International Journal of Computing and Digital Systems
2025, VOL. 17, NO. 1, 1–12

http://dx.doi.org/10.12785/ijcds/1571001227

An Efficient Internet Of Things Based Intrusion Detection And Optimization Algorithm For Smart Networks

Poornima M[1,2], Anitha T N[2], Mallikarjunaswamy S[3] and Umashankar M L[4]

[1]*Department of Information Science and Engineering, SJB Institute of Technology, Bengaluru, Karnataka, India-560060*
[2]*Department of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India- 562157.*
[3]*Department of Electronics and Communication Engineering, JSS Academy of Technical Education, Bengaluru, Karnataka, India-560060.*
[4]*Department of Engineering and Technology, Wipro Ltd, Sarjapur Road, Bengaluru, Karnataka, India-560035.*

Received 25 February 2024, Revised 17 December 2024, Accepted 4 January 2025

Abstract: Smart cities in India have significantly benefited from the integration of Internet of Things (IoT) technologies, which enable advanced sensing, data communications, and automation within smart networks. However, this complex network of IoT devices also introduces challenges such as data loss, signal attenuation, and security vulnerabilities. To combat these issues, the Intrusion Detection and Optimization Algorithm (IoT-IDOA) has been developed, tailored specifically for IoT environments. This algorithm improves upon previous methods by identifying critical 74 nodes and optimizing routing paths to enhance communication efficiency while reducing power consumption. Simulation results show that IoT-IDOA outperforms traditional approaches like the Key Match Detection Algorithm (KMA) and Cluster-based Detection Algorithm (CBA), especially in preventing selective forwarding and sink node attacks, and in lowering power dissipation to 0.235%, 0.365%, and 0.65%, respectively. the IoT-IDOA plays a crucial role in the evolution of smart cities in India. It sets new standards in network security and energy usage, contributing to the development of more sustainable, accurate, feasible, less power dissipation, resilient, and intelligent urban ecosystems. This progress is vital for realizing the full potential of smart city initiatives, paving the way for a future where urban environments are not only smarter but also more secure and environmentally friendly.

Keywords: : Internet of Things (IoT), Key match detection algorithm (KAM), Cluster based detection algorithm (CBA), Smart networks, Intrusion detection system (IDS).

1. Introduction

A few decades ago, the word Internet Protocol (IP) was coined. These days, the most cutting-edge technology that employs several protocols is known as the Internet of Things (IoT). The World Wide Web The Internet of Things (IoT) is a hybrid platform that enables heterogeneous, resource-constrained devices to exchange data through Internet connectivity. 6LoWPAN [1], RPL [2], IEEE 802.15.4 [3], CoAP [4], and many more are all part of the Internet of Things (IoT). But as the number of intelligent devices grows and some of them become mobile, the Internet of Things (IoT) becomes vulnerable to a range of threats started by different types of intelligent gadgets. When it comes to computing resources, the majority of IoT devices are severely lacking. This includes things like low power, processing capability, storage, and resilience against loss of connection. In particular, the most serious routing attacks, such as sinkhole and selective forwarding, were made possible by these restrictions, leaving the IoT vulnerable to them. Protecting against every possible routing attack is really challenging. This is why we've been concentrating on sinkhole assaults and selective forwarding [5].

A Selective Forwarding attack occurs when an adversarial node compromises many nodes in the network and deliberately discards a predetermined amount of packets. The perpetrator of this attack chooses which packets to forward and which to discard at random [6]. It is possible to filter any protocol with this attack, but its main purpose is to interrupt routing pathways. As an example, a malicious actor may selectively forward all RPL control messages while discarding all other packets along the path. When combined with other assaults, such sinkhole attacks, this attack can cause serious problems. The goal of a sinkhole assault is to damage data reception at a collecting point by attracting the maximum amount of

E-mail address: mpoornimaurs@gmail.com, anithareddytn72@gmail.com, pruthvi.malli@gmail.com

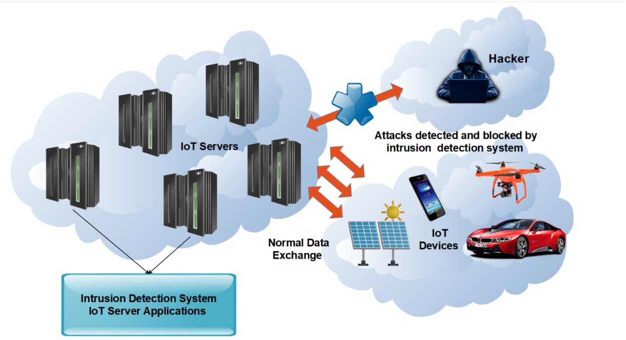

Hacker
Attacks detected and blocked by intrusion detection system
IoT Servers
Normal Data Exchange
IoT Devices
Intrusion Detection System IoT Server Applications

Figure 1. Fundamental structure of intrusion detection system (IDS) with IoT smart network

traffic to a certain place. The security and trustworthiness of the data transmitted by the devices are thus jeopardised.

Some attacks that try to disrupt the network can succeed even with message security, which provides authentication and encryption. This is why Intrusion Detection Systems (IDSs) are employed to protect networks from intrusions. Intrusion Detection Systems (IDS) monitor network traffic for signs of hostile activity or unauthorised users attempting to cause disruptions. Figure 1 illustrates the core functionality of an Intrusion Detection System (IDS) within an IoT network, highlighting the interaction between IoT devices, servers, and security mechanisms. IoT devices, such as connected vehicles, drones, and smart energy systems, engage in regular data exchanges, relaying information to and from IoT servers. These servers are the backbone of the network, responsible for processing, analyzing, and storing data, while also managing device directives and updates [7].

The IDS is integrated within this ecosystem, utilizing server applications to scrutinize network traffic continuously. Its primary role is to identify and differentiate between legitimate network activities and potential security threats
citebib8. When the system detects anomalous or suspicious behaviour indicative of a cyberattack, it reacts by blocking the intrusion, thereby safeguarding the network's integrity [8]. However, there exist several research gaps that need addressing to enhance the capabilities of IDS in IoT networks. These include improving detection accuracy to minimize false alarms, enabling real-time processing for instantaneous threat identification, and optimizing the IDS for the limited computational and energy resources typical of IoT devices. Additionally, the IDS must be adaptable to the evolving patterns of IoT traffic and scalable enough to manage the growth of IoT networks effectively. Further, the integration of sophisticated machine learning algorithms could significantly bolster the IDS's ability to detect novel and complex attack vectors. Ensuring cross-platform compatibility is also critical, given the diverse IoT protocols and standards in use. Lastly, the system must achieve a balance between effective monitoring and the preservation of user privacy. Advancements in these areas will be pivotal in advancing IDS technology, thus making IoT networks more secure and resilient to cyber threats. The innovative design of the Internet of Things (IoT) presupposes an always-on 6BR (IPv6 Border Router), mandates end-to-end message security [9], and assigns IP addresses to sensor nodes worldwide. Not only do these characteristics make IDS for the IoT difficult, but the things themselves are (i) accessible from anywhere in the world, (ii) limited in resources, and (iii) linked through lossy connections. Thus, it is worthwhile to offer an Intrusion Detection Technique for the Internet of Things that takes use of these possibilities and protects against these dangers. This is why we apply and test an intrusion detection method in an Internet of Things setting. Also, we have done our best to keep the network safe from intrusion.

Here is how the rest of the article is structured: A overview of the literature about some assaults and related works is included in section II. In section III, we go over the RPL protocol and how it forms the DODAG. Section IV details the steps taken and the outcomes, section V lays out the findings and suggests ways forward for the scientific community

A. *Related Work*

Internet of Things (IoT) sinkhole and selective forwarding attack detection has received little attention in the literature. To protect networks from selective forwarding attacks and identify topological disruptions, Wallgren et al. [10] suggested the Heartbeat self-healing protocol. One intrusion detection system (IDS) that can handle various threats, such as sinkhole and selective forwarding, is SVELTE, which is introduced in [7]. When it comes to routing threats, the centralised system specifies three modules: mapping (6Mapper), intrusion detection, and a mini-firewall. Attacks on the Routing Protocol for Low-Power and Lossy Networks (RPL) are detailed in [9]. They suggest using geographical clues to find hostile nodes that assault networks that use ETX. Semiauto should safeguard RPL-based network topologies by constructing an intrusion detection system (IDS) model according to the authors' recommendations in [10]-[11]. Analysing the trace file is the key idea for learning the states, transitions, and pertinent statistics. Their algorithm can identify RPL and topology Wormhole and selective forwarding attacks are examples of this. Zarpelao et al. [12] compiled an exhaustive list of intrusion detection systems developed specifically for the Internet of Things. Along with the benefits and drawbacks of each intrusion detection technique, they provided a number of them. ”Real-Time Intrusion Detection in IoT Networks Using Machine Learning” [13] This study presents a real-time intrusion detection system (IDS) for IoT networks that leverages machine learning algorithms. The system is trained to identify patterns and anomalies that signify potential threats, with a focus on minimizing latency in threat response. The machine learning model requires extensive training data, which may not be available for all types of intrusions, potentially reducing its effectiveness against novel attacks. ”Energy-Efficient Intrusion Detection and Resource Allocation for IoT” [14] This paper explores an algorithm that balances intrusion detection with energy efficiency. It aims to optimize resource allocation in IoT devices to extend their operational life while maintaining a high level of security. The optimization for energy efficiency may compromise the thoroughness of security surveillance, potentially leading to missed detections of low-activity intruders. ”Scalable Network Optimization for IoT Systems with Integrated Intrusion Detection” The research introduces a scalable network optimization framework that integrates with an IDS. It is designed to adapt to varying network sizes and loads, ensuring consistent performance. Scalability often comes at the complexity cost, and the paper lacks a detailed assessment of the overhead introduced by the scaling mechanisms.

”Cross-Layer Intrusion Detection System for IoT Networks” [15] This paper proposes a cross-layer IDS that monitors both the application and network layers, utilizing optimization algorithms to manage traffic and detect intrusions across different levels of the network stack. The cross-layer approach can lead to increased processing demands, making it less suitable for simpler, resource-constrained IoT devices. ”Adaptive Intrusion Detection Using Edge Computing in IoT” [16] The study focuses on an adaptive intrusion detection system that offloads computation to the edge of the network, closer to IoT devices, for faster response and optimization. Reliance on edge computing infrastructure can introduce new security vulnerabilities and requires a robust network of edge servers, which may not always be feasible. ”Blockchain-Based Secure Optimization for IoT Networks” [17] This paper details a novel approach to IoT network security and optimization using blockchain technology, ensuring integrity and traceability of operations within the network. Blockchain technology can introduce significant overhead and latency, which may not align well with the real-time operation requirements of certain IoT applications. ”Hybrid Intrusion Detection and Network Optimization for Heterogeneous IoT Environments” [18] The research introduces a hybrid approach that combines signature-based and behavior-based intrusion detection methods, coupled with network optimization for heterogeneous IoT environments. The hybrid nature of the system may result in a complex configuration process, and it may struggle to keep up with the continuously evolving IoT device landscape without constant updates. ”Quantum-Resistant Intrusion Detection Mechanisms for IoT Networks” [19] This paper explores the design of intrusion detection systems that are resistant to quantum computing-based attacks, focusing on cryptographic protocols that ensure long-term security for IoT networks. Quantum resistance often involves algorithms that are not yet widely adopted or tested, and the practicality of implementing such advanced solutions in current IoT infrastructures may be challenging. ”Deep Learning for Anomaly-Based Intrusion Detection in Industrial IoT” [20] The study presents a deep learning-based IDS specifically tailored for industrial IoT settings, using anomaly detection to identify potential threats in real-time within large-scale sensor networks. The IDS’s reliance on deep learning requires significant computational resources, which may not be readily available in all industrial environments. Additionally, the model may have a lengthy training period and require substantial data preprocessing. ”Multi-Agent Systems for Distributed Intrusion Detection in IoT” [21] This research proposes a multi-agent system framework where multiple IDS agents collaborate to detect and respond to intrusions in a distributed IoT network. Coordination among multiple agents can introduce complexity in the system, and ensuring that communication between agents is secure and efficient is a significant challenge that the paper does not fully address. ”Integrating IDS with IoT Device Management for Enhanced Security” [22] This paper examines the integration of intrusion detection systems with IoT device management platforms to create a cohesive security and management solution for IoT deployments. While integration offers a streamlined approach, it also creates a single point of failure. The paper does not sufficiently explore the implications of having both security and management coupled so tightly. ”Self-Learning Intrusion Detection for Autonomous IoT Networks” [23] The study introduces a self-learning IDS that adapts to the behavior of an autonomous IoT network, utilizing reinforcement learning to continuously improve its detection strategies. The self-learning aspect of the system depends heavily on the initial setup and the quality of the reinforcement signals. Incorrect or insufficient learning signals can lead to suboptimal detection performance. ”Cloud-Assisted Optimization for IoT Security” [24] This paper details the use of cloud computing resources to assist in the optimization of IoT security measures, including intrusion detection, through the power of distributed computing. The reliance on cloud computing raises concerns about data privacy and requires a dependable, high-bandwidth internet connection, which may not be available in all IoT applications, especially in remote areas. ”Federated Learning for Privacy-Preserving Intrusion Detection in IoT” [25] The research investigates the application of federated learning to create a privacy-preserving, decentralized IDS for IoT networks, where models are trained locally on devices without sharing sensitive data. Federated learning models can be less accurate than centralized models due to the decentralized nature of data processing, and synchronization between devices can be a complex process to manage effectively.

2. Intrusion Detection In The Internet Of Thing For Smart Network

The purpose of detecting routing attacks such as sinkholes and selective forwarding, we have integrated two detection methods. You can find both methods in MATLAB. On the other hand, our method can be expanded to detect more attacks. The following sections provide examples of the techniques that have been suggested [26].

A. Key Match detection Algorithm (KMA)

The Key Match Detection Algorithm (KMA) represents an innovative approach within the realm of IoT-based smart networks. It is a novel algorithm designed to swiftly identify and authenticate critical data packets essential for maintaining the integrity of communications. By utilizing a set of unique identifiers known as 'key signatures,' KMA rapidly scans incoming data against a secure database of trusted signatures. When a match is found, the data is deemed authentic and allowed to proceed; if not, it is flagged for further inspection or discarded, depending on the security protocols in place. This process is particularly crucial for IoT devices, which often operate autonomously and require immediate verification to respond to real-time events.

An attacker in an IoT environment could compromise individual nodes and then utilise them to launch assaults. An attacker in an RPL-based lossy network, for instance, could trick their neighbours into thinking they are more ranked than they actually are. Another issue with the IoT is the possibility of getting a distorted or incompatible picture of the network due to the broken connections. In order to fix inaccurate data, it is crucial to identify it.

Algorithm 1 Coverage Set Optimization for IoT Networks using KMA

```
Variables:
$N$: Total number of nodes
$Nodes[N]$: Array of all nodes
$Range[N]$: Communication range of each node
$Energy[N]$: Energy level of each node
$Status[N]$: Active/Inactive status of each node
$CoverageSet[]$: List of nodes in the coverage set
$Key[N]$: Security key for each node
Steps:
Step 1: Initialization:
Initialize $Nodes$, $Range$, $Energy$, $Status$ for all $N$ nodes.
Step 2: Form Initial Coverage Set:
Add eligible and active nodes to $CoverageSet$.
Step 3: Optimize Coverage:
Add neighboring nodes to $CoverageSet$ based on eligibility.
Step 4: Integrate Key Match Detection:
Assign $Key$ to each node and establish secure connections among nodes in $CoverageSet$.
Step 5: Routing and Energy Management:
for each node in $CoverageSet$ do
    Optimize routing and manage energy.
end for
Step 6: Adaptation and Maintenance:
Update $CoverageSet$ based on changes in node status.
Step 7: Monitoring and Evaluation:
Continuously monitor coverage efficiency and energy levels.
Step 8: Deployment:
Implement in actual network with real-time monitoring.
```

Algorithm 1 represents The Coverage Set Optimization Algorithm for IoT networks efficiently manages network node coverage while ensuring security. It initializes by assessing each node's status, range, and energy level. An initial coverage set is formed with active, high-energy nodes. The algorithm then optimizes this set by including neighboring nodes, enhancing network reach and reliability. Concurrently, it integrates security by assigning unique keys to nodes, enabling secure communication. This method effectively balances network efficiency, robustness, and security, making it ideal for dynamic IoT environments.

The Algorithm 2 shows the Key Match Detection Algorithm (KMA), used for identifying suspected nodes in a network, operates in a straightforward manner. Each node in the network is assigned a unique cryptographic key. The network continuously monitors communications between nodes, checking the keys used in these interactions. During communication, each node's key is verified against a list of trusted keys. If a node's key doesn't match any in the trusted list, it is flagged as a suspected node. This process ensures that only authenticated nodes can communicate securely within the network, enhancing overall security. We established the node's present degree. The present degree of the node must be met in order for the statement to be true. If the degree of the neighbouring node is more than 20%, we will consider that node to be far from the present node. Based on their current degree, nodes are suspected when they attempt to draw traffic with the improper degree. Assuming a node is suspicious, we count its degree and add it to the total. The shared key concept is used to protect the system from multiple threats in the proposed work. Each node in the network receives a copy of the key [27]. The data cannot be transmitted forward until the full message is not collaborated and confirmed. The error has been found by utilising the message authentication process, which involves comparing the message at every node and discarding it if there is a discrepancy.

Algorithm 3 explains how to prevent a sinkhole attack or selective forwarding by using the hash function. Every node must match the security pattern that is generated using the generate hash function, which is based on a random number. We established a loop based on the assumption that the first signature size is 3. There is no initial value for the signature [28]. Make a loop that goes from 1 all the way up to the route's entire number of nodes. addition of f to route (i) raised to the power of 2, plus rounding modulo (p,2) make a signature and put its value into the f variable. by assigning the value off to a new variable and storing it in the signature variable as a cell [29], [30], [31], [32], [33], [34].

Algorithm 2 Key Match Detection Algorithm to Identify Suspected Nodes

```
Require: $N$: Total number of nodes, $Key[N]$: Array of cryptographic keys, $TrustedKeys$: List of trusted keys
Ensure: $SuspectedNodes$: List of suspected nodes
Step 1: Distribute Keys
for $i \leftarrow 1$ to $N$ do
    Assign unique key $Key[i]$ to each node
end for
Step 2: Monitor
Continuously monitor communications between nodes
Step 3: Verify Keys
for each communicating node $i$ do
    if $Key[i] \notin TrustedKeys$ then
        Step 4: Flag Suspected Nodes
        Add node $i$ to $SuspectedNodes$
    end if
end for
```

Algorithm 3 Stopping Hash Function-Based Algorithm

```
Require: $N$: Total number of nodes, $Data[N]$: Data to be transmitted, $Hash[N]$: Hash values, $ReceivedHash[N]$: Received hash values
Ensure: $IntegrityCheck[N]$: Status of data integrity check (True/False)
Step 1: Compute Hash
for $i = 1$ to $N$ do
    Compute $Hash[i]$ from $Data[i]$ using a hash function
end for
Step 2: Transmit Data
for $i = 1$ to $N$ do
    Send $Data[i]$ and $Hash[i]$ to the intended recipient
end for
Step 3: Verify Integrity
for each received $Data[i]$ and $Hash[i]$ at the recipient do
    Compute $ReceivedHash[i]$ from the received $Data[i]$
    if $ReceivedHash[i] \neq Hash[i]$ then
        Step 4: Flag Integrity Issues
        Set $IntegrityCheck[i] \leftarrow$ False    ▷ Potential security issue detected
    else
        Set $IntegrityCheck[i] \leftarrow$ True
    end if
end for
```

The Prevention Algorithm 3 using a hash function is designed to ensure data integrity in network communications. Each node in the network first computes a hash value of its data before transmission. This hash, generated through a secure hash function, acts as a digital fingerprint of the data. When data is transmitted from one node to another, the accompanying hash value is also sent [35], [36], [37]. Upon receiving the data, the recipient node recalculates the hash value of the received data and compares it with the original hash value sent. If the two hash values match, it confirms that the data has not been tampered with during transmission. If there's a mismatch, it signals a potential security issue, indicating that the data integrity has been compromised. This process effectively prevents data tampering and ensures secure and reliable communication between nodes in the network. Every single node possesses a key, and the intermediate node (or nodes) inspect the key of each and every node. If there is a discrepancy between the keys, then the data transfer will be halted at that precise instant, and the node that is showing suspicious behaviour will be prohibited [38] [39].

B. Cluster-based detection algorithm (CBA)

Attackers can show a fake neighbour value to make nodes nearby think that the compromised node is their close neighbour. The attacker's node can also advertise a low rank value to bring in traffic and launch a selective forwarding attack if it is on the route path. Attackers can also start other kinds of attacks in the network. For example, they could put their own node in the route path to get traffic and make loops that use up the energy of authorised nodes. Even though encryption might help stop these attempts, it might still be possible to get into a real node. We could use asymmetric encryption with a Public Key Infrastructure in an ideal world, but this would use a lot of resources and isn't always possible in real life. Along with the key match method, we use the clustering algorithm to split the network into several groups that make routing easier. For lossy networks, we use the route matrix [40] and divide the cluster heads to look at routing attacks.

Algorithm 4 describe the CBA is designed for intrusion detection and locating intruders in IoT networks [41], [42], [43], [44]. The network is divided into several clusters, each with its own set of nodes and a designated Cluster Head responsible for monitoring activities within the cluster. These Cluster Heads constantly observe network traffic and interactions among nodes in their respective clusters. When a Cluster Head detects unusual or suspicious activity patterns, it flags this as potential intrusion within its cluster. Upon flagging such activity, the algorithm then focuses on analyzing the data patterns and network interactions to pinpoint the location of the intruder. This cluster-based approach allows for more localized and efficient monitoring, making it easier to quickly identify and respond to security threats within specific areas of the IoT network. This algorithm is used to find people who break into an IoT system using fake names to carry out different types of attacks. the environment. To begin, we figure out each node's transmission limits and keep a neighbour table with the names of the nodes that are within their range of transmission. If someone tries to change the identity, we can find out from the neighbour table that they are trying to change identities. This works well for RPL networks. Also, the neighbour table is made up of groups of nodes with

Algorithm 4 Intrusion Detection with Identification of Intruder's Location in IoT Networks

```
Require: $C$: Total number of clusters, $Nodes[C][]$: Array of nodes in each cluster, $ClusterHead[C]$: Head node for each cluster
Ensure: $IntruderLocation$: Location of detected intruder
Step 1: Cluster Formation
Divide the network into $C$ clusters, assigning nodes to each cluster
for $i = 1$ to $C$ do
    Elect $ClusterHead[i]$ for each cluster
end for
Step 2: Monitor Activity
for $i = 1$ to $C$ do
    $ClusterHead[i]$ monitors network activity within its cluster
end for
Step 3: Detect Anomalies
for $i = 1$ to $C$ do
    $ClusterHead[i]$ detects unusual or suspicious patterns in $Nodes[i][]$
end for
Step 4: Flag Suspicious Activity
for $i = 1$ to $C$ do
    if Anomalies detected in $Nodes[i][]$ then
        Set $SuspiciousActivity[i] \leftarrow$ True
    else
        Set $SuspiciousActivity[i] \leftarrow$ False
    end if
end for
Step 5: Identify Intruder Location
for $i = 1$ to $C$ do
    if $SuspiciousActivity[i] =$ True then
        Analyze data to determine $IntruderLocation$ within the cluster
    end if
end for
```

similar transmission ranges. This means that the nodes in this table are similar and have slightly different rank values. If a node from a lower cluster tries to change the name of a node in a higher cluster, the IDS can see that node as an intruder.

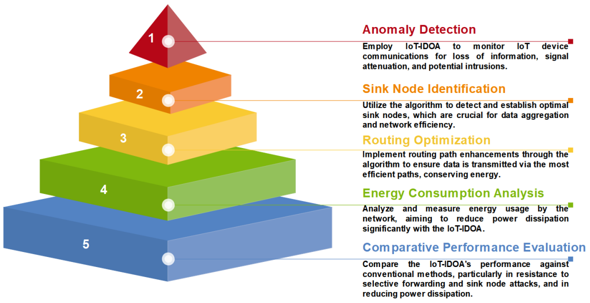
Algorithm 5 describe the steps CBA with Group Authentication is a security strategy designed to prevent the infiltration of malicious nodes in IoT networks. The network is segmented into several clusters, each led by a designated Cluster Head. These Cluster Heads are responsible for distributing unique authentication tokens to all nodes within their respective clusters. When nodes communicate with each other, their authentication tokens are verified to ensure legitimacy [45]. If a node's authentication token does not validate, it is flagged as potentially malicious. Such nodes are then subject to further scrutiny and, depending on the network's security protocol, may be isolated or removed from the network. This group authentication mechanism within each cluster enhances the overall security of the network by ensuring that only verified nodes can participate, thereby significantly reducing the risk of malicious activities [46]. Since it is quite unlikely for nodes at level 7 to have a neighbour at level 2, the base node could identify an invader claiming to have a connection rank of 25 even though its neighbours have a link rank of 76. A scalar value is rank, so keep that in mind. Algorithm 5 demonstrates how a group

Algorithm 5 Group Authentication to Stop Rogue Nodes from Connecting

```
Require: $C$: Total number of clusters, $Nodes[C][]$: Array of nodes in each cluster, $ClusterHead[C]$: Head node for each cluster, $AuthenticationToken[C][]$: Authentication tokens for nodes, $MaliciousNodeFlag[C][]$: Flags indicating malicious nodes
Ensure: Secure network communication free from rogue nodes
Step 1: Cluster Formation
Organize the network into $C$ clusters
for $i = 1$ to $C$ do
    Assign nodes to cluster $i$
    Elect $ClusterHead[i]$ for each cluster
end for
Step 2: Distribute Authentication Tokens
for $i = 1$ to $C$ do
    for each node $j$ in $Nodes[i][]$ do
        Issue $AuthenticationToken[i][j]$ to node $j$ (managed by $ClusterHead[i]$)
    end for
end for
Step 3: Verify Nodes
for each communication within a cluster do
    Verify $AuthenticationToken$ of communicating nodes
    if Authentication fails then
        Set $MaliciousNodeFlag[i][j] \leftarrow$ True for the suspicious node
    end if
end for
Step 4: Flag Malicious Nodes
for $i = 1$ to $C$ do
    for each node $j$ in $Nodes[i][]$ do
        if $MaliciousNodeFlag[i][j] =$ True then
            Mark node $j$ as a potential malicious node
        end if
    end for
end for
Step 5: Isolate or Remove Malicious Nodes
for $i = 1$ to $C$ do
    for each node $j$ with $MaliciousNodeFlag[i][j] =$ True do
        Take action: Isolate or remove node $j$ from the network
    end for
end for
```

authentication technique can thwart a malevolent identity. From the first node in the path to the total number of nodes, we constructed a loop. Create a temporary identifier for the path based on the node temp proximity value. Using the generate group key function, we ensure that the path is clear of any malicious nodes [47], [48], [49], [50]. At all times, the base station verifies the authentication key that each group has. An attack on the network could occur if any group leader's key wasn't registered at the base station. By calculating and storing proximity values in the Node Table, we were able to determine which node was under suspicion. Using a group authentication mechanism, we were able to block the network [51].

The proposed intrusion detection and optimization algorithm (IoT-IDOA): The Figure 2 shows the proposed method methodology and Figure 3 shows the proposed IoT-IDOA used in IoT networks. It outlines a systematic process for detecting intrusions and optimizing network security and efficiency. Starting from the left [52], we have a 'Dynamic Connector', which likely serves as an interface for real-time data input from various network nodes. This could include traffic data, node status, and other relevant metrics.



Employ IoT-IDOA to monitor IoT device communications for loss of information, signal attenuation, and potential intrusions.

Utilize the algorithm to detect and establish optimal sink nodes, which are crucial for data aggregation and network efficiency.

Implement routing path enhancements through the algorithm to ensure data is transmitted via the most efficient paths, conserving energy.

Analyze and measure energy usage by the network, aiming to reduce power dissipation significantly with the IoT-IDOA.

Compare the IoT-IDOA's performance against conventional methods, particularly in resistance to selective forwarding and sink node attacks, and in reducing power dissipation.

Figure 2. Proposed Methodology of proposed IoT-IDOA.

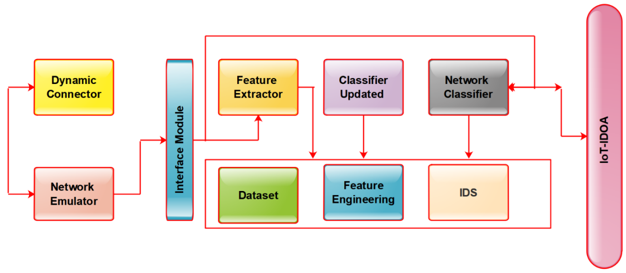


Figure 3. Proposed Fundamental Block Diagram of IoT-IDOA

Next is the 'Network Emulator', which is typically used to replicate network conditions in a controlled environment. It allows the system to simulate and evaluate the behavior of network nodes under different conditions, which is essential for testing the effectiveness of the intrusion detection system (IDS). The 'Interface Module' sits between the data inputs and the core analytical components. It standardizes and possibly preprocesses the data before passing it on to the next stages [53].

Proceeding to the core components, we have the 'Feature Extractor', which analyzes the input data to identify relevant features that can be used to detect potential security breaches. Features might include patterns of traffic that suggest malicious activity or anomalies in data transmission [54]. The 'Dataset' is a collection of data that has likely been labeled or classified in some way to train the system. This dataset is used for 'Feature Engineering', a process where data scientists select and transform variables into formats that make the data more useful for machine learning models. The 'Feature Engineering' output and the original dataset feed into the 'IDS', which stands for Intrusion Detection System. This system uses the features to monitor network traffic and identify suspicious patterns that may indicate a security threat. The 'Classifier Updated' component suggests that the system includes a learning mechanism to improve over time. When new threats are detected or when the system is exposed to new data, the classifier (a type of machine learning model) is updated to recognize these new patterns in the future [55]. The 'Network Classifier' is likely the decision-making component that categorizes network activities into normal or potentially malicious. Based on the classification results, it feeds back into the 'IDS' and the 'Feature Extractor' to refine the detection process. Finally, the output flows into the 'IoT-IDOA', which encompasses the entire process and likely includes a mechanism for acting upon the detection results. This might involve alerting network administrators, automatically adjusting network parameters to mitigate the threat, or updating security protocols.

The proposed algorithm.6 focuses on creating an optimal coverage set that not only ensures efficient network performance but also incorporates security considerations, making it well-suited for IoT environments where both coverage and security are crucial [56], [57], [58], [59], [60], [61], [62], [63], [64].

The algorithm.7 shows The IoT-IDOA steps for detecting suspected nodes using variables can be detailed as follows:

3. Results And Discussions

The process of integrating the detection and prevention algorithm into an IoT setting is detailed in this section. We have used MATLAB simulation to put the algorithms into action. To depict the network with different numbers of nodes, we multiplied the height by the breadth. With the details provided, such as the number of nodes in the network, energy consumption, delay, active time, sleep time, packet drop, and power usage, we construct a simulation. Its origin and final destination node from among the N nodes identified by their x and y coordinates. The scale of the Internet of Things network was 1000*1000. The path is assessed by the network according to the nodes' coverage. Two assaults on the network make up our simulation. In

Algorithm 6 Find Coverage Set of the Network Nodes using IoT-IDOA

```
Variables:
$N$: Total number of nodes in the network.
$Nodes[N]$: Array of all nodes.
$CoverageSet[]$: List of nodes forming the optimal coverage set.
$Range[N]$: Communication range of each node.
$Energy[N]$: Energy level of each node.
$Status[N]$: Status indicating whether a node is active or inactive.
$SecurityRating[N]$: Security rating of each node, based on its vulnerability to intrusion.
Steps:
Step 1: Initialize Network Data: Collect data for each node, including its range, energy, and status.
Step 2: Form Initial Coverage Set: Select initial nodes for $CoverageSet$ based on their energy level, range, and active status.
Step 3: Optimize Coverage Set: Expand $CoverageSet$ by including nodes that enhance overall coverage and have sufficient energy levels.
Step 4: Evaluate and Update Security Ratings: Assess each node's vulnerability to intrusion and assign/update $SecurityRating$.
Step 5: Refine Coverage Set Based on Security: Reassess $CoverageSet$ to include nodes with higher $SecurityRating$, ensuring a secure and efficient network.
Step 6: Continuous Monitoring and Adjustment: Regularly monitor the network and adjust $CoverageSet$ based on changes in node status, energy levels, and security assessments.
```

order to analyse intrusion detection, we analyse power consumption and energy usage in addition to the actual positive rate. We have compared our findings on a scenario with those of SVELTE [7]. True Positive Rate (TPR) for SVELTE was slightly more than 80% across the board. Also, our KMA algorithm achieved a TPR of 50% to over 80%. Most limited IoT devices rely on batteries, making energy and power a crucial resource for these devices. Consequently, in order to accommodate the limited power sources of IoT devices, the protocols for communication and security must be energy efficient. When a node identifies an invader node, we determine its average energy consumption. Upon detection of an incursion, we determine the average power consumption of a single node using the Equation 1.

$$P_c\,(\text{mW}) = \frac{E\,(\text{mJ})}{T\,(\text{s})} \tag{1}$$

Following the description of the implementation setup and methods utilised, we have assessed the energy consumption and true positive rate (the rate at which the network assault is accurately identified) for both sinkhole

Algorithm 7 Detection of Suspected Node in the Network using IoT-IDOA

```
Variables:
$N$: Total number of nodes in the network.
$TrafficData[N]$: Array containing the traffic data for each node.
$NormalBehavior[N]$: Profile of normal behavior patterns for each node.
$AnomalyScore[N]$: Numerical score indicating the level of deviation from normal behavior for each node.
$RiskLevel[N]$: Risk level assigned to each node based on the anomaly score and node's criticality.
$NodeStatus[N]$: Status of each node indicating whether it is suspected or trusted.
$ResponseAction[N]$: Actions to be taken for nodes that are suspected of being compromised.
Steps:
Step 1: Monitor Network:
for $i = 1$ to $N$ do
    $TrafficData[i] \leftarrow$ monitorNodeTraffic($Nodes[i]$)
end for
Step 2: Detect Anomalies:
for $i = 1$ to $N$ do
    $AnomalyScore[i] \leftarrow$ calculateAnomaly($TrafficData[i]$, $NormalBehavior[i]$)
end for
Step 3: Assess Risk Level:
for $i = 1$ to $N$ do
    $RiskLevel[i] \leftarrow$ assessRisk($AnomalyScore[i]$, $Nodes[i]$)
end for
Step 4: Classify Node Status:
for $i = 1$ to $N$ do
    $NodeStatus[i] \leftarrow$ classifyNode($RiskLevel[i]$)
end for
Step 5: Verify Suspected Nodes:
for $i = 1$ to $N$ do
    if $NodeStatus[i]$ == 'Suspected' then
        verifyNode($Nodes[i]$)
    end if
end for
Step 6: Execute Response:
for $i = 1$ to $N$ do
    if $NodeStatus[i]$ == 'Suspected' then
        $ResponseAction[i] \leftarrow$ respondToThreat($Nodes[i]$)
    end if
end for
```

and selective forwarding. From ten to sixty nodes, we have tested our findings. In the network, nodes were randomly plotted. Where P_c is defined as power consumption, E is identified the energy resources of IoT devices, T is represents the time in seconds. Table I shows the simulation parameters that have been considered for performance analysis to determine the forwarding attack, sinkhole attack, and power dissipation in the IoT network.

TABLE I. Simulation Parameters for the Performance Analysis Between Proposed and Conventional Methods

Parameters	Values
Number of Nodes (N)	100
Communication Range (CR)	50m
Traffic Rate (TR)	500kbps
Node Mobility (NM)	1.5m/s
Data Packet Size (DPS)	128 bytes
Simulation Time (ST)	2 hours
Bandwidth (BW)	2 Mbps
Energy Consumption (EC)	0.5 J

Figure 4 shows the identification of the true positive rate for selective forwarding attacks that occur in an IoT network when malicious activity occurs. As per the simulation analysis, the performance of the proposed method shows better as compared to conventional methods of 0.235%, 0.35% KMA, and CBA, respectively.

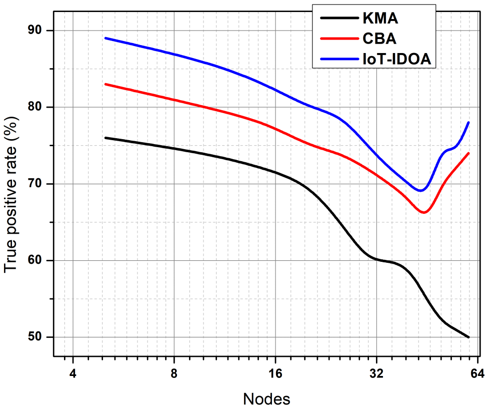


Figure 4. Performance analysis of true positive rate for selective forwarding attack in IoT network

Figure 5 shows the identification of the true positive rate for Sinkhole Attack that occur in an IoT network when malicious activity occurs. As per the simulation analysis, the performance of the proposed method shows better as compared to conventional methods 0.365%, 0.385% KMA, and CBA, respectively.

Figure 6 shows the power dissipation occur in an IoT network when malicious activity occurs. As per the simulation analysis, the performance of the proposed method shows better as compared to conventional methods 0.65%, 0.72% KMA, and CBA, respectively.

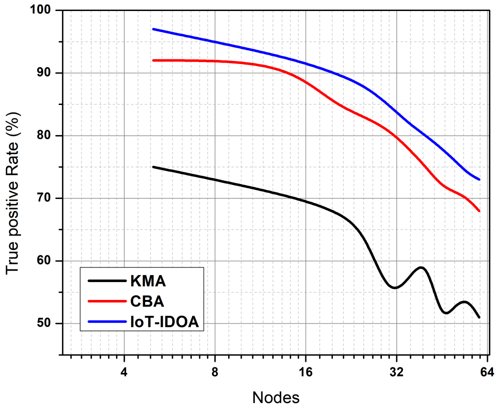


Figure 5. Performance analysis of true positive rate for Sinkhole Attack in IoT network

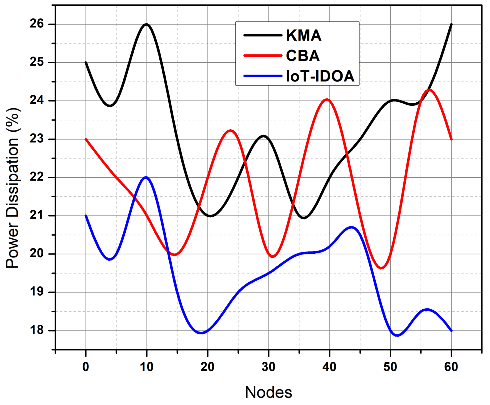


Figure 6. Performance analysis of true positive rate for power dissipation in IoT network

As indicated in Equation 2, we have estimated the average true positive rate (AT_{pr}) for both attacks using KMA and CBA. This is the ratio of the number of true alarms for sinkhole attack (TA_s) and the number of true alarms for selective forwarding (TS_{Af}) divided by the total number of alarms is given in an Equation 2.

$$AT_{pr} = \frac{TS_{Af} + TA_s}{\text{Number of alarms}} \times 100 \tag{2}$$

As part of our experiment, we selected rogue nodes from the Internet of Things environment at random. Remember that we make use of a group of ten to sixty nodes.

4. Conclusion And Future Scope

The proposed IoT-IDOA system detects intrusion, private loss information, and attenuation in a large smart network in a city. The proposed system is capable of identifying the sink nodes, enhancing the multipath routine path choice to improve the communication process in 5G applications. Apart from this, the proposed algorithm efficiently reduces the power dissipation during communication, which has been performed in the multipath routing process. This simulation shows that the proposed system works better than common methods like KMA and CBA when it comes to the selective forwarding attack, sink node attacks, and power dissipation by 0.235%, 0.365% and 0.65%, respectively., compared to common methods.

The future of IoT-based intrusion detection and optimization algorithms in smart networks is poised to revolutionize how we secure and manage connected devices. With the rapid expansion of IoT, the focus will be on developing advanced, AI-driven intrusion detection systems that can quickly identify and neutralize threats, ensuring robust network security. Optimization algorithms will also evolve to enhance network performance, focusing on improving bandwidth efficiency, reducing latency, and ensuring energy efficiency, particularly for battery-operated devices. Customizability and scalability will be key, catering to various network sizes and types. Additionally, these developments will prioritize user privacy and data protection, integrating cutting-edge encryption methods and adhering to international data privacy regulations. This integration of sophisticated technologies is crucial for harnessing the full potential of IoT in creating secure, efficient, and smart connected environments.

References

[1] B. Xiang, C. Zhang, J. Wang, and B. Wang, "Network intrusion detection method for secondary system of intelligent substation based on semantic enhancement," in *2022 4th International Conference on Electrical Engineering and Control Technologies (CEECT)*, Shanghai, China, 2022, pp. 796–800.

[2] H. V, K. R., R. Sindhuja, A. Srivastava, S. G., and S. C. Mary Sundararajan, "Automated intrusion detection and classification using binary metaheuristics with deep learning on smart cities," in *2023 International Conference on Sustainable Communication Networks and Application (ICSCNA)*, Theni, India, 2023, pp. 24–29.

[3] M. Jain and A. Arora, "A novel distributed anomaly intrusion detection model for drone swarm network in smart nations," in *2023 Second International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, Singapore, Singapore, 2023, pp. 87–91.

[4] Q. Shan, "Wireless network intrusion detection model and safety enhancement framework for campus network," in *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2022, pp. 349–353.

[5] T. Zhang and S. Bao, "A novel deep neural network model for computer network intrusion detection considering connection efficiency of network systems," in *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2022, pp. 962–965.

[6] M. Y. Wu, S. H. Wu, Y. E. Chang, Y. H. Lin, S. J. Huang, and H. T. Tseng, "Intrusion detection with radio frequency sensing based on wi-fi mesh network for home security," in *2023 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*, PingTung, Taiwan, 2023, pp. 329–330.

[7] T. Yang, J. Wang, H. Deng, and M. Li, "A data enhancement model for intrusion detection in smart home," in *2023 4th International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, Zhuhai, China, 2023, pp. 314–317.

[8] B. Xu and Y. Chen, "Wireless sensor network intrusion detection model for real-time hazardous chemical monitoring," in *2023 3rd International Conference on Smart Data Intelligence (ICSMDI)*, Trichy, India, 2023, pp. 177–180.

[9] S. J. Kamal, G. Karishma, S. T, A. Rajesh, V. Muthiah-Nakarajan, and S. B, "Ann-lstm assisted intrusion detection for next generation core networks," in *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, Vellore, India, 2023, pp. 1–5.

[10] C. I. Nwakanma, L. A. C. Ahakonye, T. Jun, J. M. Lee, and D. S. Kim, "Explainable scada-edge network intrusion detection system: Tree-lime approach," in *2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Glasgow, United Kingdom, 2023, pp. 1–7.

[11] S. S. Kanumalli, K. L, R. A, S. P, and T. M, "A scalable network intrusion detection system using bi-lstm and cnn," in *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, Coimbatore, India, 2023, pp. 1–6.

[12] M. Naveed, S. M. Usman, M. I. Satti, S. Aleshaiker, and A. Anwar, "Intrusion detection in smart iot devices for people with disabilities," in *2022 IEEE International Smart Cities Conference (ISC2)*, Pafos, Cyprus, 2022, pp. 1–5.

[13] B. S. Babu, G. A. Reddy, D. K. Goud, K. Naveen, and K. S. T. Reddy, "Network intrusion detection using machine learning algorithms," in *2023 3rd International Conference on Smart Data Intelligence (ICSMDI)*, Trichy, India, 2023, pp. 367–371.

[14] R. Vinod Kumar, G. Jayasri, M. M. A, and E. Vidya, "Smart clustering attack detection system," in *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, Coimbatore, India, 2023, pp. 1537–1542.

[15] S. Ullah, W. Boulila, A. Koubaa, Z. Khan, and J. Ahmad, "Abdnn-ids: Attention-based deep neural networks for intrusion detection in industrial iot," in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, Hong Kong, Hong Kong, 2023, pp. 1–5.

[16] S. Chatzimiltis, M. Shojafar, and R. Tafazolli, "A distributed intrusion detection system for future smart grid metering network," in *ICC 2023 - IEEE International Conference on Communications*, Rome, Italy, 2023, pp. 3339–3344.

[17] A. J. Wadate and S. P. Deshpande, "Edge-based intrusion detection using machine learning over the iot network," in *2023 11th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET - SIP)*, Nagpur, India, 2023, pp. 1–6.

[18] P. Illy, G. Kaddoum, K. Kaur, and S. Garg, "Ml-based idps enhancement with complementary features for home iot networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 772–783, June 2022.

[19] C.-C. Sun, D. J. Sebastian Cardenas, A. Hahn, and C.-C. Liu, "Intrusion detection for cybersecurity of smart meters," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 612–622, Jan. 2021.

[20] P. Illy and G. Kaddoum, "A collaborative dnn-based low-latency idps for mission-critical smart factory networks," *IEEE Access*, vol. 11, pp. 96 317–96 329, 2023.

[21] S. Thazeen, S. Mallikarjunaswamy, M. N. Saqhib, and N. Sharmila, "Doa method with reduced bias and side lobe suppression," in *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, Chennai, India, 2022, pp. 1–6.

[22] P. Dayananda, M. Srikantaswamy, S. Nagaraju, R. Velluri, and M. K. Doddananjedevaru, "Efficient detection of faults and false data injection attacks in smart grid using a reconfigurable kalman filter," *International Journal of Power Electronics and Drive Systems*, vol. 13, no. 4, pp. 2086–2097, 2022.

[23] H. N. Mahendra, S. Mallikarjunaswamy, and S. R. Subramoniam, "An assessment of vegetation cover of mysuru city, karnataka state, india, using deep convolutional neural networks," *Environmental Monitoring and Assessment*, vol. 195, no. 4, p. 526, 2023.

[24] S. Mallikarjunaswamy, K. R. Nataraj, and K. R. Rekha, "Design of high-speed reconfigurable coprocessor for next-generation communication platform," in *Emerging Research in Electronics, Computer Science and Technology*, ser. Lecture Notes in Electrical Engineering, V. Sridhar, H. Sheshadri, and M. Padma, Eds. Springer, New Delhi, 2014, vol. 248, pp. 77–85.

[25] G. S. Pavithra, S. Pooja, V. Rekha, H. N. Mahendra, N. Sharmila, and S. Mallikarjunaswamy, "Comprehensive analysis on vehicle-to-vehicle communication using intelligent transportation system," in *Soft Computing for Security Applications*, ser. Advances in Intelligent Systems and Computing, G. Ranganathan, Y. EL Allioui, and S. Piramuthu, Eds. Springer, Singapore, 2023.

[26] H. N. Mahendra, S. Mallikarjunaswamy, D. M. Kumar, S. Kumari, S. Kashyap, S. Fulwani, and A. Chatterjee, "Assessment and prediction of air quality level using arima model: A case study of surat city, gujarat state, india," *Nature Environment & Pollution Technology*, vol. 22, no. 1, pp. 199–210, 2023.

[27] M. L. Umashankar, S. Mallikarjunaswamy, N. Sharmila, D. M. Kumar, and K. R. Nataraj, "A survey on iot protocol in real-time applications and its architectures," in *ICDSMLA 2021*, ser. Lecture Notes in Electrical Engineering, A. Kumar, S. Senatore, and V. K. Gunjan, Eds. Springer, Singapore, 2023, vol. 947, pp. 155–166.

[28] Q. Liu, V. Hagenmeyer, and H. B. Keller, "A review of rule learning-based intrusion detection systems and their prospects in smart grids," *IEEE Access*, vol. 9, pp. 57 542–57 564, 2021.

[29] J. Shi, S. Liu, B. Chen, and L. Yu, "Distributed data-driven intrusion detection for sparse stealthy fdi attacks in smart grids," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 3, pp. 993–997, March 2021.

[30] L. Nie, Z. Ning, X. Wang, X. Hu, J. Cheng, and Y. Li, "Data-driven intrusion detection for intelligent internet of vehicles: A deep convolutional neural network-based method," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2219–2230, 2021.

[31] A. A. Elsaeidy, N. Jagannath, A. G. Sanchis, A. Jamalipour, and K. S. Munasinghe, "Replay attack detection in smart cities using deep learning," *IEEE Access*, vol. 8, pp. 137 825–137 837, 2020.

[32] W. Alhakami, A. ALharbi, S. Bourouis, R. Alroobaea, and N. Bouguila, "Network anomaly intrusion detection using a non-parametric bayesian approach and feature selection," *IEEE Access*, vol. 7, pp. 52 181–52 190, 2019.

[33] S. Jiang, J. Zhao, and X. Xu, "Slgbm: An intrusion detection mechanism for wireless sensor networks in smart environments," *IEEE Access*, vol. 8, pp. 169 548–169 558, 2020.

[34] Y. Li, J. Zhang, Y. Yan, Y. Lei, and C. Yin, "Enhancing network intrusion detection through the application of the dung beetle optimized fusion model," *IEEE Access*, vol. 12, pp. 9483–9496, 2024.

[35] M. A. Ferrag, L. Shu, O. Friha, and X. Yang, "Cyber security intrusion detection for agriculture 4.0: Machine learning-based solutions, datasets, and future directions," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 3, pp. 407–436, March 2022.

[36] Y. Zhang, L. Wang, and W. Sun, "Trust system design optimization in smart grid network infrastructure," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 184–195, March 2013.

[37] H. Mahendra, S. Mallikarjunaswamy, and S. Subramoniam, "An assessment of built-up cover using geospatial techniques–a case study on mysuru district, karnataka state, india," *International Journal of Environmental Technology and Management*, vol. 26, no. 3-5, pp. 173–188, 2023.

[38] S. Pooja, M. Mallikarjunaswamy, and S. Sharmila, "Image region driven prior selection for image deblurring," *Multimedia Tools and Applications*, vol. 82, pp. 24 181–24 202, 2023.

[39] S. Rathod, N. Ramaswamy, M. Srikantaswamy, and R. Ramaswamy, "An efficient reconfigurable peak cancellation model for peak to average power ratio reduction in orthogonal frequency division multiplexing communication system," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 6, pp. 6239–6247, 2022.

[40] S. Mallikarjunaswamy, N. Basavaraju, N. Sharmila, H. Mahendra, S. Pooja, and B. Deepak, "An efficient big data gathering in wireless sensor network using reconfigurable node distribution algorithm," in *2022 Fourth International Conference on Cognitive Computing and Information Processing (CCIP)*, 2022, pp. 1–6.

[41] H. Mahendra and S. Mallikarjunaswamy, "An efficient classification of hyperspectral remotely sensed data using support vector machine," *International Journal of Electronics and Telecommunications*, vol. 68, no. 3, pp. 609–617, 2022.

[42] R. Shivaji, K. Nataraj, S. Mallikarjunaswamy, and K. Rekha, "Implementation of an effective hybrid partial transmit sequence model for peak to average power ratio in mimo ofdm system," in *ICDSMLA 2020*, ser. Lecture Notes in Electrical Engineering, A. Kumar, S. Senatore, and V. Gunjan, Eds. Springer, Singapore, 2022, vol. 783.

[43] A. Savitha and M. Jayaram, "Development of energy efficient and secure routing protocol for m2m communication," *International Journal of Performability Engineering*, vol. 18, no. 6, pp. 426–433, 2022.

[44] D. Venkatesh, K. Mallikarjunaiah, and M. Srikantaswamy, "A comprehensive review of low density parity check encoder techniques," *Ingénierie des Systèmes d'Information*, vol. 27, no. 1, pp. 11–20, 2022.

[45] E. B. Mbaya *et al.*, "Secfedidm-v1: A secure federated intrusion

detection model with blockchain and deep bidirectional long short-term memory network," *IEEE Access*, vol. 11, pp. 116 011–116 025, 2023.

[46] 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.

[47] X. Wang, J. He, Z. Xie, G. Zhao, and S.-C. Cheung, "Contractguard: Defend ethereum smart contracts with embedded intrusion detection," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 314–328, 2020.

[48] Z. Wang, X. Xie, L. Chen, S. Song, and Z. Wang, "Intrusion detection and network information security based on deep learning algorithm in urban rail transit management system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2135–2143, 2023.

[49] S. Thazeen, S. Mallikarjunaswamy, and M. Saqhib, "Septennial adaptive beamforming algorithm," in *2022 International Conference on Smart Information Systems and Technologies (SIST)*, 2022, pp. 1–4.

[50] H. Mahendra, S. Mallikarjunaswamy, N. Basavaraju, P. Poojary, P. Gowda, M. Mukunda, B. Navya, and V. Pushpalatha, "Deep learning models for inventory of agriculture crops and yield production using satellite images," in *2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)*, 2022, pp. 1–7.

[51] H. Mahendra, S. Mallikarjunaswamy, C. Nooli, M. Hrishikesh, N. Kruthik, and H. Vakkalanka, "Cloud based centralized smart cart and contactless billing system," in *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, 2022, pp. 820–826.

[52] S. Mallikarjunaswamy, N. Sharmila, G. Siddesh, K. Nataraj, and M. Komala, "A novel architecture for cluster based false data injection attack detection and location identification in smart grid," in *Advances in Thermofluids and Renewable Energy*, ser. Lecture Notes in Mechanical Engineering, P. Mahanta, P. Kalita, A. Paul, and A. Banerjee, Eds. Springer, Singapore, 2022.

[53] S. Thazeen, S. Mallikarjunaswamy, G. Siddesh, and N. Sharmila, "Conventional and subspace algorithms for mobile source detection and radiation formation," *Traitement du Signal*, vol. 38, no. 1, pp. 135–145, 2021.

[54] P. Satish, M. Srikantaswamy, and N. Ramaswamy, "A comprehensive review of blind deconvolution techniques for image deblurring," *Traitement du Signal*, vol. 37, no. 3, pp. 527–539, 2020.

[55] M. Umashankar, M. Ramakrishna, and S. Mallikarjunaswamy, "Design of high speed reconfigurable deployment intelligent genetic algorithm in maximum coverage wireless sensor network," in *2019 International Conference on Data Science and Communication (IconDSC)*, 2019, pp. 1–6.

[56] H. Mahendra, S. Mallikarjunaswamy, V. Rekha, V. Puspalatha, and N. Sharmila, "Performance analysis of different classifier for remote sensing application," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 1, pp. 7153–7158, 2019.

[57] S. Thazeen and S. Mallikarjunaswamy, "The effectiveness of 6t beamformer algorithm in smart antenna systems for convergence analysis," *IIUM Engineering Journal*, vol. 24, no. 2, pp. 100–116, 2023.

[58] D. Y. Venkatesh, K. Mallikarjunaiah, and M. Srikantaswamy, "An efficient reconfigurable code rate cooperative low-density parity check codes for gigabits wide code encoder/decoder operations," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 6, pp. 6369–6377, 2023.

[59] M. M. Pandith, N. K. Ramaswamy, M. Srikantaswamy, and R. K. Ramaswamy, "An efficient reconfigurable geographic routing congestion control algorithm for wireless sensor networks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 6, pp. 6388–6398, 2023.

[60] S. Thazeen and M. Srikantaswamy, "An efficient reconfigurable optimal source detection and beam allocation algorithm for signal subspace factorization," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 6, pp. 6452–6465, 2023.

[61] C. Chikkasiddaiah, P. Govindaswamy, and M. Srikantaswamy, "An efficient hydro-crop growth prediction system for nutrient analysis using machine learning algorithm," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 6, pp. 6681–6690, 2023.

[62] P. Goravi Sukumar and M. Krishnaiah, "An efficient adaptive reconfigurable routing protocol for optimized data packet distribution in network on chips," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 1, pp. 305–314, 2024.

[63] R. Sathyanarayana and N. K. Ramaswamy, "An efficient unused integrated circuits detection algorithm for parallel scan architecture," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 1, pp. 469–478, 2024.

[64] M. M. Pandith, N. K. Ramaswamy, M. Srikantaswamy, and R. K. Ramaswamy, "Efficient geographic routing for high-speed data in wireless multimedia sensor networks," *Journal Européen des Systèmes Automatisés*, vol. 56, no. 6, pp. 1003–1017, 2023.