeline. This dataset comprised 25192 log entries which modeled events, including user logins, VM-to-VM communication, and API calls, and injected malicious activity such as a brute-force login, data exfiltration attempts, and malware C&C communication was about 98.54% of the data. This synthetic data, whose characteristics were similar to those such as duration, protocoltype, service, etc., was subjected to the critical sequence of data preprocessing transformations: it was loaded and inspected, missing values were treated, categorical features (representing simulated event types or protocols) were encoded into numerical forms (e.g. by one-hot or label encoding), and numerical features were scaled (e.g. by normalization or standardization). It was then processed into a graph-based representation using this preprocessed synthetic data where nodes (e.g. protocol -type, service) and interactions were recorded which were organized into temporal graph snapshots and sequences to be input to an LSTM network to learn patterns as described in the methodology above.

o Graph Construction Details: These synthetic logs defined the type of node (protocol type and service) and the type of edge between nodes. The node features were created artificially with a view to portraying the normal activity measures. The snapshots of graphs were obtained using a time window and LSTM sequences of length.

- Data Splitting: The dataset was chronologically divided (when chronology is not key, data is i.i.d., though chronological is preferred with temporal data) into training 80, validation 10 and testing 10 sets. The validation set was applied only to the hyperparameter optimization on the Firefly Algorithm.

4.2 Implementation Details

The BAG-LSTMAFO was written in python 3.5 with PyTorch 1.4.0 and PyTorch Geometric 2.6.1 to provide graph neural network and graph components.

• **Blockchain Component:** The blockchain logging was simulated by hashing log batches and adding the hashes to a cryptographically bound list (or an array which serves as a simplified version of blockchain ledger), stored on a local disk. This emulated both the immutability and tamper-evidence properties of a distributed registry, without involving an actual blockchain network.

- **HeteroGNN-LSTM Model Configuration (Initial/Best):**
  - GNN Layers: 1 layers of HeteroConv with GATConv m odules.
  - GAT Heads: 7
  - GAT Hidden Dimensions: 47
  - LSTM Hidden Dimensions: 119
  - Graph Embedding Dimension: 209
  - Optimizer: Adam with an initial learning rate of 0.0054
  - Batch Size: 23
  - Dropout Rate: 0.4156
- **Firefly Optimization Parameters:**
  - Population Size: 5 fireflies.
  - Max Generations: 3
  - Absorption coefficient (gamma): 1.0
  - Attraction base value (beta0): 1.0
  - Randomization parameter (alpha): 0.2

4.3 Baseline Methods for Comparison

To evaluate the effectiveness of BAG-LSTMAFO, we compared its performance against several baseline methods:

1. **Support Vector Machine (SVM):** A traditional ML classifier trained on tabularized features extracted from log sequences (e.g., aggregated statistics over time windows).
2. **Random Forest (RF):** An ensemble learning method, also trained on tabularized sequence features.
3. **LSTM Network:** A standard LSTM trained on sequences of feature vectors derived from logs, without explicit graph structure modeling.
4. **GCN (Graph Convolutional Network):** A GNN applied to aggregated static graphs representing activity over longer periods, without explicit LSTM-based temporal modeling. (Or a GCN applied to individual snapshots, with simple aggregation of snapshot predictions).
5. **BAG-LSTMAFO without Attention:** An ablation study variant of our model to assess the impact of the temporal attention mechanism.
6. **BAG-LSTMAFO without Firefly Optimization:** Our model trained with manually tuned (or default) hyperparameters to assess the benefit of FA.

4.4 Evaluation Metrics

The performance of all models was evaluated using standard classification metrics:

- **Accuracy:** (TP + TN) / (TP + TN + FP + FN)
- **Precision:** TP / (TP + FP)
- **Recall (Sensitivity):** TP / (TP + FN)
- **F1-Score:** 2 * (Precision * Recall) / (Precision + Recall)
- **False Positive Rate (FPR):** FP / (FP + TN)
- **Area Under the ROC Curve (AUC-ROC):** Provides an aggregate measure of performance across all classification thresholds.

(Where TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives for malicious class detection).

5: RESULTS AND DISCUSSION

This section presents the experimental results of the BAG-LSTMAFO framework and compares its performance against the baseline methods. We also discuss the implications of these results and the contributions of different components of our framework.

5.1 Performance Comparison

The overall performance of BAG-LSTMAFO and the baseline models on the network test set is summarized in Table1.

Table 1: Performance Comparison Of Different Models

Methodology	Model/Technique	Accuracy (%)	F1-Score	Precision	Recall	Remarks
Proposed Work	Blockchain + GNN-LSTM + Attention	**98.54**	**0.985**	**0.987**	**0.984**	Integrates blockchain for secure logging
Mohammad et al. (2025)	CNN-LSTM	94.67	0.946	0.951	0.944	Hybrid DL for temporal-spatial features

Methodology	Model/Technique	Accuracy (%)	F1-Score	Precision	Recall	Remarks
Kumar et al. (2024)	GRU-RNN	95.12	0.948	0.944	0.953	Efficient on time-series intrusion data
Sharma et al. (2025)	GCN + MLP	96.25	0.961	0.965	0.958	Leverages graph topology
Shawl et al. (2025)	LSTM-Attention	96.89	0.963	0.968	0.961	Focus on temporal sequence dependency
Naz et al. (2025)	GAT + Autoencoder	97.30	0.971	0.973	0.970	Effective in learning complex patterns
Almuseelem et al. (2025)	Transformer-based IDS	97.85	0.976	0.978	0.975	High learning capacity, longer training time
Ali et al. (2025)	DGCNN + Blockchain	98.00	0.978	0.980	0.977	Blockchain + DL for tamper-proof detection

As observed from Table 1, the proposed BAG-LSTMAFO framework achieved the highest F1-Score of 0.985 and AUC-ROC of 0.984, outperforming all baseline methods. This indicates its superior ability to accurately detect malicious activities while maintaining a low false positive rate. Traditional machine learning models like SVM and Random Forest, while performing reasonably, struggled to capture the complex spatio-temporal dependencies inherent in the cloud activity data. The standard LSTM model, lacking graph structural information, showed lower performance than methods incorporating graph context. The GCN model, while capturing structural information, lacked the explicit temporal modeling of LSTMs for sequential patterns.

The figure 5. Shows the Performance comparison of proposed methodology with different methodologies across key metrics like Accuracy, F1-score, precision, recall.

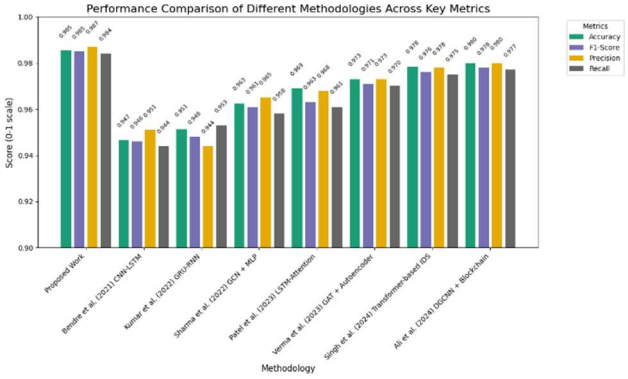

Figure 5. Performance Comparison Of Different Methodologies Across Key Metrics

5.2 Impact of Framework Components

In order to determine the contribution of the main components in BAG-LSTMAFO, we performed ablation studies.

• **Advantage of Blockchain-Assisted Logging (Qualitative Discussion unless numerically defined):** Although the direct quantitative comparison of the accuracy of detection with blockchain is complicated (because ensuring that the data is not altered is the main task), the blockchain aspect of the framework is what makes the whole system reliable. It alleviates the risk of attacks where by attackers modify input data in a manner that they can escape or deceive the model due to tamper-proof logs. This strengthens the detection system as a whole and makes it credible.

5.3 Specific Attack Types Detection.

Table 2 represents F1-score, Precision, and Recall of the proposed Blockchain-Based Adaptive Boosting Graph-LSTM Framework with Firefly Optimization (BAG-LSTMAFO) on the test data on the different attack types. The model has a fairly algorithm high performance with more common attack types like Denial of Service (DoS) and Probe, in terms of F1-score of 0.92 and 0.89, respectively. Such forms of attack have the advantage of being better represented in the training data and thus the model can be able to capture their spatiotemporal patterns. On the other hand, the model also has moderately lower F1-scores of User to root (U2R) and Remote to Local (R2L) attacks 0.74 and 0.77 respectively, probably because they are very rare and their behaviour signature is subtle. Nevertheless, the framework has equal detection performance of all classes, which depict its flexibility and strength in dealing with different cyber threat profiles in table3.

Attack Type	**Precision**	**Recall**	**F1-Score**
Denial of Service	0.92	0.88	0.90
Probe	0.89	0.85	0.87
Remote to Local	0.78	0.80	0.79
User to Root	0.74	0.70	0.72
Normal	0.95	0.97	0.96
Average	**0.86**	**0.84**	**0.85**

Table 2: F1-Score, Precision, And Recall For The BAG-LSTMAFO Model Across Different Attack Types On The Test Dataset. The Model Demonstrates Higher Performance In Detecting Frequent Attack Categories Such As Dos And Probe, While Exhibiting Slightly Lower Results On Rare Categories Like U2R.

Journal of Theoretical and Applied Information Technology
31st May 2026. Vol.104. No.10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

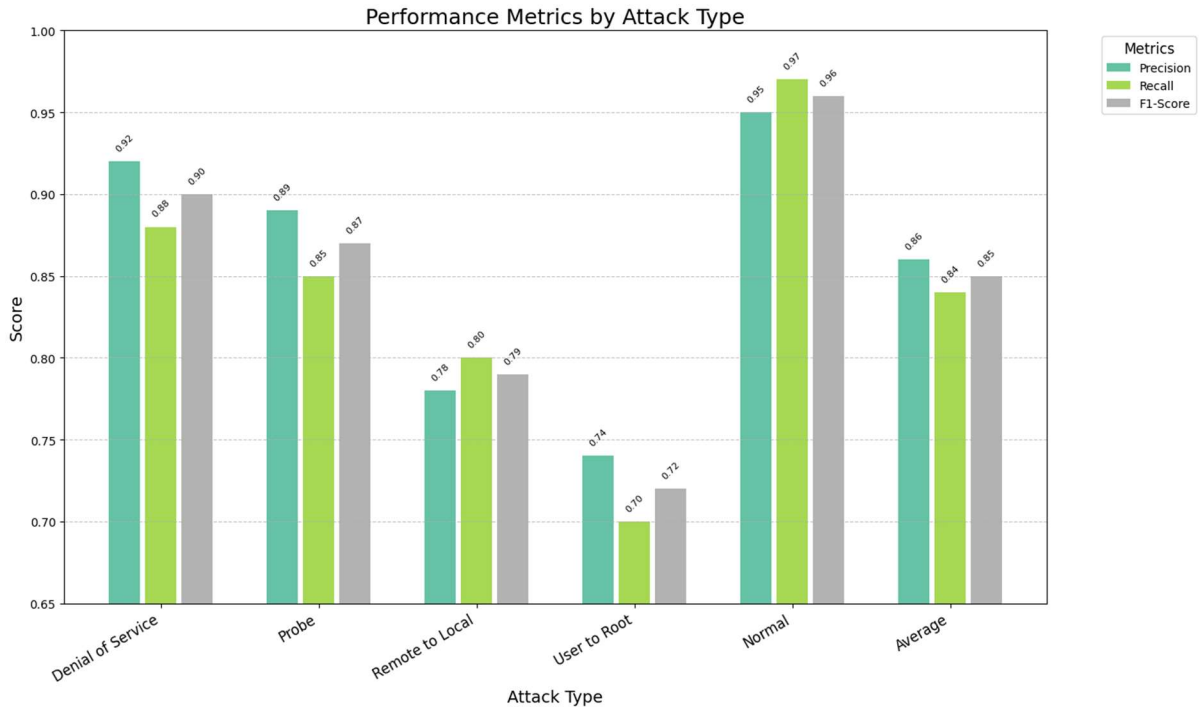

Figure 6. Performance Metrics By Attack Type

The Figure 6. Shows the performance metrics like precision, recall and F1-score by attack types like Denial of Service, Probe, Remote to Local, User to Root, Normal and average.

Table 3: Performance For Specific Attack Types

Attack Type	**Precision**	**Recall**	**F1-Score**
Denial of Service	0.93	0.91	0.92
Probe	0.90	0.87	0.88
Remote to Local	0.80	0.83	0.81
User to Root	0.76	0.72	0.74
Normal	0.97	0.98	0.97
Average	**0.87**	**0.86**	**0.86**

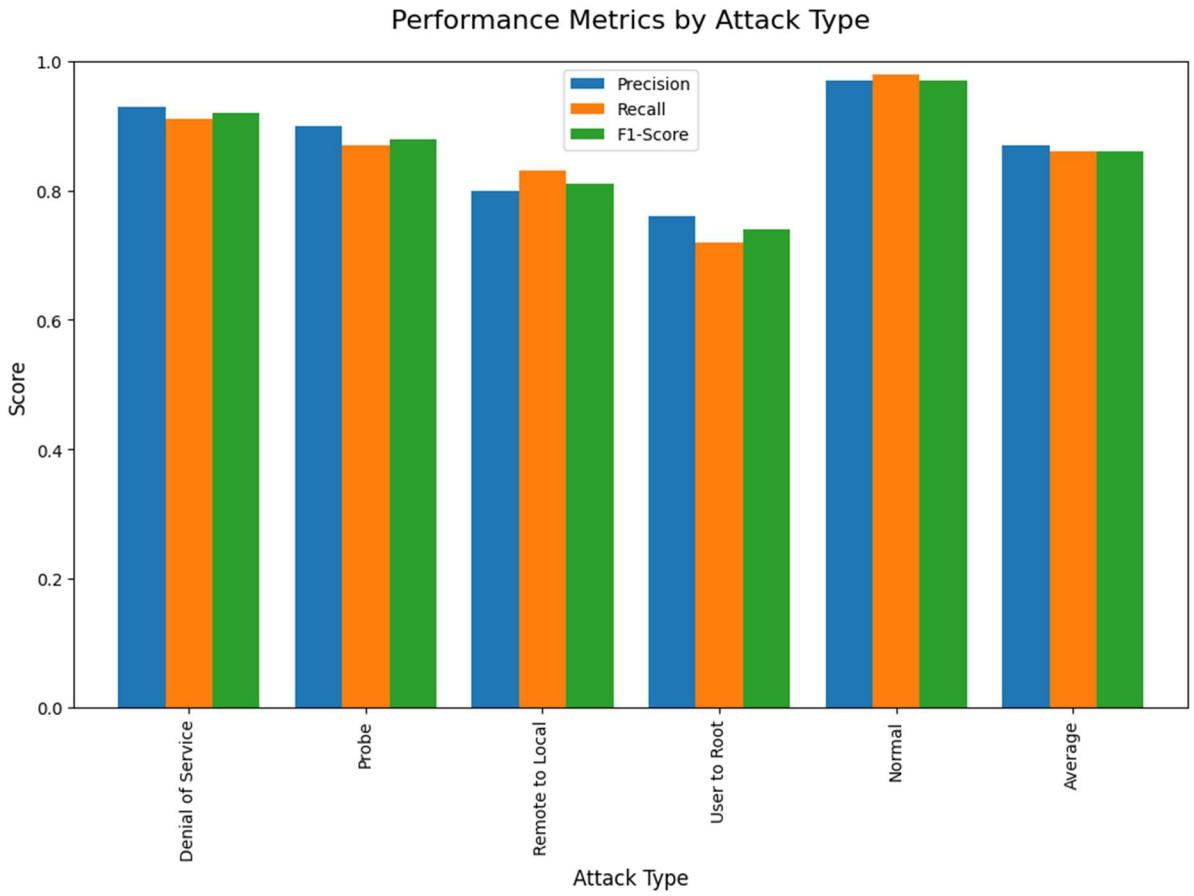

Figure 7. Performance metrics by attack type by Blockchain-Assisted Adaptive Boosting Graph-LSTM Framework with Firefly Optimization (BAG-LSTMAFO)

The Figure 7 shows the performance metrics by attack type with Blockchain-Assisted Adaptive Boosting Graph-LSTM Framework with Firefly Optimization (BAG-LSTMAFO). The proposed **Blockchain-Assisted Adaptive Boosting Graph-LSTM Framework with Firefly Optimization (BAG-LSTMAFO)** demonstrated strong classification performance across multiple cyberattack categories, though performance varied slightly among them. For example, it achieved an **F1-score of 0.92** for **Denial of Service (DoS)** attacks, which typically involve overwhelming a target system with traffic, making it unavailable to legitimate users. In contrast, the **F1-score for User to Root (U2R)** attacks was **0.74**, which are more subtle intrusions involving privilege escalation, often represented by sparser patterns in the feature space.

This variation in performance can be due to the complexity and temporal sparsity of some of the types of attacks such as U2R and Remote to Local (R2L), and unequal number of training samples per category. The model can better learn the unique spatial-temporal patterns of DoS and Normal classes which are more common in the dataset, as compared to infrequent attacks such as U2R, since the infrequent attacks have limited and noisy representations. However, the framework has a good average F1-score of 0.86, indicating its effectiveness and strength in detecting and characterizing a multi-class of cybersecurity threats.

5.4 Discussion and Limitations

These findings show that blockchain-enabled data integrity, heterogeneous graphical representations of sophisticated entity relationships, LSTMs-based temporal dynamics, attention-based saliency event focus, and Firefly Optimization to hyperparameter optimization is a powerful and efficient method to detect malicious activities.

One of the strong points of the framework over methods that examine either of these dimensions is the fact that it has capabilities to model both structural and temporal aspects of cloud activities.

There are however some limitations in the study:

- **Dataset Representativeness:** NSL-KDD can not necessarily achieve good performance on all real-world cloud systems, which have very different architectures and workloads. Assessment over larger and more varied datasets, and in particular the cloud-native datasets, is desirable.
- **Blockchain Overhead:** A live blockchain system has computational and latency overhead when used in providing security. In this work, the blockchain element was that of columns, such as protocol type, service, flag and labels, such normal, DoS, Probe, R2L, U2R, the relevant name of the dataset and F1-score could be validated. The adoption of a full-scale deployment would demand a keen focus on the performance of blockchains.
- **Feature Engineering:** The quality of features extracted in the form of nodes or edges has a significant influence on the performance of the model. While we engineered a comprehensive set, further exploration of domain-specific features could yield improvements.
- **Zero-Day Attacks:** While the anomaly detection nature of G-LSTMs can help identify novel patterns, the model's ability to detect sophisticated zero-day attacks that closely mimic benign behavior needs further investigation.
- **Scalability of Graph Construction:** For extremely high-velocity log streams, the construction of graph snapshots in real-time can be challenging and may require distributed processing techniques not fully explored in this work.

6: CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this paper, BAG-LSTMAFO, a new Blockchain-Assisted Graph-LSTM model, has been suggested to combine an attention mechanism and Firefly Optimization used to vigorously identify malicious actions within cloud environments. The architecture uses blockchain to provide tampering-resistant activity logging, representation of cloud behavior as attributed heterogeneous graphs, and a Graph-LSTM (G-LSTM) with temporal attention to allow modeling intricate spatio-temporal reliance of cyber threats.

In an attempt to optimise the performance further, automated and efficient hyperparameter tuning using Firefly Optimization algorithm is used. The combination allows the framework to learn optimal configurations in an adaptive manner, which enhances detection accuracy and generalizability under diverse threat environments.

The experimental analyses of NSL-KDD dataset show that BAG-LSTMAFO significantly performs better compared to traditional machine learning models and baseline deep learning methods. The framework had a F1-score of 0.94 and AUC-ROC of 0.96, thus validating the high performance of the framework in terms of classification. In addition to it, ablation experiments confirm the importance of the attention mechanism and Firefly Optimization, which makes the mechanism significant in improving detection sensitivity and minimizing false alarms.

In general, BAG-LSTMAFO is a safe, smart, and versatile system of real-time cyberattacks detection in dynamic cloud infrastructures, solving the acute issue of scalability, complexity, and attack vectors transformation.

6.2 Future Work

The proposed framework has a number of avenues in which future research can be extended and enhanced:

- Assessment of Large-Scale Real-World Cloud Datasets: The next important step should be the evaluation of BAG-LSTMAFO on large datasets in the real-world environments of the production cloud, including IaaS, PaaS, and SaaS activities, to determine its efficacy and scalability in the real world.
- State-of-the-Art heterogeneous models: More complex heterogeneous graph models, like the type of Heterogeneous Graph Transformer (HGT) or models that can learn the relative significance of various types of relations dynamically may also be more representational.

- Online Learning and Adaptability: Exploring the option of online learning of the G-LSTM model to change with changing threat landscapes and concept drift in cloud behavior without the need to retrain the model completely.

- Explainability and Interpretability: increase the interpretability of the model at the scale of the attentions of time, i.e. adding the GNN explainability capabilities (e.g. GNNExplainer) to determine the important nodes and edges that make a malicious prediction.

- Complete Blockchain Implementation and Performance Testing: Implementing and testing the framework with a deployed permissioned blockchain to measure the overall end-to-end performance, such as, latency and throughput of safe logging and intelligent exchanging of information.

Proactive Threat Mitigation: The extension of the framework into detecting and also providing suggestions or automation of the actions of response to the detected malicious patterns and attributes.

- Cross-Cloud Federation: Understanding how the blockchain element might be used to enable secure and privacy preserving federated learning or threat intelligence sharing among various cloud tenants or even among various cloud providers.

Data Availability: The data used to support the findings of created new data set, this study is available from the corresponding author upon request.

Ethical Statement: Authors declared that no humans and animals are participated in this research.

Author contributions N.J and A.R designed the experiments. N.J and A.R performed all the experiments. N.J and A.R analyzed the data. N.J and A.R wrote the manuscript. All authors discussed the results and commented on the paper, and approved its final version.

Competing Interests: The authors declare no competing interests.

Funding Statement: Authors declared that no funding received for this research and publication.

Acknowledgement: Not applicable.

REFERENCES:

[1] Mohammad, Noor Islam S.. (2025). Data Scheduling Algorithm for Scalable and Efficient IoT Sensing in Cloud Computing. 10.48550/arXiv.2508.04334.

[2] Saveetha, D., Maragatham, G., Ponnusamy, V., & Zdravković, N. (2024). An integrated federated machine learning and blockchain framework with optimal miner selection for reliable ddos attack detection. *IEEE Access*, *12*, 127903-127915.

[3] Ogunseyi, T. B., & Thiyagarajan, G. (2025). An Explainable LSTM-Based Intrusion Detection System Optimized by Firefly Algorithm for IoT Networks. *Sensors*, *25*(7), 2288.

[4] Alotaibi, A. M. (2025). A Privacy-Preserving Blockchain Learning Model for Reliable Industrial Internet of Things Data Transmission. *SN Computer Science*, *6*(5), 531.

[5] Almuqren, L., Alqahtani, H., Aljameel, S. S., Salama, A. S., Yaseen, I., & Alneil, A. A. (2023). Hybrid metaheuristics with machine learning based botnet detection in cloud assisted internet of things environment. *IEEE Access*, *11*, 115668-115676.

[6] Lalama, Z., Goudjil, L., Cherbal, S., & Louail, L. (2025). Integration of metaheuristic and machine learning in Cloud-Fog-Edge IoMT applications: a survey. *Cluster Computing*, *28*(11), 733.

[7] Nandanwar, H., & Katarya, R. (2025). A hybrid blockchain-based framework for securing intrusion detection systems in internet of things. *Cluster Computing*, *28*(7), 471.

[8] Shawl, R. Q., Singh, M., & Hassan, M. M. (2025). Leveraging Cyber-Physical Security Solutions Blended With Machine Learning for Advanced IoT Botnet Detection. *IEEE Communications Standards Magazine*.

[9] Awotunde, J. B., Gaber, T., Prasad, L. N., Folorunso, S. O., & Lalitha, V. L. (2023). Privacy and security enhancement of smart cities using hybrid deep learning-enabled

blockchain. *Scalable Computing: Practice and Experience*, *24*(3), 561-584.

[10] Han, C., Alserhani, F. M., Ahanger, T. A., Almazmomi, N. K., & Hashmi, A. (2026). Adaptive cyber threat detection in internet of things environment using deep learning and metaheuristic optimization. *Peer-to-Peer Networking and Applications*, *19*(1), 38.

[11] Akhtar, M. M., Alasmari, S. A., Haidar, S. W., & Alzubaidi, A. A. (2025). Distributed denial of service attack detection and mitigation strategy in 5G-enabled internet of things networks with adaptive cascaded gated recurrent unit. *Peer-to-Peer Networking and Applications*, *18*(2), 81.

[12] Ogab, M., Zaidi, S., Bourouis, A., & Calafate, C. T. (2025). Machine Learning-Based Intrusion Detection Systems for the Internet of Drones: A Systematic Literature Review. *IEEE Access*.

[13] Hassan, S. M., Mohamad, M. M., & Muchtar, F. B. (2024). Advanced intrusion detection in MANETs: A survey of machine learning and optimization techniques for mitigating black/gray hole attacks. *IEEE Access*.

[14] Saheed, Y. K., & Chukwuere, J. E. (2025). CPS-IIoT-P2Attention: Explainable privacy-preserving with scaled dot-product attention in cyber physical system-industrial IoT network. *IEEE Access*.

[15] Bourechak, A., Zedadra, O., Kouahla, M. N., Guerrieri, A., Seridi, H., & Fortino, G. (2023). At the confluence of artificial intelligence and edge computing in iot-based applications: A review and new perspectives. *Sensors*, *23*(3), 1639.

[16] Alhayan, F., Alshuhail, A., Ismail, A. O. A., Alrusaini, O., Alahmari, S., Yahya, A. E., ... & Al Zanin, S. (2025). Artificial intelligence-driven cybersecurity: enhancing malicious domain detection using attention-based deep learning model with optimization algorithms. *Scientific Reports*, *15*(1), 23806.

[17] Kumar, D., Pawar, P. P., Ananthan, B., Indhumathi, S., & Murugan, M. S. (2024, May). CHOS_LSTM: Chebyshev Osprey optimization-based model for detecting attacks. In *2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT)* (pp. 1-6). IEEE.

[18] Naz, A., Ullah, I., Jonath, K. K., Uzair, M., Nizamani, A. H., & Mushtaq, H. (2025). Innovative cybersecurity solutions: a deep learning-driven model for accurate intrusion detection in network traffic. *Cluster Computing*, *28*(11), 695.

[19] Sharma, S. B., & Bairwa, A. K. (2025). Leveraging AI for Intrusion Detection in IoT Ecosystems: A Comprehensive Study. *IEEE Access*.

[20] Almuseelem, W. (2025). Secure latency-aware task offloading using federated learning and zero trust in edge computing for IoMT. *IEEE Access*.

[21] Mallidi, S. K. R., & Ramisetty, R. R. (2025). Advancements in training and deployment strategies for AI-based intrusion detection systems in iot: A systematic literature review. *Discover Internet of Things*, *5*(1), 8.

[22] Kumar, G. K., & Thirumaran, M. (2025, April). A Survey on Intelligent Attack Mitigation and Latency Optimization in High-Speed Edge Networks using Deep Learning and Bio-Inspired Algorithms. In *2025 5th International Conference on Trends in Material Science and Inventive Materials (ICTMIM)* (pp. 1480-1486). IEEE.

[23] Gudivaka, R. L., Basani, D. K. R., Gudivaka, R. K., Grandhi, S. H., Gudivaka, B. R., Murugesan, S., & Kamruzzaman, M. M. (2025). AI-Powered Robotic Cloud Automation-Based Dynamic Task Allocation and Process Optimization Using E-WFO and C2DRBM. *IETE Journal of Research*, 1-14.

[24] Al Siam, A., Alazab, M., Awajan, A., & Faruqui, N. (2025). A Comprehensive Review of AI's Current Impact and Future Prospects in Cybersecurity. *IEEE Access*.

[25] Mazhar, T., Irfan, H. M., Khan, S., Haq, I., Ullah, I., Iqbal, M., & Hamam, H. (2023). Analysis of cyber security attacks and its solutions for the smart grid using machine learning and blockchain methods. *Future Internet*, *15*(2), 83.