Blockchain Meets Adaptive Honeypots: A Trust-Aware Approach to Next-Gen IoT Security

Yazan Otoum*, Arghavan Asad*, Amiya Nayak†

*School of Computer Science and Technology, Algoma University, Canada
†School of Electrical Engineering and Computer Science, University of Ottawa, Canada

Abstract—Edge computing-based Next-Generation Wireless Networks (NGWN)-IoT offer enhanced bandwidth capacity for large-scale service provisioning but remain vulnerable to evolving cyber threats. Existing intrusion detection and prevention methods provide limited security as adversaries continually adapt their attack strategies. We propose a dynamic attack detection and prevention approach to address this challenge. First, blockchain-based authentication uses the Deoxys Authentication Algorithm (DAA) to verify IoT device legitimacy before data transmission. Next, a bi-stage intrusion detection system is introduced: the first stage uses signature-based detection via an Improved Random Forest (IRF) algorithm. In contrast, the second stage applies feature-based anomaly detection using a Diffusion Convolution Recurrent Neural Network (DCRNN). To ensure Quality of Service (QoS) and maintain Service Level Agreements (SLA), trust-aware service migration is performed using Heap-Based Optimization (HBO). Additionally, on-demand virtual High-Interaction honeypots deceive attackers and extract attack patterns, which are securely stored using the Bimodal Lattice Signature Scheme (BLISS) to enhance signature-based Intrusion Detection Systems (IDS). The proposed framework is implemented in the NS3 simulation environment and evaluated against existing methods across multiple performance metrics, including accuracy, attack detection rate, false negative rate, precision, recall, ROC curve, memory usage, CPU usage, and execution time. Experimental results demonstrate that the framework significantly outperforms existing approaches, reinforcing the security of NGWN-enabled IoT ecosystems.

Keywords— Blockchain, Adaptive Honeypots, IoT Security, Intrusion Prevention, Trust-Aware Networks, Next Generation IDS.

I. INTRODUCTION

The rapid proliferation of IoT applications has been facilitated by advancements in NGWN. These networks enable efficient and large-scale service provisioning through technologies such as Mobile Edge Computing (MEC), which enhances computational efficiency by processing data closer to users [1]. However, the increasing connectivity and integration of IoT devices within NGWN have exposed these networks to a wide range of cybersecurity threats [2]. Traditional intrusion detection and prevention mechanisms struggle to provide robust security due to the constantly evolving attack strategies employed by adversaries [3]. A recent 2024 cyberattack on a 5G-enabled IoT network disrupted critical healthcare and industrial services, underscoring the urgency of strengthening security in NGWN-based IoT environments. IDS play a crucial role in securing IoT networks, with detection methods categorized as Signature-based IDS (SIDS) and Anomaly-based IDS (AIDS). SIDS typically provide high detection accuracy by comparing incoming network traffic against predefined attack signatures. However, they fail to detect novel and unknown attacks [4]. On the other hand, AIDS attempts to identify attacks by extracting and analyzing various traffic features, but its effectiveness is often hampered by limited generalization ability and computational inefficiency [5]. While deep learning-based anomaly detection has been explored to improve detection accuracy [6], classification based solely on network flow features has proven to be insufficient [7]. Moreover, existing IDS approaches do not adequately address the need for real-time detection with minimal false positives and high adaptability to evolving attack strategies [8, 9]. Blockchain technology has emerged as a promising solution to enhance security in IoT environments by ensuring tamper-resistant and decentralized trust management [10]. While blockchain-based IDS solutions mitigate various security threats, they introduce concerns regarding computational overhead, transaction latency, and scalability, particularly in resource-constrained environments [11], [12]. Furthermore, deploying multiple network entities for monitoring and intrusion detection increases the complexity of security management [13]. The deployment of honeypots as deception mechanisms has proven effective in attracting attackers and collecting attack patterns to improve network security [14]. High-interaction honeypots engage attackers longer, allowing for comprehensive analysis of attack strategies [15]. However, existing honeypot-based approaches either deploy static honeypots or fail to dynamically adapt to emerging threats. Additionally, game-theoretic models proposed for attack detection [16] often assume perfect knowledge of attacker strategies, which is unrealistic in real-world scenarios. Despite advancements in IDS and security frameworks, critical security challenges persist in NGWN-enabled IoT networks due to the following limitations:

- High Latency: Traditional IDS approaches rely on centralized cloud processing, introducing delays in attack detection and response.
- High False Alarm Rate: Anomaly detection models often produce a high number of false positives due to insufficient training on diverse attack scenarios.

- Lack of Adaptive Defense Mechanisms: Existing solutions lack dynamic attack mitigation strategies, allowing attackers to bypass static security measures.

To address these challenges, we propose a trust-aware security framework that integrates blockchain authentication, advanced intrusion detection, and dynamic honeypot deception. Our approach is designed to enhance real-time threat mitigation while minimizing computational overhead. The key contributions of this research include:

- A blockchain-based authentication mechanism that ensures the legitimacy of IoT devices and prevents impersonation attacks using the DAA.
- A bi-stage intrusion detection framework that combines IRF-based SIDS with DCRNN-based AIDS, improving detection accuracy while maintaining efficiency.
- A moving target defence strategy that dynamically migrates services to trusted edge nodes based on trust evaluation, mitigating the risk of service compromise.
- The deployment of high-interaction, on-demand virtual honeypots to deceive attackers, extract attack signatures, and enhance IDS training using securely stored attack patterns.

The proposed approach is implemented and evaluated in the NS3 simulation environment, demonstrating a 25% improvement in detection accuracy, a 30% reduction in false negatives, and enhanced resource efficiency compared to existing methods. The remainder of this paper is organized as follows: Section II gives the background. Section III provides a comprehensive literature review of existing intrusion detection and prevention mechanisms. Section IV outlines the problem statement and research challenges. Section V presents the proposed methodology in detail. Section VI discusses the experimental setup and performance evaluation. Finally, Section VII concludes the paper and outlines directions for future work.

II. Background

This section discusses the background of Blockchain technology, the Need for honeypot deployment, and integrated blockchain honeypot deployment to improve security attack mitigation in an IoT environment [17]. Blockchain is an open-access ledger of digital transactions for public networks where the attacker cannot change or compromise. It is a decentralized platform managed by several authorities and mitigates the issue of less security caused by the centralized model. It allows its users to deal directly with each other and reduces security issues caused by third parties. Blockchain technology in the field of IoT demands high levels of demand because of its decentralized nature. All transactions, such as authentication, network access, etc., are stored in blockchain because of its resource-constraint nature and tamperproof nature, which allows its users to verify by digital transaction. Information collected by the users is chained by blocks using a ledger, which is tightly chained with each other with the help of hashed cryptographic keys. The most common method to store transactions in the blockchain is the Merkle tree, which stores the transactions done by individuals, and the root hash

tree, which stores them in the blockchain. Each transaction is hashed in a cryptographic manner. Miners are used to store the upcoming transactions, which generate a key for every user and allow them to join the ledger. The data in the blockchain cannot be modified or hacked, providing that it has an immutable character. Some of the other significant features of blockchain are fault tolerance, transparency, improved security, and decentralized nature.

Honeypot is an intelligent attack detection technology that shows the attacker's intention and reports the system's status, whether there was an abnormal behaviour or an external attacker in the network environment and captures the way they attack. It is the copied environment of the original environment with the same configurations that attracts the attackers to hack and record their patterns to store them in the log files. The storage of attack patterns helps to drop the malicious users in the future, who will be identified as attackers based on their patterns, which improves the overall safety of the system.

A secure and reliable IoT system must integrate blockchain and honeypot technology to improve IDS security. Physically deployed honeypots are used for IDS, which is effective and improves the overall system safety; however, the attackers may compromise them, or the attackers might have awareness about the honeypot, which creates risk in the IoT environment. By using physically deployed honeypots, the overall cost of the system would be high, leading to more energy consumption. To overcome the above challenges, virtual honeypots are deployed at the time of attack, which improves overall safety and low cost. Table I provides the distinct characteristics of blockchain and honeypot technologies. Hence, in our approach, we integrate the blockchain honeypot methodology in IDS to detect and prevent intrusions in the NGWN-IoT environment.

III. Literature Survey

This section surveys the literature on several intrusion detection and prevention schemes, including Artificial intelligence (AI)- based schemes and honeypot-based schemes. It analyzes the objectives, workings, and research gaps of these approaches.

A. Intrusion Detection and Prevention Schemes

An intrusion detection and prevention system for NC-assisted mobile small cells was proposed in [18]. Initially, the nodes divide the message into a sequence of packets. The proposed work has four steps; the first is the tag generation process that calculates the tag value for every coded packet. The second one is the swapping process, which avoids tag pollution attacks from the coded packets, in which the source node swaps the tag and coded packets. The third one is the key distribution process; here, the maximum counts of key vectors are assigned for every intermediate and destination node. Finally, the verification process verifies the swapped coded packets. During verification, if the result equals zero, the packet will be sent to the next level; otherwise, it will be dropped. However, this approach uses a single security parameter for key generation, which is insufficient for security;

TABLE I: FEATURES OF BLOCKCHAIN AND HONEYPOT

Features	Blockchain	Honeypot
Security	Provides high security due to its immutable feature. Privacy is protected using hashing.	Security is provided by attracting attackers and learning their strategies.
Risk	Low risk due to robustness.	Moderate risk when relying solely on a honeypot for security.
Information Gathering	Information is stored in blocks after consensus validation.	Information gathered is stored in log files, revealing attacker strategies.
If Integrated with IDS	Provides authenticity and security.	Helps in intrusion prevention by learning new attack strategies.

hence, it will be easily compromised by the attackers, leading to poor security. Multi-agent assisted IDS in IoT environment using Blockchain technology was proposed in [19]. Three operations were used in this paper: data collection, data management, and data response. For testing the proposed work, the NSL-KDD dataset is used to detect attacks from the transport layer. For attack detection reasons, the NSL-KDD dataset is used, and the agents used for intrusion detection are as follows: communication agent, response agent, collection agent, host agent, detection agent, and training agent. Based on the attack information, the database is trained for the network. However, the deployment of a larger number of agents increased the network complexity. The computational overhead for running all these agents over the blockchain increases transaction verification and hash generation times.

A practical approach for automatically protecting NGWN mobile networks using multiple tenants was proposed in [20]. The network architecture is comprised of a radio access segment, edge computing segment, core network segment, and interdomain segment. The dynamic traffic of the NGWN network was considered, and an effective IDS was proposed. The network is divided into two types, namely, data network and management network. The automatic detection of intrusions is executed using several agents, namely flow control agents, action enforcers, decision-makers, and security monitoring agents. The packets in the traffic flow were classified by the security monitoring agent, which was responsible for the generation of alerts for the decision maker to decide the corresponding action. The significant attacks mitigated by this approach are UDP flooding attacks and DDoS attacks.

A multi-view consistent Generative Adversarial Network (GAN) model was designed to enhance Intrusion Detection and Prevention Systems (IDPS) for IoT [21]. This model leverages multiple data views to improve the detection of security threats. A practical framework for detecting intrusions in the IoT based on real-time dataset was proposed in [22]. The possibility of attacks on several aspects of the IoT network was described, and the necessity of an effective intrusion detection system was demonstrated. The effect of the dataset's quality, which affects the performance of IDS, was evaluated to design a real-time dataset. The proposed dataset consisted of data obtained from various IoT nodes containing attack patterns of known attacks. Some of the attacks considered in this approach were flooding, black holes, and selective forwarding attacks. Thereby, the users were able to obtain the dataset to perform efficient intrusion detection. Several known attacks were identified by utilizing the real-time dataset, but the features extracted for identification were not sufficient to perform the detection of other complex intrusions.

An intrusion mitigation technique for mitigating attacks in the IoT environment was proposed in [23]. The authentication of the user is carried out to allow only legitimate users to participate in data transmission. Further, an ongoing traffic analysis was carried out based on the packet flow features. This work undergoes three stages in which the log files of the authenticated entity are examined to detect attacks caused by bots. Then, the attacker was detected by analyzing the typical pattern of attack, thereby isolating the attacker from the network. An encryption-based approach for secure data collection, storage and access in cloud-assisted IoT was proposed in [24]. The issues in the proper management of data in cloud storage were addressed. The IoT sensors were used for periodic data acquisition from the environment and stored in cloud storage. Here, the attacker in the network was able to compromise data to exploit its integrity. For this purpose, the authors proposed a conditional identity-based broadcast proxy re-encryption (CIBRE) to maintain the confidentiality of the data. The re-encryption of data was also provided to ensure the integrity of the cloud data each time it was decrypted. Identity-based encryption was implemented to ensure the confidentiality of the data, but entities such as cloud storage can be compromised by malicious attackers, causing a single point of failure.

B. AI-based intrusion detection schemes

A deep learning method for detecting network anomalies by extracting features from the network traffic in NGWN networks was proposed in [25]. The proposed architecture has four components: virtual infrastructure, virtualized network functions, management and orchestration, and operations and business support systems. Initially, the network flow is collected from the user equipment. From the network flow, features are extracted by a feature vector. Based on the extracted features, the anomaly symptoms are detected and sent to the LSTM for recognizing the anomaly patterns. The simulation result shows that the proposed system achieves high performance in terms of anomaly detection accuracy. This approach takes only flow-based features for anomaly detection, thus reducing classification accuracy. It takes all the packets as input, therefore increasing the latency and complexity of the process. A machine learning algorithm for detecting network anomalies in a 5G network was proposed in [26]. The data are pre-processed to enhance detection

TABLE II: RESEARCH GAPS IN LITERATURE SURVEY

Schemes	Reference	Objective	Algorithms/Methodology	Disadvantages
Intrusion Detection and Prevention Schemes	Parsamehr, R. et al. [18]	Intrusion detection and prevention system for NC-assisted mobile small cells	Four-step approach (Tag generation, Swapping, Key distribution, and Verification)	1. Poor Security 2. Less efficiency in detecting malicious users
	Liang, C. et al. [19]	Multi-agent assisted IDS in IoT environment using Blockchain technology	Agents-based approach (Communication agent, Response agent, Collection agent, Host agent, Detection agent, and Training agent)	1. High Complexity 2. High Computational overhead 3. Lack of Blockchain authentication parameters
	Mamolar, A. S. et al. [20]	Automatic protection of 5G mobile networks using multiple-tenants	Agents-based approach (Flow control agents, Action enforcer, Decision maker, and Security monitoring agent)	1. Energy consumption due to multiple agents 2. Less secure
	Al-Hadhrami, Y. et al. [22]	Framework for the detection of intrusions in the IoT based on real-time dataset	Dataset-based approach (Flooding attack, Black hole attack, and Selective forwarding attack)	1. Lack of features 2. High Computation required
	Hatzivasilis, G. et al. [23]	Intrusion mitigation technique for mitigation of attacks in the IoT environment	Three-stage approach (Authentication, Detection and Security)	1. Inefficient legitimate users' authentication 2. Inefficiency in malicious user classification and detection
	Wang, W. et al. [24]	Encryption-based approach for secure collection of data, storage and access in cloud-assisted IoTs	Conditional Identity-based Broadcast proxy RE-encryption (CIBRE) method	1. Single point failure 2. No transitivity by using CIBRE 3. Poor security
	Rajkumar, M. et al. [21]	Generative adversarial networks (GANs) based approach to enhance protection against security threats in IoTs	Developing a GAN architecture to learn from multiple data gathered from various sources in IoT simultaneously	1. High Computational Complexity 2. Challenges in obtaining diverse and qualified data dynamically 3. Poor Scalability
AI-based Intrusion Detection Schemes	Maimó, L. F. et al. [25]	Deep learning method for detecting network anomalies by extracting features from the network traffic in 5G network	Four-step approach (Virtual infrastructure, Virtualized network functions, Management and Orchestration)	1. High latency 2. High complexity 3. Low classification accuracy
	Lam, J. et al. [26]	Machine learning algorithm for detecting network anomalies in 5G network	Machine Learning Algorithm	1. Time complexity due to lots of training required 2. Lack of classification parameters
	Yang, A. et al. [27]	Intrusion detection system for IoT environment using improved BP neural network	LM-BP Algorithm	1. Attack detection inefficiency 2. Sensitive to noise 3. Poor security
	Mondal, A. et al. [28]	Security approach for cloud storage based on encryption mechanism	Co-occurrence Matrix Algorithm and cryptographic algorithm	1. Required several training data 2. Inefficient honeypot adoption leads to less security
	Eskandari, M. et al. [29]	Gateway-based detection of network intrusions	Gateway-based (Timings involved in the traffic flow and Data flow statistics over a particular time interval)	1. Pour scalability 2. Inefficient features classification for intrusion detection leads to several attacks
	Li, B. et al. [30]	Intrusion detection scheme in industrial cyberphysical systems	Three model approach (Trusted authority, Industrial agents and Cloud server)	1. High latency 2. High computation due to several agents
	Mahdi, M. A. et al. [31]	Implementing a trained hybrid machine learning technique based on the extracted features from live IoT traffic	Hybrid (Decision trees-support vector machines-neural networks)	1. High Computational Overhead 2. Gathering Large amounts of high-quality data from various IoT devices to train effectively
	Mallidi, S. K. R. et al. [32]	Examining different machine learning models and integrating them into IoT systems for effective intrusion detection	Systematically analyze and summarize existing research on the training and deployment strategies of AI-based IDS in IoT	1. Complexity 2. Data privacy concern 3. Frequent retraining and updates
Honeypot-Based Intrusion Detection and Prevention Schemes	Al-Mohannadi, H. et al. [33]	Honeypot-based analysis of cybersecurity attacks	Parameters-based approach (IP address, Domain name, Username, Password, and Geographic location)	1. Lack of authentication parameters leads to malicious user registration 2. Honeypot creation seems to be less secure
	Lee, J. et al. [34]	Detection of malicious users using a file-based deception technique	Three-step approach (Regular files, Fake files, and Sensitive files)	1. Files are easily compromised if containing sensitive data 2. High latency due to file generation during honeypot deployment
	Li, B. et al. [35]	Efficient approach for learning attack strategies using a honeypot-based deception strategy	Hybrid Game Theoretic-based Approach (One player and ICPS defender)	1. Leads to security threats 2. Does not consider all types of attacks
	Dara, N. et al. [36]	Deployment of honeypots—decoy systems mimicking real IoT devices designed to attract potential attackers	Establishing a honeypot designed for IoT devices to provide deeper insights into IoT threats	1. High resource intensive 2. Limited Coverage 3. Lack of Real-Time Detection
	Ntizikira, E. et al. [37]	Honeypot technology with blockchain principles combination	Integration of edge computing with ensemble learning models	1. Poor Scalability 2. Real-time Processing Constraints 3. High Cost and Resource Overhead

accuracy. The features are extracted from the pre-processed data. Weibull distribution is used to select the features from the normal traffic. The Convolutional Neural Network (CNN), which was designed with the NAS method, is used to classify anomalies using features. The proposed work achieves high accuracy and low latency by optimizing the neural network in the 5G network. A hybrid machine learning technique (decision trees-support vector machines-neural networks) to improve detection accuracy and robustness against various types of abnormal or malicious activities in IoT was proposed in [31]. Various training methodologies for AI-based IDS, proposed in [32], highlighted the importance of selecting appropriate datasets, feature selection techniques, and model training processes to enhance IoT accuracy and efficiency. The paper discussed deployment strategies, including integrating IDS into IoT networks, real-time monitoring capabilities, and the scalability of AI models to handle the dynamic nature of IoT environments. An intrusion detection system for an IoT environment using an improved BP neural network was proposed in [27]. The proposed system used the LM-BP algorithm for intrusion detection. Initially, the data are collected from the data source. From the collected data, features are extracted using the proposed neural network. Here, the neural network was divided into two sections; one section was used to optimize the connection weight value, and another was used to optimize the learning rate, which adjusts dynamically to improve the detection rate. It was used to detect intrusion behaviour from the extracted features. After detecting the intrusions, the neural network was updated. The proposed system detected DoS, R2L, U2L and probing attacks using the KDD CUP 99 dataset. However, in this approach, only particular attacks were detected, and the prevention of these attacks was not investigated, which affects the overall security of the system. A practical security approach for cloud storage based on an encryption mechanism was proposed in [28]. Initially, the data was normalized in the dataset to improve the accuracy by removing the unwanted data and processing the missing data. Then, the grey-level co-occurrence matrix algorithm was implemented to extract the significant features from which attacks were classified using a CNN-based classifier. The security of the data was ensured by adopting a honeypot-based cryptographic algorithm, which is responsible for performing encryption or decryption of data. The gateway-based detection of network intrusions was presented in [29]. Generally, it needs a scalable dataset which enables the addition of new attacks in various settings. Dynamic features were considered on the gateway to extract the features for normal and abnormal classification. It avoids considering features which were static concerning the environment; the features considered were related to the flow of packets, timings involved in the traffic flow and data flow statistics over a particular time interval. Since Passban was an anomaly-based IDS, it could be trained while observing the target system's network traffic in a normal state.

An intrusion detection scheme in industrial cyber-physical systems based on federated deep learning was proposed in [30]. The system model comprises three main entities: trusted authority, industrial agents and cloud servers. The industrial agents represent the industrial administrators responsible for improving the intrusion detection system. The trusted authority was responsible for key generation to authenticate both the cloud server and industrial agents. The threat model, consisting of several significant threats, such as command injection attacks, response injection attacks, reconnaissance attacks and DoS attacks, was explained. The deep-fed scheme was proposed, which comprised the integration of CNN and Gated Recurrent Unit (GRU) based intrusion detection system and secure communication protocol based on Pallier cryptosystem was executed.

C. Honeypot-based intrusion detection and prevention schemes

The honeypot-based analysis of cyber security attacks was proposed in [33]. Cyber security threats are identified to detect and mitigate the behaviours of web services. Information Collected from the data included IP address, domain name, user name, password and geographic location of the attacker. Due to the centralized environment, it was complex to identify and attract attackers. For intrusion detection, honeypot technology is used, which mimics real-time systems and determines the attacks. Finally, a log file report is generated for the collected data and stored in the cloud servers.

An approach for detecting malicious users using a file-based deception technique was proposed in [34]. A hidden interface was established for legitimate users, which malicious users cannot access. The system files are divided into three types, namely, regular files, fake files and sensitive files, in which the regular files and fake files are accessible for both the regular interface and hidden interface. The sensitive files were accessible only to legitimate users who communicate through a hidden interface. The honeypot-based deception technique was implemented in which the malicious users were attracted to attack the honey files, which are referred to as fake files, from which the behaviour of attackers, including their resources, was identified to improve the security of the sensitive files. An efficient approach for learning the attack strategies using a honeypot-based deception strategy for industrial cyber-physical systems was proposed in [35]. Here, the attack strategies of attackers after identifying the honeypot were studied to provide overall security to the ICPS. The application of advanced analytics techniques to the data gathered from honeypots, aiming to identify and understand emerging threat patterns in IoT environments, was proposed in [36]. Here, a machine learning model is used to analyze attackers' behaviour captured by honeypots and identify subtle, previously unseen anomalies that may signal new or evolving attack strategies. An edge-assisted ensemble learning model within the honey-block framework to proactively identify and mitigate potential security breaches for intrusion detection in IoT was proposed in [37]. This framework combines the power of edge computing and ensemble learning to create a more secure environment for IoT systems. By aggregating predictions from various models, ensemble methods reduce the likelihood of false positives and enhance the system's ability to detect complex, previously unknown attacks.

A hybrid game theoretic-based approach was implemented, and the players had different objectives. The two-player game was proposed with attackers as one player and ICPS defenders as another player. The one-shot signalling model was developed. The perfect Bayesian equilibrium-based probability was calculated for the successful defence of all the attacks provided by the attackers. However, performing only honeypot-based intrusion detection in the ICPS is not enough to mitigate all types of attacks that threaten the system's security. Table II provides the working of existing approaches from which the research gaps in the literature survey are determined based on methodology and disadvantages.

IV. PROBLEM STATEMENT

This section deals with the major problems encountered in the existing approaches to detecting and preventing cyber-attacks in the NGWN-IoT environment. These problems are considered as the problem statement of our approach, and the research solutions provided by our approach are also mentioned. The IoT devices' legitimacy was considered a significant property to be ensured to mitigate several cyber-attacks in the IoT environment [38]. The RFID-based authentication was implemented to ensure the security of transmitted data. The RFID credentials of each entity were registered, and a public address was generated. The lightweight authentication was performed by implementing simple bit-wise exclusive OR functions. The problems faced by this approach are mentioned as follows,

- The authentication of entities was performed based on RFID tags, but considering only one credential for authentication limits the system's security as the attackers could manipulate the RFID.
- The RFID-based authentication doesn't capture unknown attacks, thereby increasing the attackers' involvement rate to launch more attacks in the IoT user environment.

Several game theoretic models were proposed for intrusion detection and prevention in the cloud server. The risk level of virtual machines in the cloud server was computed, and the honeypot-based analysis of the attack pattern was performed [39]. The game model for intrusion detection in the cloud was executed, which carried out various types of IDS such as SIDS, AIDS, and honeypot-based IDS [40]. However, these approaches encounter several problems that are mentioned as follows:

- Since all the processes take place in a cloud environment, there is a great threat to the security of the cloud, and the centralized nature of the cloud environment results in a single point of failure and violation of quality of service and SLA.
- The game theory-based IDS model detected attacks by performing signature-based, anomaly-based and honeypot-based IDS, but the overall security of the cloud computing was not ensured as it can be compromised through malicious users.
- The intrusion detection was implemented as a game theory-based model in which if the number of attacks

increased, the approach's complexity also increased, thereby increasing the latency.

The honeypot-based detection and mitigation of cyberattacks were performed using various approaches. The rule-based determination of malicious behaviour possessed by the data packet was presented in [41]. The multiport honeynet model was designed with several interactive level honeypots to extract information about the attacker strategy [42]. The problems faced by these approaches are mentioned as follows:

- Only honeypot-based intrusion detection was carried out in which the attacker was assumed not to be aware of honeypots. But if the attacker identifies a honeypot and performs the attack in the system file by omitting it, it will be useless.
- The type of honeypot used in this approach was a low interaction honeypot, which will obtain only less information about the attacker pattern, which was not enough to perform effective intrusion detection.
- The log files are generated on the corresponding servers to improve the security of the SOAP ports. However, these files can also be compromised by the attackers, affecting the integrity of log files.

The proposed approach is designed to mitigate all the research problems encountered in the existing approaches. In our approach, the blockchain-based authentication of IoT devices is performed based on attributes such as PUF, device ID and MAC address. These attributes are unchangeable and hence provide a high degree of security. The SIDS and AIDS models are utilized based on attack patterns and spatiotemporal features to detect known and unknown attacks. The deployment of global and local mobile edge computing for provisioning services was carried out, which mitigates the limitation of increased latency and performs in a decentralized manner. The service provisioning is executed in the mobile edge computing node, where the migration of services is computed based on trust calculation. As the process is carried out in the edge node, the time consumed is reduced, fulfilling the quality of service and SLA constraints. The on-demand placement of virtual high-interaction honeypots is carried out to attract the attackers and analyze their strategies. These strategies are stored in the log files in an encrypted manner and are used to train the SIDS model.

V. PROPOSED MODEL

In this work, we focus on detecting and preventing intrusions in an IoT environment. Here, security and privacy of the IoT environment are accomplished for both IoT devices and users by using honeypots and blockchain technologies. Fig. 1 illustrates the system topology and conceptual flow of the proposed approach, and all the notations and parameters used in this paper are shown in Table III.

A. System Model

The system model comprises a set of IoT devices $D = \{D_1, D_2, ..., D_n\}$ from which the data are generated and sent to the cloud server for processing and storage purposes. The IoT
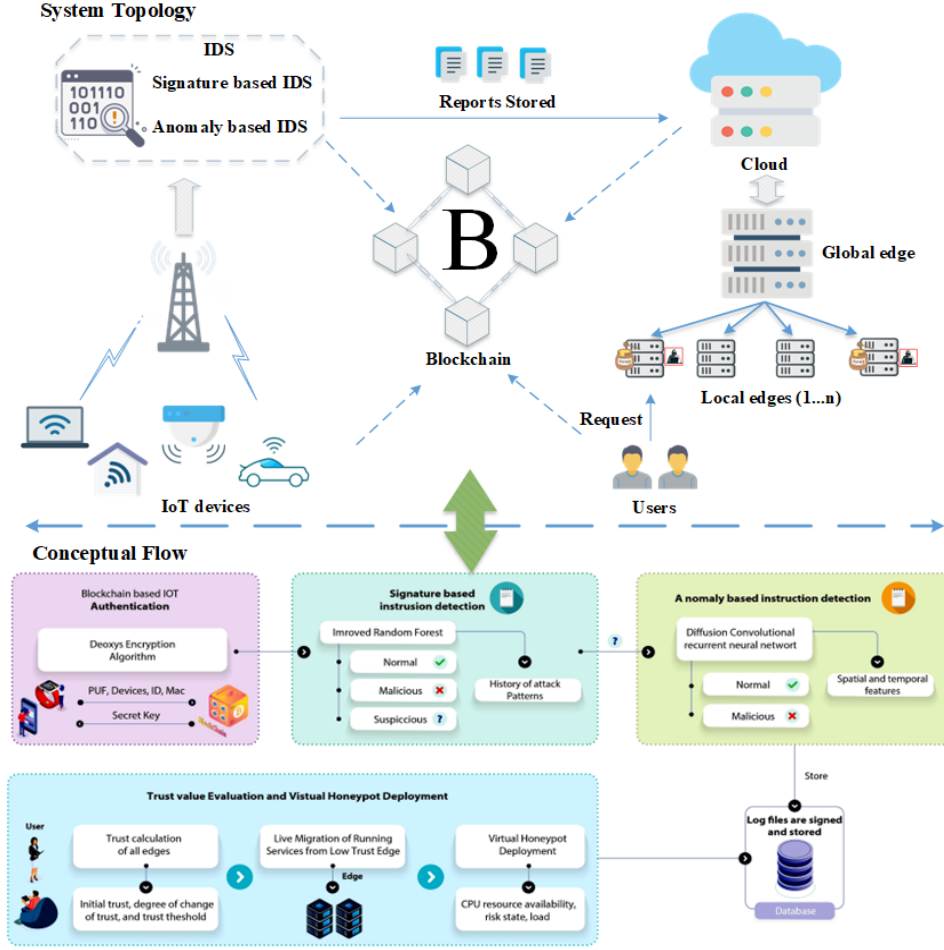
Fig. 1: Model topology and proposed architecture

environment consists of a set of IoT users $U = \{U_1, U_2, ..., U_n\}$ who request service from the cloud server. The proposed IDS is deployed between the IoT devices and the cloud server for filtering data packets from the devices through the 5G Base Station (BS). The edge layer comprises a global edge server (GE) and a set of local edge servers $LE = \{LE_1, LE_2, ..., LE_n\}$ that are placed at the edge of the network to provide services to the IoT users with optimal QoS, quality of service SLA requirements. The attackers in the system are considered on both the IoT devices side and the user side, who try to affect the provisioning of legitimate services to the users. On the IoT devices side, the attackers deploy a malicious IoT device or compromise the legitimate IoT devices to generate malicious packets. On the other hand, the attackers on the user side try to compromise the user device or directly local edge servers to affect the provisioning of services. The processes involved in the proposed approach to perform secure service provisioning in the IoT environment are described as follows.

B. Blockchain-based IoT authentication

In an IoT environment, all the devices are connected to the internet to share their data. Due to the sharing of large amounts of data over the internet, the IoT environment faces security challenges. To ensure the legitimacy of the IoT nodes, we perform authentication. Initially, all the IoT devices and users register their information, such as PUF, Device ID ($Dv - ID$), User $ID(Ur - ID)$, and MAC address to the key generator (KG) integrated with blockchain. The user provides all four parameters, whereas the IoT devices provide three parameters except $Ur - ID$. The KG then provides a secret key (K_{pr}) to the respective devices and users by utilizing DAA, which is stored in the blockchain. Once the registration is over, the respective IoT devices and users become a part of the network. The purpose of DAA's purpose is to provide authentication with less time consumption. During packet transmission and request generation, the devices and users generate a tag (k_{tg}) and encrypt the k_{tg} using the encryption key $EN(k_{tg})$. The generation of tags provides more security during authentication. The transaction consisting of (K_{pr}), $EN(k_{tg})$, key lifetime (k_L), and current timestamp along with the message and request are submitted to the blockchain. The blockchain verifies the transaction by using the consensus mechanism. During consensus, the blockchain nodes verify whether the submitted credentials are the same as those of the registered ones. If the consensus is achieved and the transaction is verified, the new block is mined, and the respective transaction is recorded in the blockchain. By doing so, the legitimacy of the IoT devices and users is verified before taking part

TABLE III: TABLE OF NOTATIONS

Notation	Description
D_{ID}	Device ID
U_{ID}	User ID
K_{pr}	Secret Key
K_{tg}	Tag
K_L	Key lifetime
Θ_0	Initial forest
Z_0	Number of Trees
ς_0	Feature vector
$W^T(k)$	Weight of the feature K
α^T	Weight of the tree T
Δ	Important feature bags
$\delta h, \delta g$	Variation of important and unimportant features
M_{av}	Average number of nodes in a tree
I_l	Input image length
Q	Number of zeroes
K_r	Size of the kernel
SK	Kernel stride
$tanh, \lambda$	Activation functions
$\tau(E)$	Trust value of edge server
In_τ	Initial trust value
ϑ	Degree of change of trust
RA_{CPU}	CPU resource availability
E_i, T_i	Number of edge servers, Number of tasks
SA_N^{DIM}	N search agent with dimensions
FV	Fitness value
R_n, B_n	Root node and branch nodes
G_{2P}^{nxm}	Gaussian samples with matrix size and prime
N_σ^n	Gaussian distribution with n dimensions
K_c	Shift vector
re	Random bit
d_s	Sample vector
F	Challenge vector
$\vec{R}, \vec{Bn}$	Index vector position
$T(ft)$	Temporary fitness value

Algorithm 1: Authentication Process

Input: K_{pr}, message packet/request, time stamp, k_L
Result: Authentication
1 **Begin**
2   Generate a tag $k_{tg} \in R_N$
3   Encrypt $k_{tg} \to \text{EN}(k_{tg})$
4   Generate transaction T.Authenticate to blockchain
5   **foreach** *T.Authenticate* **do**
6   **if** $k_L < time\ stamp$ **then**
7   Verify T.Authenticate through consensus
8   **if** *res.consensus = 0* **then**
9   Drop message/request // Illegitimate
10   **else**
11   Accept message/request
12   **else**
13   Drop transaction
14 **End**

C. Intrusion detection System

The intrusion detection system is used to detect attacks which are performed through compromised devices in the environment. This improves the security of the IoT environment. In our work, we perform two types of intrusion detection, such as

1) Signature-based intrusion detection: It is used to detect known attacks that have already been trained and stored in the database. It sends the data to perform intrusion detection. First, we detect signature-based intrusion detection using the IRF. The IRF possesses increased classification accuracy by selecting the important features and optimally constructing the number of trees. The forest Θ_0 is initially grown with Z_0 number of trees and feature vector $\varsigma_0(.)$. The ranking of features is executed based on the respective weight of the features, which can be computed as,

$$w(k) = \frac{\sum_{\forall T} W^T(k) \cdot \partial^T}{\max_k \sum_{\forall T} W^T(k) \cdot \partial^T} \quad (1)$$

Where $W^T(k)$ represents the weight of feature k with respect to tree T, and ∂^T denotes the overall weight of tree T. From the ranked list of features, the important features h_0 are selected and grouped into a pool of important features denoted as Δ. Similarly, the pool of unimportant features is denoted by Δ'. The mean and standard deviation of the weights of features in Δ' are represented as α_0 and β_0, respectively. Let Rf_0 be the number of features in Δ' whose weights are less than $(\alpha_0 - 2\beta_0)$. These features Rf_0 are further removed from Δ' to form a refined feature set $\varsigma_1(\cdot) = \varsigma_0(\cdot) - Rf_0$. The condition for including a feature k from Δ' into Δ can then be formulated as:

$$w(k) \geq \min_{j \in \Delta} w(j), \quad k \in \Delta' \quad (2)$$

If a feature is termed an important feature, it cannot be removed. The number of important and unimportant features
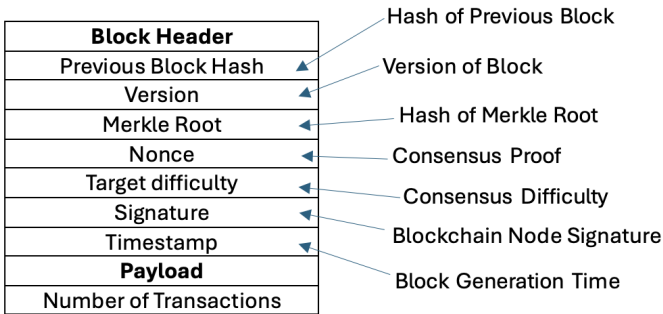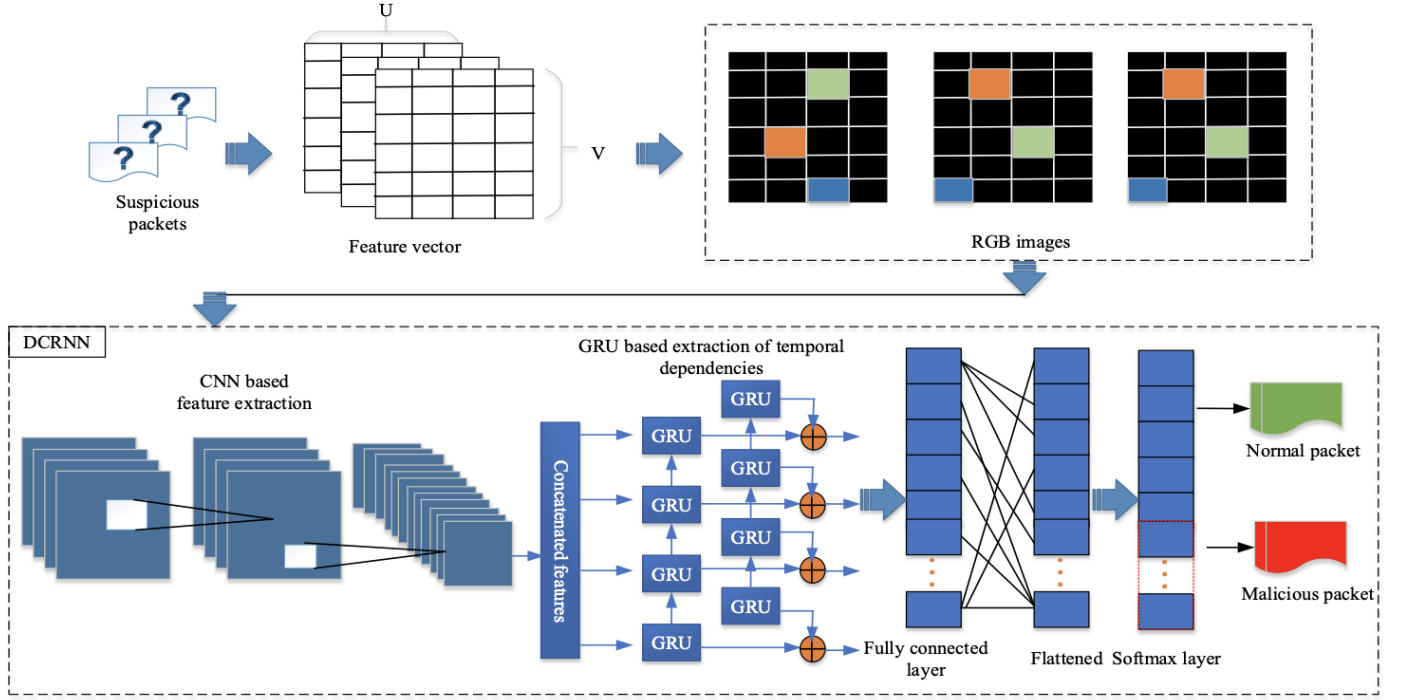

Fig. 2: Block format.

in transmission. Only authenticated IoT devices can send the data through a 5G gateway. The Pseudo code for the proposed blockchain-based authentication process is provided in Algorithm 1. The structure of the generated block in the blockchain is illustrated in Fig. 2 The description of each field is provided in the figure.

Fig. 3: DCRNN-based detection of suspicious packets.

can be expressed as h and g from which the variation of important and unimportant features can be computed as:

$$\delta h = \Delta_{n+1} - \Delta_n \tag{3}$$

and

$$\delta g = \Delta'_{n+1} - \Delta'_n \tag{4}$$

The number of trees to be added to the existing number of trees in the forest in order to improve the accuracy can be computed as:

$$|\delta Z| \leq \left| \frac{l(p_u \delta h + p_g \delta g)}{g} \right| \tag{5}$$

where l is expressed as:

$$l = Z M_{av} P^{M_{av}-1}(1 - P^{M_{av}})^{Z-1} \tag{6}$$

where $P < 1$, M_{av} is the average number of nodes in a tree, and p_u, p_g represent the probability of good split concerning u and g. The IRF classifies the data into three classes: normal, malicious, and suspicious. This is done by computing the similarity between the features of the incoming packet and the feature set of known attacks. From the classified packets, the malicious packets are denied, and suspicious packets are sent to the next phase of intrusion detection. The Pseudocode for the proposed IRF-based SIDS is provided in Algorithm **??**.

2) Anomaly-based intrusion detection: It is used to identify unknown attacks in the IoT environment. The suspicious packets are classified as normal or malicious using DCRNN by considering spatial and temporal features. The proposed neural

---

Algorithm 2: Improved Random Forest Algorithm

1 **Begin**
2 *Initialization:* Φ_0, Z_0, $\varsigma_0(\cdot)$, and number of passes n
3 Compute feature weight using Eq. (1) and perform feature ranking
4 Select h_0 features and group into Δ
5 Group the rest of the features g_0 into Δ'
6 **while** $g_n \geq f$ **do**
7   Compute α_n, β_n of features in Δ'_n
8   Compute Rf_n and remove it from Δ'_n
9   Perform Eq. (2) and select a set of features S_n in Δ'_n
10   Move S_n from Δ'_n to Δ_n
11   Compute $\varsigma_{n+1}(\cdot) = \varsigma_n(\cdot) - Rf_n$
12   Compute $\Delta_{n+1} = \Delta_n + S_n$ and $\Delta'_{n+1} = \Delta'_n + S_n$
13   Compute δh, δg using Eq. (3)
14   Perform Eq. (4) to determine the value of δZ
15   Compute $Z_{n+1} = Z_n + \delta Z$
16   Grow forest Φ_{n+1}, Z_{n+1}, and $\varsigma_{n+1}(\cdot)$
17      $n = n + 1$
18 **End while**
19 **End**

---

network possesses a series connection of a CNN and a Gated Recurrent Unit (GRU). The CNN acts as a feature extractor that extracts all the packets' features through the convolution and pooling layers. The extracted features are then provided to the GRU to analyze the temporal dependency between the

features to achieve improved classification accuracy. Initially, the packets' local features can be extracted and converted into feature vectors. These vectors are then rescaled from 0 to 255 to get converted into images of $U \times V$ pixels where U represent the number of columns, and V represents the number of rows, respectively. The CNN model utilized in our approach extracts the features from the images as formulated below:

$$I'_l = \frac{I_l - kr + 2Q}{SK} + 1 \tag{7}$$

where I_l represents the input image length, Q refers to the number of zeros, and Kr refers to the size of the kernel. The stride of the kernel is represented as SK. The output of the CNN is led to the GRU, which possesses two gates, namely, the update gate and the reset gate, that work as follows:

$$
\begin{aligned}
c_t &= \lambda \left(w_c \cdot [I_{t-1}, y_t] + y_t \right), \\
b_t &= \lambda \left(w_b \cdot [I_{t-1}, y_t] \right), \\
I'_t &= \tanh \left(w \cdot [b_t \cdot I_{t-1}, y_t] \right), \\
I_t &= (1 - c_t) \cdot I_{t-1} + c_t \cdot I'_t.
\end{aligned}
\tag{8}
$$

where

$$\lambda = \frac{1}{1 + e^{-t}} \tag{9}$$

$$\tanh(t) = \frac{1 - e^{-2t}}{1 + e^{-2t}} \tag{10}$$

where the input features are represented by y. The I_t, c_t represent the update parameters respectively. The weight function is represented as w, and the activation function utilized for the flow of sequences is represented as tanh and λ, respectively. The relationship between both spatial and temporal features is determined with the help of sequence vectors, which are fed into the fully connected and softmax layers for the classification of the packet. If the packet is classified as malicious, then it will be dropped, and the normal packets will be sent to the cloud. Fig. 3 shows the DCRNN-based detection of suspicious packets.

D. Trust Value Evaluation and Virtual Honeypot Deployment

In this section, we perform a honeypot-based attack prevention mechanism based on trust computation for efficient service provisioning in the network. Fig. 4 illustrates the trust-based migration of services and virtual honeypot deployment for intrusion prevention.

1) Trust value calculation: In this model, we proposed two types of edge servers: a global edge server and a local edge server. The global edge server has the responsibility to manage the local edge servers. The user service requests are collected by the local edge server. Then, we calculate the trust value for the edge server as follows:

$$\tau(LE) = In_\tau \pm (1 - In_\tau)X\vartheta \tag{11}$$

where, In_τ and ϑ denote the initial trust value and degree of change of trust value of the edge servers, respectively. Based on the change in the degree of trust value, the trust value will be dynamically increased or decreased.

2) Live migration: After calculating the trust value $\tau(LE)$ of the local edge servers, the global edge server computes the threshold value $\tau_{TH}(LE)$ based on the average of all trust values of the local edge servers. The nodes that possess $\tau(LE) \geq \tau_{TH}(LE)$ are classified as high-trusted local servers, denoted as $H(\tau(LE))$, while nodes with $\tau(LE) < \tau_{TH}(LE)$ are classified as low-trusted local edge servers, denoted as $L(\tau(LE))$. The $L(\tau(LE))$ nodes are vulnerable to attacks, and to ensure uninterrupted service to users, we implement live migration using a moving target defence strategy. This approach enables the migration of running services from $L(\tau(LE))$ to a suitable $H(\tau(LE))$. The selected $H(\tau(LE))$ must have a system configuration similar to that of $L(\tau(LE))$. To determine an appropriate $H(\tau(LE))$ with the same system configuration as $L(\tau(LE))$, we propose a Heap-Based Optimization (HBO) method that considers CPU resource availability, trust value, and system load. The definitions of the input parameters are as follows:

- CPU Resource availability: CPU resource availability RA_{CPU} is defined as the summation of the number of currently running tasks running on the edge server, and the tasks are provided to the available server based on the user request. It is denoted as:

$$RA_{CPU} = \sum_{i=1}^{n} E_i T_i \tag{12}$$

- Load: The load (L) represents the number of user tasks waiting (T_w) to be processed by the edge server, relative to the available CPU resources (RA_{CPU}).

$$L = \frac{RA_{CPU}}{T_w} \tag{13}$$

Based on the above input descriptions, Heap-Based Optimization (HBO) is performed. The HBO is an optimization algorithm for finding the optimal path shown in Algorithm **??**. Here, the objective of HBO is to find the H ($\tau(LE)$) based on maximum CPU resource, maximum trust value and minimum load. Construction of an optimal heap needs a Search Agent (SA), which includes Fitness Value (FV), which is the heap nodes' key, and index (I), which has the heap nodes' value. The SA is calculated based on the no. of LEs which are present in the IoT environment, and it is denoted as:

$$
\begin{bmatrix}
SA_1^{DIM} & SA_2^{DIM} & \cdots & SA_M^{DIM} \\
\vdots & \vdots & \ddots & \vdots \\
SA_N^{DIM} & SA_{N-1}^{DIM} & \cdots & SA_M^{DIM}
\end{bmatrix}
\tag{14}
$$

where N denotes the number of SA, and DIM denotes the corresponding dimensions. A FV is calculated based on RA_{CPU}, $\tau(LE)$, L for each LE servers which is denoted as:

$$FV = \{FV(LE_1), FV(LE_2), \cdots, FV(LE_n)\} \tag{15}$$

Based on the number of search agents (SA), the fitness values (FV), and the corresponding index values (I) are calculated. The index (I) represents the value of the heap node based on CPU resource availability (RA_{CPU}), trust level
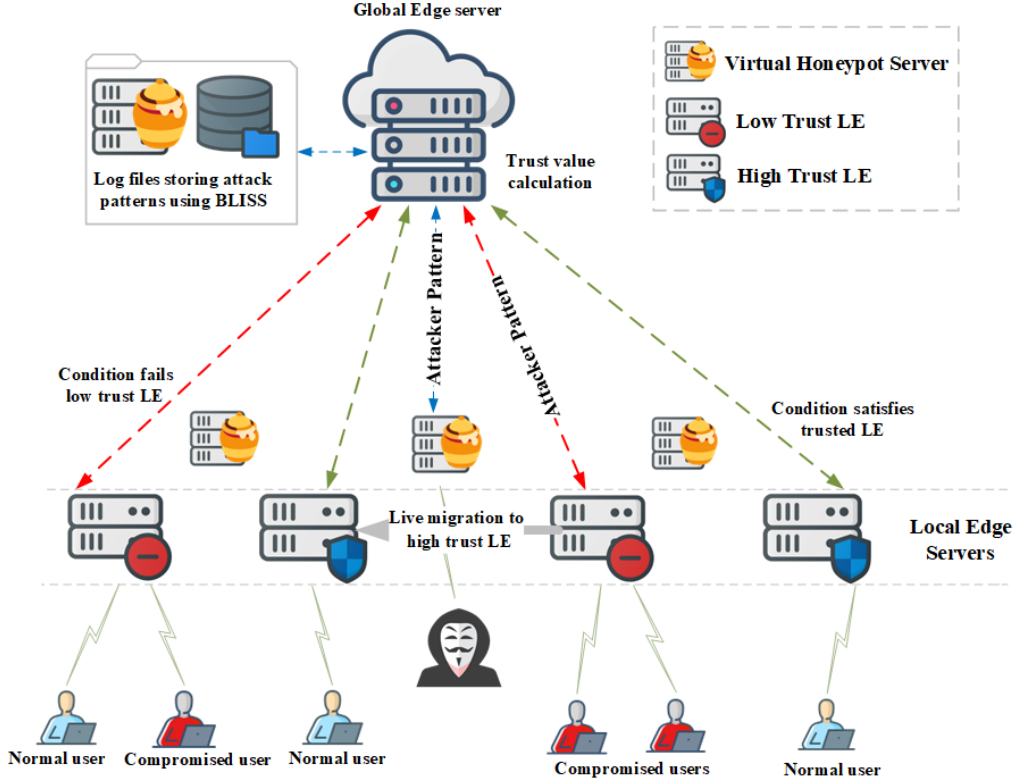
Fig. 4: Virtual Honeypot Deployment and Live Migration

$(\tau(LE))$, and system load (L). The average fitness value (AVG_{FV}) is computed based on the objective conditions. If any node satisfies all the objective conditions during the computation of AVG_{FV}, it is assigned as the root node (R_n) at depth level $d = 0$. Subsequently, for corresponding depth levels $d = 1, 2, 3, \ldots, n$, branch nodes (B_n) are assigned dynamically based on system conditions (i.e., trust level, CPU availability, and load). For root node calculation, we have:

$$R_n = \left\{ \text{high } RA_{\text{CPU}}, \quad \text{if } d = 0 \right. \tag{16}$$

For branch node allocation at increasing depths:

$$B_n(d) = \begin{cases} H(\tau(LE)) \geq \tau_{TH}(LE), & d = 1 \\ L_{\text{low}}, & d = 2 \\ B_n(d-1) > B_n(d), & d = 3, 4, \ldots, n \end{cases} \tag{17}$$

For corresponding depth levels $d = 1, 2, 3, \ldots, n$, branch nodes (B_n) are assigned progressively in a hierarchical manner according to trust level, load, and CPU availability. The root node (R_n) is initialized at $d = 0$, and as depth increases, branch nodes are dynamically selected based on system conditions. To maintain hierarchical ordering:

$$B_n(d) < R_n, \quad \text{for all} \quad d \geq 1 \tag{18}$$

For building a heap R_n branch nodes based on d levels are denoted as:

$$H(LE) = \begin{cases} R_n \geq highRA_{CPU}, & \tau(LE), L, d = 0 \\ B_n(1) < R_n, & d = 1 \\ B_n(n) < B_n(n-1), & d = n \end{cases} \tag{19}$$

After heap construction based on AVG_{FV}, the services from $L(\tau(LE))$ are migrated to heaps of $H(LE)$ based on heaps arranged hierarchically.

TABLE IV: SYSTEM CONFIGURATION

Hardware Specifications	**Hard Disk**	60 GB
	RAM	8 GB
	Processor	Intel (R) Core (TM) i5-4590S CPU @ 3.00 GHz
Software Specifications	**Operating System**	Ubuntu 14.04 LTS
	Network Simulator	NS3.26

3) Virtual Honeypot Deployment: During live migration, virtual honeypots with the same system configuration as the Local Edge Servers (LEs) are deployed to attract attackers and prevent the compromise of LEs. These honeypots detect attack signatures and store them in log files, which are then securely stored in the database. To enhance security, the Bimodal Lattice Signature Scheme (BLISS) algorithm is employed for encryption, as shown in Algorithm 4. BLISS is a digital signature scheme that uses a public key for verification and a private key for signature generation. The primary objective

---

Algorithm 3: Heap-based Optimization

Input: RA_{CPU}, $\tau(LE)$, L

Result: R_n, high RA_{CPU}, $H(\tau(LE)) \geq \tau_{TH}(LE)$, low L

1 Begin

2 Calculate heap nodes based on RA_{CPU}, $\tau(LE)$, and L; Compute average fitness value AVG_{FV} based on objectives;

3 foreach *node in nodes* **do**

4   **if** *node satisfies all objective conditions for* AVG_{FV} **then**

5   Assign node as Root node (R_n) at depth level $d = 0$;

6   Assign branch nodes B_n dynamically for depth levels $d = 1, 2, 3, ..., n$ based on trust, load, CPU;

7   **foreach** *transaction request* **do**

8   **if** *timestamp is valid* **then**

9   Validate request via consensus; **if** *consensus result = valid* **then**

10   Accept transaction;

11   **else**

12   Drop transaction (illegitimate request);

13   **else**

14   Drop transaction (invalid timestamp);

15 End

Algorithm 4: BLISS Signature Generation

Input: Attack pattern w, Public key $G \in \mathfrak{D}_{2p}^{n \times m}$, Private key $H \in \mathfrak{D}_{2p}^{n \times m}$, Security parameter $\sigma \in \mathbb{R}$

Result: Signature (S, F) of the packet pattern w

1 Begin

2 $d \leftarrow$ Sample vector from Gaussian distribution $\mathcal{N}(0, \sigma^2 I_n)$

3 $F \leftarrow H((G \cdot d) \mod 2p, w)$ // Compute hash-based challenge

4 Randomly select bit $re \in \{0, 1\}$

5 $S \leftarrow d + (-1)^{re} K_c$ // Compute signature response

6 Output (S, F) if acceptance criteria are satisfied

7 Store (S, F) securely encrypted in log file Lf

8 if *acceptance criteria are not satisfied* **then**

9   Restart signature generation

10 End

---

is to encrypt log files containing attack signatures before storing them in the database. During migration, attackers might attempt to compromise $H(\tau(LE))$. To counter this, virtual honeypots $(HP_1, HP_2, HP_3, \ldots, HP_n)$ — configured identically to the LEs — are deployed, misleading attackers into targeting the honeypots instead of the legitimate LEs. Once an attack is detected on a honeypot, its signature is recorded in a log file Lf and encrypted using the BLISS algorithm. The attack pattern, denoted as ω, captures various attacker behaviours targeting the honeypot. The verification function G and signature generation function H process the attack signature, where (S, F) represents the generated signature and the challenge vector for computing the hash function. The sample vector d is derived from an n-dimensional Gaussian distribution with a standard deviation σ, ensuring randomness in signature generation. A random bit $r_e \in \{0, 1\}$ is selected, and a shift vector K_c is applied to finalize the signature. The resulting S is encrypted using BLISS and securely stored in the log file.

$$F = H(\omega) \quad \text{// Compute hash of attack pattern}$$
$$d \sim \mathcal{N}_\sigma^n \quad \text{// Sample from Gaussian distribution}$$
$$r_e \in \{0, 1\} \quad \text{// Generate random bit} \quad \quad (20)$$
$$S = d + r_e K_c \quad \text{// Generate attack signature}$$
$$Lf = \text{Encrypt}(S, F) \quad \text{// Encrypt and store in log file}$$

VI. RESULTS AND ANALYSIS

This section deals with the simulation of the proposed approach. The performance of this approach is validated by comparing it with existing approaches. This section is divided into four subsections, namely simulation setup, dataset description, comparative analysis, and research summary, which are presented below.

TABLE V: SIMULATION CONFIGURATION

Network Parameters	
Area of simulation	900×1200
Number of IoT user nodes	50
Number of IoT device nodes	50
Number of edge gateways	6
Number of cloud server	1
Time for simulation	200s
Energy at initial stage	40J
Modules	IPV4, IPV6, MAC
Number of malicious nodes	10-30 nodes
Number of 5G base station	1
Packet Transmission parameters	
Packets data rate	500Mbps
Packets interval	$2^6, 2^7, 2^8, 2^9, 2^{10}$ bytes
Number of packets	1250
Retransmission rate	10
Protocol used	UDP
Network Traffic Parameters	
BW of the channel	100KHz
Queue used	First In, First Out
Traffic type	UDP, TCP, CBR
Security Parameters	
Attack probability ratio	1:5
Attacks interval	2-6 p/sec
Attacks detected	93%
Number of attacks	~ 8
Frequency of attacks	15-30p/sec
Mobility parameters	
Country Name or Area Name	ISO ALPHA 2 Code
Transmission range	250m
Model of mobility	Random waypoint model
Nodes mobility	15.3m/s

A. Simulation Setup

The simulation tool used for the implementation of the proposed approach is NS3.26, in which IDS detection, AIDS detection, virtual honeypot deployment and attack signature pattern are successfully verified. Table IV shows the system configuration for attaining the simulation. Table V shows the simulation configuration for the proposed approach. In our proposed approach, the dataset used is CICIDS-2017 for designing an effective attack detection system, which is specially designed for IDS and AIDS. This dataset contains current and old benign (normal traffic) and attacks (anomaly traffic) such as DoS, DDoS, Web-based, Heart bleed, Infiltration, Scan and Bot. Based on parameters such as source and destination ports, source and destination IPs and time stamps, network traffic was analyzed using a CIC flow meter and labelled in a CSV file. At that time of implementation, data can be separated into training data and test data. The training data contains 157,722 packets, among which 137,626 are labelled as normal (benign) and 20,076 are labelled as attacks. The test data contains 66,834 packets. Among that, 54,284 are labelled as normal,
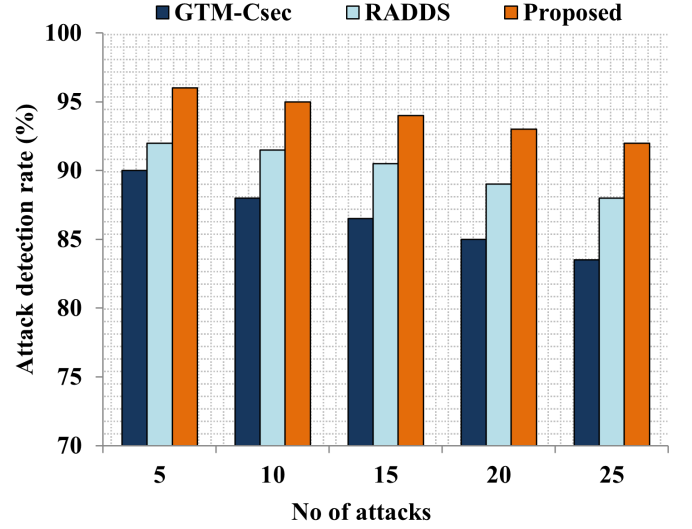

Fig. 5: Number of attacks vs. attack detection rate

and 12,540 are labelled as attacks. The records of the attacks are stored in a CSV file. The features of the dataset are shown in Table V.

B. Comparative analysis

The validation of the proposed approach is presented in this sub-section, in which our approach is compared with the existing approaches such as RADDS [39] and GTM-Csec [40] by means of several performance metrics such as accuracy, attack detection rate, false negative rate, precision, recall, ROC curve, memory usage, CPU usage, and execution time.

1) Impact of attack detection rate: The attack detection rate of any system is defined as the rate of detection of attack packets from the overall packets. It must be high so that the system is secure in any environment; if any system has a low attack detection rate, it will lead to severe security threats.

Compared to the existing works in Fig. 5, the attack detection rate of the proposed approach is 96.5%, even though the attacks increased. The deployment of a virtual honeypot collects the pattern of attacks and stores it in the log in the database in an encrypted manner. These attack patterns are further used to train the SIDS model, which helps to detect the attacks earlier, while the existing works use VM techniques and physical honey pot deployment, which have an attack detection rate of 92% and 89.7%, and do not focus on the collection of attack patterns. The existing approaches utilized game-theoretic approaches in which they assume that the attack strategies of the attacker are known to the IDS, but in reality, this is not so; this leads to a low attack detection rate. The proposed approach outperforms existing approaches with an increased attack detection rate of 4.5% to 7.2%.

2) Impact of accuracy: Accuracy is termed the capability of a system to determine the type of attack. The accuracy should be high for any system in terms of attack detection; low accuracy leads to increased risk in the system's environment and causes security threats. The detection accuracy of our proposed model, shown in Fig. 6, is 97%, which is more efficient
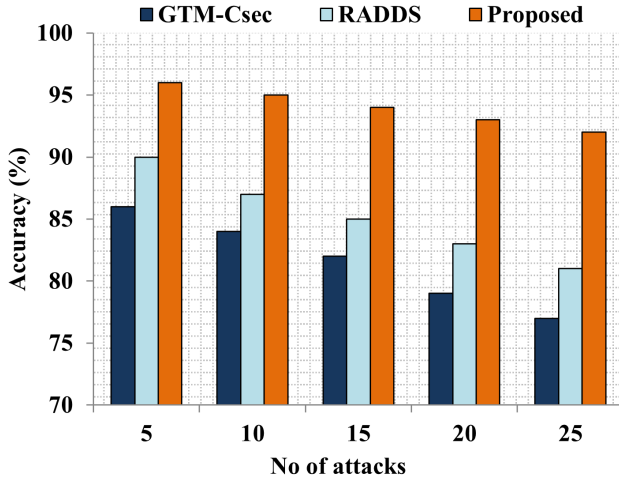
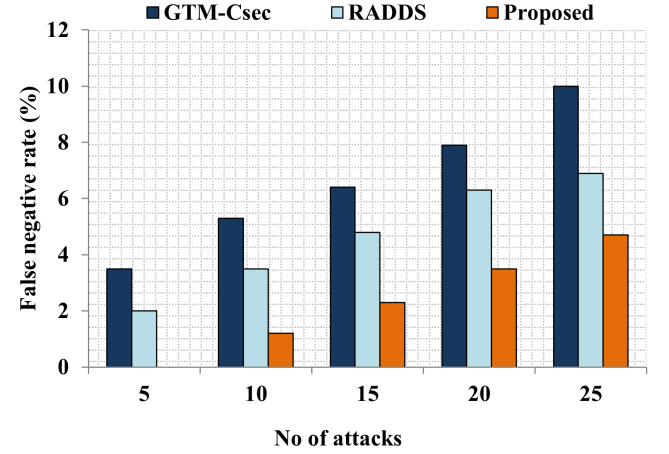Fig. 6: Number of Attacks vs. Accuracy

Fig. 7: Number of Attacks vs. False Negative Rate

than the existing works; however, the no. of attacks increases. The DCRNN-based detection of anomalies based on both spatial and temporal features increases the accuracy, while the existing works use a Game-theoretic model for security, which has an accuracy of 89% and 84%, and does not focus on the attack pattern collection, which reduces the overall accuracy of these approaches. The proposed model outperforms the existing approaches with an increased accuracy rate of 8% to 13%. The False negative rate comparison of the existing and proposed model is shown in Fig. 7. Whenever the number of attacks increased, the false negative rate also increased. The false negative rate is just 4.5% in our proposed method for 25 attacks because of the integrated action of the SIDS and AIDS detection process, which accurately detects both the known and unknown attacks. The existing approaches do not consider the unknown attack strategies, which results in a false negative rate of 7.6% and 9.7%, and degrades the performance. The proposed approach outperforms existing approaches with a decreased false negative rate of 3.1% to 5.2%.

3) Impact of false negative rate: The false negative rate is defined as the number of attacks that were falsely determined; a system is a good system when it has a low false negative rate.

4) Impact of precision: Precision is defined as the trueness of a system; if a system has a high trueness value, it has a high precision rate and increased accuracy. Fig. 8 shows the comparison of the precision of the proposed method and existing approaches with respect to a number of attacks. The precision of the proposed approach is 97%, whereas the existing works have precision rates of 92.7% and 90%, respectively. The trueness of the system in the proposed method is improved by IDS techniques and virtual honeypot deployment for attack pattern collection, while the existing works assumed knowledge of the attack patterns, which decreases the precision rate. The proposed approach outperforms existing approaches with an increased precision rate of 5.2% to 7%.

5) Impact of recall: Recall rate is defined as the detection of falseness of the system with high accuracy; if a system has a high recall rate, it would have high overall accuracy. The
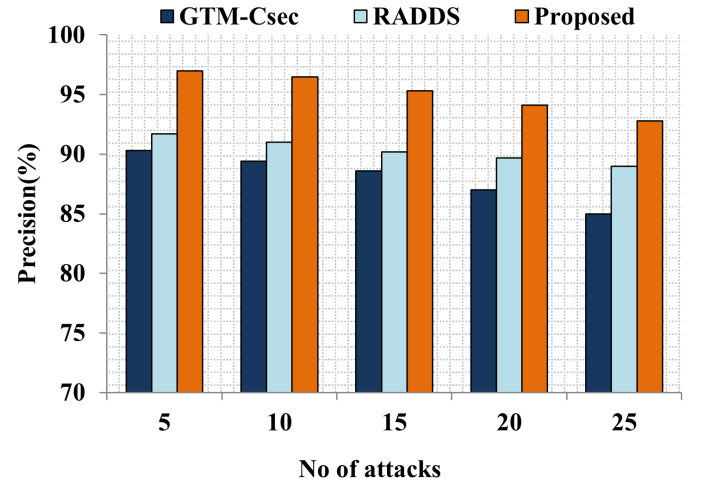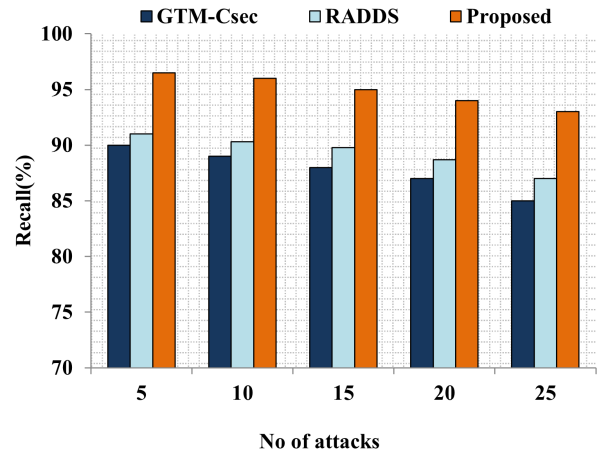

Fig. 8: Number of Attacks vs. Precision
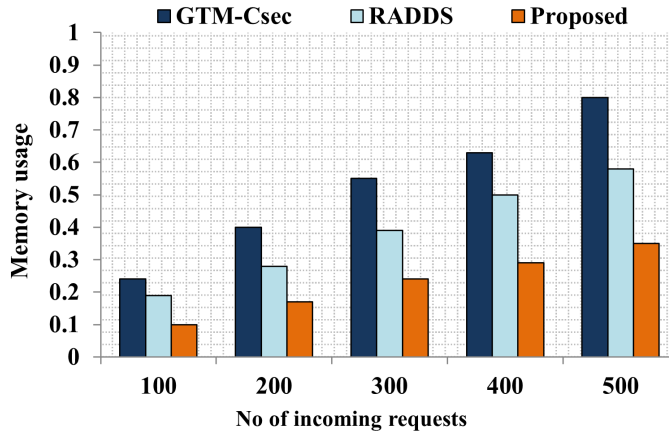

Fig. 9: Number of Attacks vs. Recall

Fig. 10: Number of incoming requests vs. memory usage.

Fig. 11: Number of incoming requests vs. CPU usage.

Fig. 12: Number of incoming requests vs. execution time.

comparison of the proposed method, which has a recall rate of 96%, and existing works, which have a recall rate of 92% and 90.5%, respectively, with respect to the number of attacks, is shown in Fig. 9. The high recall rate is due to the improved detection techniques, which consider both IDS and anomaly IDS in our proposed method, while the existing works did not focus on two types of IDS, leading to a lower recall rate and less overall accuracy. The proposed work improves the recall rate of 4% to 5.5% in existing works.

6) Impact of memory usage: Memory usage depends on the amount of memory used for a task to complete. If more memory is utilized, it leads to high energy consumption and overload to the system. The memory usage comparison of the proposed method and existing works is shown in Fig. 10. Whenever the number of user requests increases, memory usage increases. The memory usage of the proposed approach is 0.35 due to the migration of services from less trusted to highly trusted by considering the load, which reduces the overhead and energy consumption, while the other existing works have high memory usage of 0.8 and 0.56. This is due to the lack of focus on the load, which increases the memory usage and overhead. The proposed approach outperforms existing approaches with decreased memory usage of 0.45 to 0.21.

7) Impact of CPU usage: CPU usage is defined as the amount of resources consumed to complete tasks; an efficient system must have low CPU usage and complete more tasks. If a system uses more CPU, it would be termed as an inefficient system and would degrade the overall performance of that system. Fig. 11 shows a comparison of the CPU usage to 6.7% with respect to the number of incoming requests with state-of-the-art works. The results show that the CPU usage of the proposed method is less with an increase in requests; however, the existing works have 12% and 15.6%, which did not consider the decentralized edge nodes for management, leading to high CPU usage and high resource consumption. The proposed work reduces the CPU usage to 5.5% and 8.8% from existing works.

8) Impact of execution time: Execution time must be less for any system to complete a task, which reduces the time complexity and increases the efficiency; if a system has a
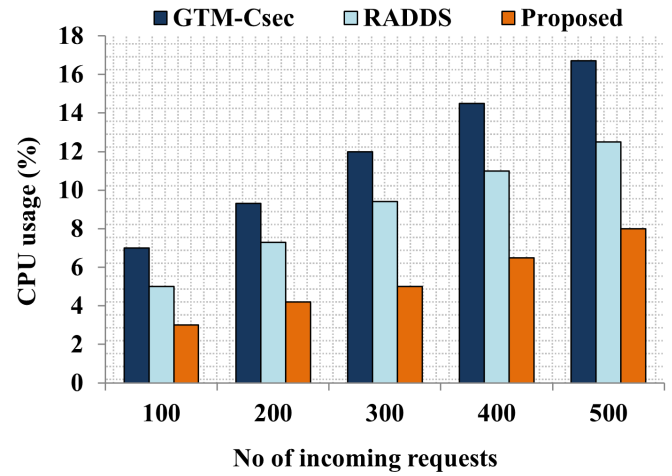
high execution time, it will lead to high time complexity and performance degradation. A comparison of the proposed method and existing works is shown in Fig. 12 with respect to the number of incoming requests and execution time. The results show that the execution time of any task is 2.5s even though the incoming requests are increased. This is due to the use of edge computing servers, while the existing works have 9.2s and 14.2s and do not consider the edge servers for network management, which increases the execution time and increases latency. Our approach outperforms existing approaches with a low execution time of 12.3s to 7.3s.

9) Impact of ROC curve: The Receiver Operating Characteristics (ROC) curve is the classification of the true positive rate and false positive rate of any system at different thresholds. Fig. 13 shows that the ROC comparison of existing methods and the proposed method has reduced the false positive rate and increased the true positive rate than existing works because of the collection of attack patterns, resulting in an increased detection rate. The existing works assume that the attack strategies of the attacker are known to the IDS, and unknown IDS are not considered during authentication, which increases the false positive rate and decreases the true positive rate.
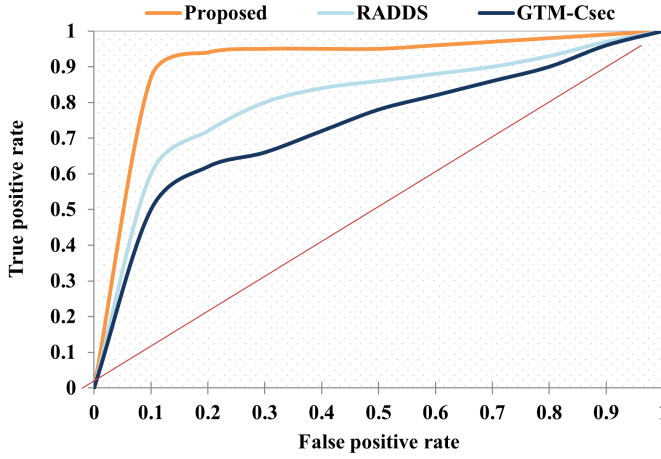
Fig. 13: ROC curve.

10) Security Analysis: The security of service provisioning in the 5G-IoT environment is hindered by various cyberattacks prevailing in the network. These attacks pose a serious threat to the legitimacy and integrity of the entities in the network. The following are a few of the serious attacks mitigated by the proposed approach.

- Fuzzing attacks – In this type of attack, the attacker attempts to steal the credentials of the devices and users in the network. The proposed approach mitigates these attacks by utilizing blockchain technology in which the credentials are stored in a hashed format. The DAA strengthens the mitigation of fuzzing attacks by employing a tweakable block cipher (Deoxys-TBC) to encrypt and authenticate credentials before they are stored on the blockchain. Unlike conventional hashing techniques, Deoxys-TBC provides enhanced security by introducing tweak-based encryption, ensuring that each authentication request is uniquely processed with a cryptographic nonce, making brute-force credential guessing infeasible. Additionally, mutual authentication mechanisms verify both the requester and the verifier, preventing unauthorized access to stored credentials.
- DDoS attacks – The DDoS attackers forward a massive number of packets or user requests with the intention of shutting down edge gateways or the cloud server. The proposed approach performs integrated IDS to determine the nature of the packets, and on the user side, the blockchain verifies the timestamp of the requests. Additionally, the system employs a rate-limiting mechanism and anomaly-based traffic filtering to detect and mitigate high-volume malicious requests. The IDS utilizes machine learning-based pattern recognition to distinguish between legitimate and malicious traffic, reducing false positives while ensuring uninterrupted service availability. Furthermore, blockchain's decentralized validation prevents single points of failure, as distributed nodes collaboratively verify request authenticity, making it significantly harder for attackers to overwhelm the system.
- Impersonation attack – The attacker attempts to imitate

a legitimate user to get into the network to perform various attacks. This attack is mitigated by performing blockchain-based authentication in which the tag generated by the authentication algorithm is verified. To further strengthen security, multi-factor authentication (MFA) can be integrated as an additional layer of defence. By combining blockchain authentication with a secondary verification factor—such as a one-time password (OTP) or biometric authentication—unauthorized access attempts can be significantly reduced. Implementing MFA alongside the existing authentication mechanism would ensure that even if one credential is compromised, attackers cannot easily impersonate legitimate users.

- Replay attack – The attacker injects previously captured legitimate packets into the network at regular intervals to exhaust system resources and disrupt normal operations. The proposed approach mitigates these attacks by considering spatio-temporal features for attack detection, which analyze the timing and spatial distribution of incoming packets to distinguish between legitimate transmissions and replayed attacks. By leveraging historical traffic patterns and anomaly detection, the system effectively flags repeated identical requests, preventing attackers from overwhelming network resources. Additionally, blockchain's timestamping mechanism ensures the uniqueness of transactions, making it difficult for attackers to reuse previously recorded packets without detection.

11) Analysis Summary: In this section, the discussion about the performance of the proposed approach is presented in an elaborative manner. The numerical analysis of the performance of the proposed approach is provided in the table. From the table, it is clear that the proposed approach possesses an increased attack detection rate and accuracy; this is due to the integration of IRF-based SIDS and DCRNN-based AIDS, considering both attack patterns and spatiotemporal features of packets. The updation of attack strategies for the newly caused attacks with the help of the deployment of virtual honeypot nodes also contributes to the increased attack detection rate. The reduced false negative rate and increased precision and recall possessed by the proposed approach are due to the working of the intrusion detection system, in which the ranking of features is performed in SIDS to classify the packets based on important features in the attack patterns. The temporal dependencies between the features possessed by the suspicious packets are analyzed to precisely determine the packets. The overall reduction in memory usage, CPU usage and execution time is due to the adoption of edge-based architecture, in which the service migration is carried out based on configuration, resource availability, trust and load of the edge servers. Further, blockchain-based authentication ensures the legitimacy of the entities in the network. From the above discussion, it is clear that the proposed approach performs more effectively than the existing approaches in terms of all the metrics considered, providing security. Table **??** shows the performance metrics comparison of existing and proposed approaches.

VII. CONCLUSION AND FUTURE DIRECTIONS

The rapid expansion of the IoT within NGWN has introduced significant security challenges. In this paper, we proposed a trust-aware security framework that integrates blockchain authentication, advanced intrusion detection, and dynamic honeypot deception to enhance security in NGWN-enabled IoT environments. Our framework effectively mitigates cybersecurity threats by ensuring robust authentication, reducing false positives in intrusion detection, and dynamically adapting to evolving attack strategies. The proposed system leverages blockchain technology for decentralized and tamper-proof authentication, preventing impersonation and unauthorized access. The integration of an IRF-based SIDS and a DCRNN-based AIDS enables precise and efficient threat detection. Furthermore, a moving target defence strategy dynamically migrates services to trusted edge nodes, mitigating the risk of service compromise. The deployment of high-interaction, on-demand virtual honeypots allows for proactive deception and real-time attack pattern analysis, significantly improving intrusion detection and response mechanisms. Performance evaluation in the NS3 simulation environment demonstrates the superiority of our approach over existing methods. The proposed framework achieves a 25% improvement in detection accuracy, a 30% reduction in false negatives, and enhanced resource efficiency compared to conventional security mechanisms. These results confirm the effectiveness of our approach in securing IoT environments while maintaining low computational overhead. Despite its advancements, this research opens avenues for further exploration. Future work will focus on integrating Explainable AI (XAI) techniques to improve transparency and interpretability in intrusion detection decisions. Incorporating XAI, security professionals and system administrators will gain deeper insights into the decision-making process of the detection models, ensuring better trust, usability, and adaptability of the security framework.

REFERENCES

[1] Y. Zhang, G. Chen, H. Du, X. Yuan, M. Kadoch, and M. Cheriet, "Real-time remote health monitoring system driven by 5g mec-iot," *Electronics*, vol. 9, no. 11, p. 1753, 2020.

[2] M. Wazid, A. K. Das, S. Shetty, P. Gope, and J. J. Rodrigues, "Security in 5g-enabled internet of things communication: issues, challenges, and future research roadmap," *IEEE Access*, vol. 9, pp. 4466–4489, 2020.

[3] V. Nilkanthsing, "Dynamic orchestration of security services at fog nodes for 5g iot," in *IEEE International Conference on Communication*, 2020.

[4] P. Aravamudhan and T. Kanimozhi, "A survey on intrusion detection system and prerequisite demands in IoT networks," in *Journal of Physics: Conference Series*, vol. 1916, no. 1. IOP Publishing, 2021, p. 012179.

[5] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for iot security based on learning techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019.

[6] Q. A. Al-Haija and A. Droos, "A comprehensive survey on deep learning-based intrusion detection systems in internet of things (iot)," *Expert Systems*, vol. 42, no. 2, p. e13726, 2025.

[7] A. Hinojosa and N. E. Majd, "Edge computing network intrusion detection system in iot using deep learning," in *2024 33rd International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2024, pp. 1–6.

[8] A. A. Toony, F. Alqahtani, Y. Alginahi, and W. Said, "Multi-block: A novel ml-based intrusion detection framework for sdn-enabled iot networks using new pyramidal structure," *Internet of Things*, vol. 26, p. 101231, 2024.

[9] Y. K. Saheed, O. H. Abdulganiyu, and T. A. Tchakoucht, "Modified genetic algorithm and fine-tuned long short-term memory network for intrusion detection in the internet of things networks with edge capabilities," *Applied Soft Computing*, vol. 155, p. 111434, 2024.

[10] V. Maurya, V. Rishiwal, M. Yadav, M. Shiblee, P. Yadav, U. Agarwal, and R. Chaudhry, "Blockchain-driven security for iot networks: State-of-the-art, challenges and future directions," *Peer-to-Peer Networking and Applications*, vol. 18, no. 1, pp. 1–35, 2025.

[11] Y. Otoum, P. Singh, and A. Nayak, "Advancing iomt defenses: Deep collaborative learning for robust healthcare security," in *GLOBECOM 2024-2024 IEEE Global Communications Conference*. IEEE, 2024, pp. 2966–2971.

[12] N. A. Dawit, S. S. Mathew, and K. Hayawi, "Suitability of blockchain for collaborative intrusion detection systems," in *2020 12th Annual Undergraduate Research Conference on Applied Computing (URC)*. IEEE, 2020, pp. 1–6.

[13] F. Louati and F. B. Ktata, "A deep learning-based multi-agent system for intrusion detection," *SN Applied Sciences*, vol. 2, no. 4, p. 675, 2020.

[14] J. Franco, A. Aris, B. Canberk, and A. S. Uluagac, "A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2351–2383, 2021.

[15] Z. A. Khan and U. Abbasi, "Reputation management using honeypots for intrusion detection in the internet of things," *Electronics*, vol. 9, no. 3, p. 415, 2020.

[16] T. Zheng, Y. Du, K. Hua, X. Wu, S. Yuan, X. Wang, Q. Chen, and J. Tan, "Predictive analytics for cyber-attack timing in power internet of things: A flipit game-theoretic approach," *Internet of Things*, p. 101522, 2025.

[17] Y. Otoum, C. Hu, E. H. Said, and A. Nayak, "Enhancing heart disease prediction with federated learning and blockchain integration," *Future Internet*, vol. 16, no. 10, p. 372, 2024.

[18] R. Parsamehr, A. Esfahani, G. Mantas, A. Radwan, S. Mumtaz, J. Rodriguez, and J.-F. Martínez-Ortega, "A novel intrusion detection and prevention scheme for network coding-enabled mobile small cells," *IEEE Transactions on Computational Social Systems*, vol. 6,

no. 6, pp. 1467–1477, 2019.

[19] C. Liang, B. Shanmugam, S. Azam, A. Karim, A. Islam, M. Zamani, S. Kavianpour, and N. B. Idris, "Intrusion detection system for the internet of things based on blockchain and multi-agent systems," *Electronics*, vol. 9, no. 7, p. 1120, 2020.

[20] A. S. Mamolar, P. Salvá-García, E. Chirivella-Perez, Z. Pervez, J. M. A. Calero, and Q. Wang, "Autonomic protection of multi-tenant 5g mobile networks against udp flooding ddos attacks," *Journal of Network and Computer Applications*, vol. 145, p. 102416, 2019.

[21] M. Rajkumar, J. Karthika *et al.*, "Multi-view consistent generative adversarial network for enhancing intrusion detection with prevention systems in mobile ad hoc networks against security attacks," *Computers & Security*, vol. 150, p. 104242, 2025.

[22] Y. Al-Hadhrami and F. K. Hussain, "Real time dataset generation framework for intrusion detection systems in iot," *Future Generation Computer Systems*, vol. 108, pp. 414–423, 2020.

[23] G. Hatzivasilis, O. Soultatos, P. Chatziadam, K. Fysarakis, I. Askoxylakis, S. Ioannidis, G. Alexandris, V. Katos, and G. Spanoudakis, "Wardog: Awareness detection watchdog for botnet infection on the host device," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 1, pp. 4–18, 2019.

[24] W. Wang, P. Xu, and L. T. Yang, "Secure data collection, storage and access in cloud-assisted iot," *IEEE cloud computing*, vol. 5, no. 4, pp. 77–88, 2018.

[25] L. F. Maimó, Á. L. P. Gómez, F. J. G. Clemente, M. G. Pérez, and G. M. Pérez, "A self-adaptive deep learning-based system for anomaly detection in 5g networks," *Ieee Access*, vol. 6, pp. 7700–7712, 2018.

[26] J. Lam and R. Abbas, "Machine learning based anomaly detection for 5g networks," *arXiv preprint arXiv:2003.03474*, 2020.

[27] A. Yang, Y. Zhuansun, C. Liu, J. Li, and C. Zhang, "Design of intrusion detection system for internet of things based on improved bp neural network," *Ieee Access*, vol. 7, pp. 106 043–106 052, 2019.

[28] A. Mondal and R. T. Goswami, "Enhanced honeypot cryptographic scheme and privacy preservation for an effective prediction in cloud security," *Microprocessors and Microsystems*, vol. 81, p. 103719, 2021.

[29] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban ids: An intelligent anomaly-based intrusion detection system for iot edge devices," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6882–6897, 2020.

[30] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "Deepfed: Federated deep learning for intrusion detection in industrial cyber–physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5615–5624, 2020.

[31] M. A. Mahdi, "Secure and efficient iot networks: An ai and ml-based intrusion detection system," in *2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT)*. IEEE, 2024, pp. 1–6.

[32] S. K. R. Mallidi and R. R. Ramisetty, "Advancements in training and deployment strategies for ai-based intrusion detection systems in iot: a systematic literature review," *Discover Internet of Things*, vol. 5, no. 1, p. 8, 2025.

[33] H. Al-Mohannadi, I. Awan, and J. Al Hamar, "Analysis of adversary activities using cloud-based web services to enhance cyber threat intelligence," *Service Oriented Computing and Applications*, vol. 14, no. 3, pp. 175–187, 2020.

[34] J. Lee, J. Choi, G. Lee, S.-W. Shim, and T. Kim, "Phantomfs: File-based deception technology for thwarting malicious users," *IEEE Access*, vol. 8, pp. 32 203–32 214, 2020.

[35] B. Li, Y. Xiao, Y. Shi, Q. Kong, Y. Wu, and H. Bao, "Anti-honeypot enabled optimal attack strategy for industrial cyber-physical systems," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 250–261, 2020.

[36] N. Dara, P. Shankar, P. V. Arvind, and V. Singh, "Intelligent insight into iot threats: Leveraging advanced analytics with honeypots for anomaly detection," in *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*. IEEE, 2024, pp. 1–6.

[37] E. Ntizikira, L. Wang, J. Chen, and K. Saleem, "Honeyblock: Edge assisted ensemble learning model for intrusion detection and prevention using defense mechanism in iot," *Computer Communications*, vol. 214, pp. 1–17, 2024.

[38] S. Jangirala, A. K. Das, and A. V. Vasilakos, "Designing secure lightweight blockchain-enabled rfid-based authentication protocol for supply chains in 5g mobile edge computing environment," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 7081–7093, 2019.

[39] O. A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, "Resource-aware detection and defense system against multi-type attacks in the cloud: Repeated bayesian stackelberg game," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 605–622, 2019.

[40] K. S. Gill, S. Saxena, and A. Sharma, "Gtm-csec: Game theoretic model for cloud security based on ids and honeypot," *Computers & Security*, vol. 92, p. 101732, 2020.

[41] S. S. Chakkaravarthy, D. Sangeetha, M. V. Cruz, V. Vaidehi, and B. Raman, "Design of intrusion detection honeypot using social leopard algorithm to detect iot ransomware attacks," *IEEE Access*, vol. 8, pp. 169 944–169 956, 2020.

[42] W. Zhang, B. Zhang, Y. Zhou, H. He, and Z. Ding, "An iot honeynet based on multiport honeypots for capturing iot attacks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3991–3999, 2019.