DOI: 10.21917/ijme.2025.0345

# ENHANCED DEEP LEARNING-BASED TRUST MANAGEMENT FOR UNRELIABLE SERVICE DISCOVERY IN HIGHLY DYNAMIC SOC-MANET ENVIRONMENTS

**P. Ramya[1], S. Venkatesh Babu[2] and D. Jebakumar Immanuel[3]**

*[1]Department of Artificial Intelligence and Data Science, PSNA College of Engineering and Technology, India*
*[2]Department of Information Technology, KGISL Institute of Technology, India*
*[3]Department of Artificial Intelligence and Data Science, Karpagam Institute of Technology, India*

*Abstract*

***System on Chip - Mobile Ad Hoc Networks (SoC-MANETs) are characterized by their decentralized architecture, node mobility, and frequent topology changes. Ensuring secure and reliable service discovery in such environments is critical but challenging due to node misbehavior and trust uncertainties. Traditional trust management mechanisms often fall short in detecting malicious behavior and adapting to network dynamics, especially under unreliable service conditions. This paper proposes a Deep Self-Organizing Control (Deep SOC)-based trust management framework that integrates deep learning with adaptive behavior profiling to enhance the reliability of service discovery. A Convolutional Neural Network (CNN) is employed to predict node trustworthiness based on real-time communication patterns, mobility behavior, and packet integrity. The proposed Deep SOC model was tested using NS-3 simulation with 100 nodes under varying mobility and attack scenarios. It achieved a Packet Delivery Ratio (PDR) of 92.8%, End-to-End Delay (E2ED) of 116 ms, Detection Accuracy of 95.4%, Trust Convergence Time of 8.4s, and Energy Consumption of 21.3J outperforming existing methods by 12–18% across metrics.***

***Keywords:***

***SoC-MANET, Trust Management, Deep Learning, Service Discovery, Network Security***

## 1. INTRODUCTION

Mobile Ad Hoc Networks (SoC-MANETs) are self-organizing networks that consist of mobile nodes that communicate wirelessly without the need for a centralized infrastructure. This inherent flexibility and autonomy make SoC-MANETs particularly suitable for applications in dynamic and emergency environments, such as military communication, disaster recovery, and vehicular networks. However, the open nature and lack of centralized control in SoC-MANETs also introduce several challenges, especially in terms of security, trust, and resource management.

In SoC-MANETs, trust management plays a critical role in ensuring the reliable and secure exchange of information among nodes. Nodes in these networks often rely on trust to determine which nodes are reliable and which may be malicious. In many cases, nodes form trust relationships based on prior interactions, which are then used to make decisions about routing, resource sharing, and service discovery. Trust is fundamental to the proper functioning of the network, as it allows nodes to cooperate and share information, and provides mechanisms to mitigate malicious behaviors.

However, unreliable service discovery in SoC-MANETs is a significant challenge due to the dynamic nature of the network. Nodes may join or leave the network unexpectedly, and their mobility can lead to frequent changes in the topology. These dynamics make it difficult to maintain reliable service discovery protocols, further complicating trust management. As a result, a node may not always be able to accurately assess the trustworthiness of its peers, leading to unreliable communication and potential attacks, such as packet dropping, blackhole attacks, or sybil attacks.

Moreover, the resource constraints of SoC-MANET nodes, including limited battery life, computational power, and bandwidth, exacerbate these issues. Traditional trust management systems often fail to address the dynamic nature of the network, which leads to delays in trust convergence and high energy consumption. Therefore, there is a pressing need for trust management frameworks that are adaptive, efficient, and capable of handling the dynamic and unreliable conditions characteristic of SoC-MANETs.

The main challenges in trust management for SoC-MANETs arise from several interrelated factors:

- **Dynamic Network Topology:** As nodes in a SoC-MANET are mobile, the network topology can change frequently, making it difficult to maintain stable trust relationships. This dynamic nature challenges traditional trust models, which assume a more stable network environment.
- **Unreliable Service Discovery:** In SoC-MANETs, nodes may need to discover services offered by other nodes. However, due to the mobility and unpredictability of nodes, service discovery can become unreliable. Trust management models need to be robust against such variations and ensure that only trusted nodes are involved in service discovery [4].
- **Energy Efficiency:** Nodes in a SoC-MANET typically have limited energy resources. Traditional trust management models may require extensive communication, which can deplete the battery of nodes, making them unsuitable for long-term operations in large-scale networks [5]. This calls for the development of energy-efficient trust models that can balance the trade-off between trust management and energy consumption.
- **Security Vulnerabilities:** Trust management systems in SoC-MANETs are vulnerable to attacks, such as the Sybil attack, where a malicious node can create multiple fake identities to manipulate the trust relationships. These attacks can undermine the accuracy of trust evaluations and disrupt the performance of the network [6].
- **Scalability:** As SoC-MANETs scale to hundreds or thousands of nodes, the complexity of managing trust relationships increases. Efficient algorithms that can handle large-scale networks without excessive computational or communication overhead are necessary.

The problem at hand involves the development of a robust trust management framework for unreliable service discovery in SoC-MANETs. The main issues include: Handling dynamic topology changes while maintaining accurate trust evaluations. Efficient and accurate trust propagation despite the unreliable and frequently changing network environment. Minimizing the energy consumption of trust management processes. Ensuring resilience to security threats such as malicious nodes trying to manipulate trust relationships.

This problem requires a solution that can adapt to the frequent changes in the network topology and trust values, while providing a mechanism for maintaining reliable communication and service discovery.

The objectives of this work are to: Develop an adaptive trust management framework that can handle the dynamic and unreliable environment of SoC-MANETs. Propose an efficient mechanism for trust propagation and reputation updates that minimizes energy consumption while maintaining trust accuracy. Implement a solution that is resistant to common security threats, including Sybil attacks and packet dropping.

The novelty of this approach lies in its combination of deep learning techniques (e.g., CNNs) and a reinforcement feedback loop for adaptive trust management. The deep learning model allows for real-time learning of trust dynamics from interactions, while the reinforcement feedback loop ensures continuous adaptation to changing network conditions. Additionally, the proposed method integrates time-window-based smoothing and adaptive trust thresholds for better trust propagation and faster convergence.

The key contributions of this work include development of a Deep SOC-based Framework for trust management that leverages CNNs for feature extraction and reinforcement learning for trust adaptation. Introduction of an adaptive reputation update mechanism that efficiently balances growth and conservation of resources. A novel trust propagation engine that incorporates time-window-based smoothing and adaptive thresholds to enhance the accuracy and speed of trust convergence. A detailed analysis of the proposed methods performance in terms of packet delivery ratio, energy consumption, and trust convergence time compared to existing methods like T-RAT, TRM-FL, and GTMS.

## 2. RELATED WORKS

In the literature, several trust management models have been proposed for SoC-MANETs, each focusing on different aspects of trust and security in mobile networks. Some of the notable works are outlined below.

T-RAT (Trust-aware Routing with Adaptive Thresholds) method focuses on adapting trust thresholds based on network conditions. T-RAT dynamically adjusts the routing decisions based on trust values, which can help mitigate attacks in the network [9]. However, its reliance on fixed thresholds can lead to suboptimal performance in highly dynamic environments.

TRM-FL (Trust and Reputation Model with Fuzzy Logic): TRM-FL utilizes fuzzy logic to evaluate the trustworthiness of nodes based on multiple factors, including past behavior and reputation. The model allows for a more nuanced understanding of trust relationships and can handle uncertainty in the trust evaluation process [10]. However, the fuzzy logic-based approach can be computationally expensive and may not scale well in large networks.

GTMS (General Trust Management Scheme): GTMS is a generic framework for trust management that uses a combination of reputation-based and behavior-based trust evaluation. It is designed to work in heterogeneous and dynamic environments, making it suitable for SoC-MANETs [11]. However, GTMS does not account for the energy consumption of trust management processes, which is crucial in resource-constrained mobile environments.

ETR (Energy-efficient Trust and Reputation): ETR integrates energy efficiency into trust management by considering the energy consumption of nodes during the trust evaluation process. By adjusting trust decisions based on energy levels, this model aims to reduce the energy consumption of nodes, but it may still suffer from delayed trust convergence in large networks [12].

Cooperative Trust Model (CTM): The Cooperative Trust Model focuses on nodes cooperation levels and uses these levels to build trust relationships. While this model is effective in fostering cooperation, it does not explicitly address the issue of unreliable service discovery or malicious behavior in dynamic environments [13].

SecTrust (Secure Trust Management): SecTrust introduces security into trust management by adding cryptographic techniques to the trust evaluation process. This ensures the integrity of the trust relationships, but the computational overhead may limit its practicality in resource-constrained SoC-MANETs [14].

Trust-based Reputation System (TRS): TRS is a reputation-based approach that evaluates trust based on the feedback from other nodes. It uses a scoring system to assess the reliability of nodes, but this system can be vulnerable to attacks that manipulate feedback [15].

RAT (Reputation-based Trust Management): RAT focuses on establishing reputation scores for nodes and uses these scores to make routing decisions. It helps improve security in SoC-MANETs by identifying malicious nodes, but it may not be effective in networks with high mobility or rapidly changing topologies [16].

Each of these methods has its strengths and weaknesses, particularly in terms of scalability, adaptability to dynamic environments, and energy efficiency. While some models focus on trust propagation or security, others aim to optimize energy usage. However, most existing solutions fail to address all the challenges posed by SoC-MANETs, particularly in large, highly dynamic networks with unreliable service discovery and high energy constraints.

## 3. PROPOSED METHOD

The proposed method leverages a Deep SOC framework that integrates Convolutional Neural Networks (CNNs) for trust evaluation with dynamic trust updates through a reinforcement feedback loop. Nodes in the SoC-MANET continuously monitor neighbor behaviors by tracking packet forwarding ratios, delay patterns, and signal fluctuations. These metrics are transformed into structured inputs for the CNN, which classifies nodes as trustworthy, suspicious, or malicious. The system then feeds this trust classification into a trust propagation engine, adjusting node trust scores in real time. To address uncertainty in dynamic environments, the Deep SOC employs a time-window-based smoothing algorithm to filter transient anomalies. CNN is trained on a labeled dataset of node behaviors under normal and attack scenarios, allowing it to generalize across diverse network conditions. Additionally, the model features a reputation update mechanism based on multi-hop observations to enhance detection of colluding attackers. The adaptive trust thresholding mechanism ensures resilience to false positives, improving reliability in unreliable service discovery scenarios.

## 3.1 PROPOSED DEEP SOC FRAMEWORK

The Deep SOC Framework integrates deep learning techniques with adaptive trust propagation mechanisms to enhance service discovery and trust management in dynamic SoC-MANET environments. The framework utilizes Convolutional Neural Networks (CNNs) for node trust evaluation and an adaptive reinforcement feedback loop for real-time trust updates. The approach is designed to work under unreliable conditions, where nodes experience intermittent connectivity, mobility, and frequent attacks.

### *3.1.1 Data Collection for Trust Evaluation:*

In a SoC-MANET, each node maintains a local record of its interactions with neighboring nodes. This includes data such as packet forwarding ratios, delay times, signal strength, and reliability metrics. These records are used as inputs for CNN to determine the trustworthiness of each node. CNN works by processing time-series data to detect patterns of behavior that indicate normal or malicious activity. The network state is periodically updated, and trust scores are computed using deep learning models. For a clearer understanding, consider Table.1, which shows an example dataset for node interactions. This dataset will serve as input for the CNN model.

Table.1. Node Interaction Dataset

| Node ID | Packet Forwarding Ratio | Delay (ms) | Signal Strength (dBm) | Trust Score |
|---|---|---|---|---|
| 1 | 0.85 | 120 | -65 | 0.80 |
| 2 | 0.95 | 110 | -60 | 0.90 |
| 3 | 0.40 | 200 | -80 | 0.30 |
| 4 | 0.70 | 150 | -70 | 0.60 |
| 5 | 0.55 | 180 | -75 | 0.50 |

## 3.2 CNN MODEL ARCHITECTURE FOR TRUST CLASSIFICATION

The CNN model consists of multiple layers designed to extract temporal features and spatial patterns from the node behavior data. The first few layers apply convolutions to detect low-level features like changes in packet forwarding ratios and delay variations, which are indicative of node reliability. The subsequent fully connected layers interpret the high-level features and produce a trust classification output. The CNN architecture can be represented as:

$$\text{Trust Score} = \text{FC}_n(\text{Conv}_3(\text{Conv}_2(\text{Conv}_1(\text{Input Data})))) \quad (1)$$

where,

Conv1,Conv2,Conv3 are convolutional layers,

$FC_n$ refers to the fully connected layer for output trust score.

This output trust score, ranging from 0 to 1, indicates how trustworthy a node is. A higher score signifies more reliable behavior, while a lower score corresponds to suspicious or malicious activity.

## 3.3 TRUST PROPAGATION AND REINFORCEMENT FEEDBACK LOOP

Once the CNN provides an initial trust score, the reinforcement feedback loop refines this score over time. In SoC-MANETs, trust is dynamic and changes based on ongoing interactions. The feedback loop works as follows:

- **Initial Trust Computation:** Trust is computed using the CNN as explained above.
- **Reinforcement Feedback:** The trust scores are updated every few seconds based on observed changes in node behavior. If a node exhibits malicious activity (e.g., dropping packets or sending incorrect data), the feedback loop adjusts its trust score downward.
- **Adaptive Thresholding:** A threshold is defined for each nodes trust score. If the score drops below a certain threshold, it is flagged as suspicious. The trust values are updated periodically to maintain the most accurate representation of node behavior.

The trust update can be expressed as:

$$\text{Trust}_i(t+1) = \alpha \cdot \text{Trust}_i(t) + \beta \cdot \text{Reward}_i(t) \quad (2)$$

where,

$Trust_i(t)$ is the trust score of node $i$ at time $t$,

$Reward_i(t)$ is the reward (or penalty) based on node $i$'s actions at time $t$,

$\alpha$ and $\beta$ are constants that determine the influence of previous trust and the current reward respectively.

The reward can be defined based on the nodes behavior, such as whether it successfully forwards packets or engages in malicious activities.

## 3.4 TRUST AGGREGATION AND NODE CLASSIFICATION

Once the trust scores have been updated, nodes aggregate their trust data from multiple hops and interactions. This aggregated trust score helps classify nodes into categories such as Trustworthy, Suspicious, and Malicious. Nodes with trust scores above a certain threshold are classified as trustworthy, while those below a threshold are flagged as suspicious or malicious.

Table.2. Trust Classification Based on Thresholding

| Node ID | Initial Trust Score | Updated Trust Score | Classification |
|---|---|---|---|
| 1 | 0.80 | 0.85 | Trustworthy |
| 2 | 0.90 | 0.92 | Trustworthy |
| 3 | 0.30 | 0.25 | Malicious |

| | | | |
|---|---|---|---|
| 4 | 0.60 | 0.65 | Trustworthy |
| 5 | 0.50 | 0.55 | Suspicious |

### 3.5 REAL-TIME DECISION MAKING AND TRUST UPDATE

Finally, based on the updated trust scores and classification, the Deep SOC framework makes real-time decisions about which nodes to trust for service discovery. For example, nodes with lower trust values may be excluded from the networks routing path or service discovery protocols to prevent malicious nodes from compromising the networks integrity. These decisions are fed back into the system, which further refines the trust levels of all nodes involved.

The Deep SOC framework is designed to address the challenges of trust management in dynamic and unreliable SoC-MANET environments. By combining CNN-based trust evaluation with an adaptive reinforcement feedback loop, the framework ensures accurate and real-time trust classification of nodes. Through continuous learning and feedback, it is able to adapt to varying network conditions, providing more reliable service discovery and enhancing the overall security of the network.

## 4. PROPOSED TRUST CLASSIFICATION AND PROPAGATION ENGINE

In the proposed Deep SOC framework, trust classification and propagation are central to achieving reliable and adaptive trust management in Mobile Ad Hoc Networks (SoC-MANETs). The framework uses structured inputs derived from various network metrics (such as packet forwarding ratio, delay, and signal strength) to evaluate the trustworthiness of nodes. The process of trust classification feeds into the trust propagation engine, where node trust scores are updated over time, ensuring real-time adaptations based on node behaviors.

### 4.1 METRICS INTO STRUCTURED INPUTS

To begin the trust evaluation process, the framework collects real-time network metrics from each node. These metrics reflect the nodes behavior and interaction with its neighbors. The key metrics that are structured into inputs for the trust classification module include:

- **Packet Forwarding Ratio (PFR):** The fraction of packets successfully forwarded by a node relative to the total packets received.
- **Delay:** The average delay in forwarding packets, a key indicator of node reliability.
- **Signal Strength (SS):** The strength of the signal received from neighboring nodes.
- **Energy Consumption (EC):** The energy used by the node in performing communication tasks, which can also indicate reliability (e.g., energy-constrained nodes may exhibit abnormal behaviors).
- **Packet Delivery Ratio (PDR):** A performance metric measuring the ratio of successfully delivered packets to those sent.

These metrics are normalized and combined into a structured input format, which serves as the input to the CNN model for trust classification.

Table.3. Example of Metrics into Structured Inputs

| Node ID | PFR | Delay (ms) | Signal Strength (dBm) | Energy Consumption (J) | PDR | Structured Input |
|---|---|---|---|---|---|---|
| 1 | 0.85 | 120 | -65 | 0.5 | 0.92 | [0.85, 120, -65, 0.5, 0.92] |
| 2 | 0.95 | 110 | -60 | 0.4 | 0.95 | [0.95, 110, -60, 0.4, 0.95] |
| 3 | 0.60 | 150 | -70 | 0.7 | 0.88 | [0.60, 150, -70, 0.7, 0.88] |
| 4 | 0.75 | 130 | -68 | 0.6 | 0.90 | [0.75, 130, -68, 0.6, 0.90] |
| 5 | 0.50 | 200 | -80 | 0.8 | 0.75 | [0.50, 200, -80, 0.8, 0.75] |

### 4.2 TRUST CLASSIFICATION USING CNN

Once the structured input is created, the next step is to classify the trustworthiness of each node based on the network metrics. The CNN model is used to perform this classification. It learns to identify patterns in the structured input data that indicate trustworthy or malicious behaviors. After processing the data, the CNN outputs a trust score ranging from 0 (untrustworthy) to 1 (trustworthy). The trust score for each node is calculated as follows:

$$\text{Trust Score}_i = \text{CNN}\left(\text{Input}_i\right) \tag{3}$$

where,

*Trust Score$_i$* is the trust value of node *i*,

*Input$_i$* is the structured input vector for node *i* (e.g., [0.85,120,−65,0.5,0.92]).

For example, Node 1 with input vector [0.85,120,−65,0.5,0.92] would output a trust score of 0.88, indicating that it is likely to be trustworthy.

### 4.3 TRUST PROPAGATION ENGINE

The trust propagation engine is responsible for updating the trust scores of nodes over time, based on their interactions and behaviors observed in the network. Trust propagation ensures that nodes continuously adapt their trust values to reflect ongoing interactions with neighboring nodes. The trust score of a node is propagated through the network based on the interactions it has with other nodes. The trust propagation rule is formulated as follows:

$$\text{Trust}_i(t+1) = \alpha \cdot \text{Trust}_i(t) + \beta \cdot \sum_{j \in \text{N}\ (i)} \text{Trust}_j(t) \tag{4}$$

where,

*Trust$_i$*(*t*) is the current trust score of node *i* at time *t*,

N (*i*) represents the set of neighboring nodes of *i*,

α and β are weighting factors that control the influence of the nodes own trust score and the trust of its neighbors.

The propagation process is iterative, with each node updating its trust score based on the behavior of neighboring nodes. For instance, if a node interacts with several trustworthy neighbors, its trust score will increase, reinforcing its reliability. Conversely, interactions with malicious nodes lead to a decrease in trust.

### 4.4 TRUST THRESHOLDING AND NODE CLASSIFICATION

Once trust scores are propagated and updated, a thresholding mechanism is applied to classify nodes into Trustworthy, Suspicious, or Malicious categories. The threshold for classification is set based on the systems tolerance for errors and attack types. For example, nodes with trust scores greater than 0.8 are considered Trustworthy, scores between 0.6 and 0.8 are classified as Suspicious, and those with trust scores lower than 0.6 are classified as Malicious.

Table.4. Trust Classification Based on Propagation

| Node ID | Initial Trust Score | Updated Trust Score | Classification |
|---|---|---|---|
| 1 | 0.88 | 0.92 | Trustworthy |
| 2 | 0.90 | 0.93 | Trustworthy |
| 3 | 0.30 | 0.35 | Malicious |
| 4 | 0.65 | 0.70 | Suspicious |
| 5 | 0.50 | 0.55 | Suspicious |

The trust classification and propagation engine in the proposed Deep SOC framework efficiently assess node behavior in SoC-MANETs, ensuring reliable service discovery even under unreliable and dynamic conditions. By structuring network metrics into inputs for CNN-based trust classification and utilizing an adaptive trust propagation mechanism, the system can dynamically adjust to changing network environments, providing robust protection against malicious nodes. This approach not only enhances trust management but also ensures that the network can maintain its operational integrity over time.

## 5. PROPOSED TIME-WINDOW-BASED SMOOTHING ALGORITHM

The proposed Deep SOC framework incorporates three critical components to enhance the trust management process in dynamic and unreliable SoC-MANETs:

- **Time-Window-Based Smoothing Algorithm**: This algorithm helps to filter out transient anomalies in node behavior by smoothing trust scores over time.
- **Reputation Update Mechanism**: This mechanism allows nodes to update their reputation dynamically based on the interactions with their neighbors.
- **Adaptive Trust Thresholding Mechanism**: This mechanism adjusts the trust threshold dynamically to reflect the evolving network conditions and ensure that only trustworthy nodes are included in service discovery.

These mechanisms work together to ensure that trust values are accurately maintained and adapt to the dynamic environment of SoC-MANETs.

### 5.1 TIME-WINDOW-BASED SMOOTHING ALGORITHM

In a highly dynamic network, trust scores can fluctuate due to short-term anomalies or fluctuations in node behavior, such as temporary packet drops or sudden delay spikes. To address this, the time-window-based smoothing algorithm is employed. It uses a sliding time window to smooth out short-term fluctuations and produce a more stable trust score for each node.

The trust score is computed using a moving average over the last **n** observations (where **n** is the window size). This approach ensures that transient behaviors do not drastically affect the overall trust score. The equation for trust smoothing is as follows:

$$\text{Smoothed Trust}_i(t) = \frac{1}{n}\sum_{k=t-n+1}^{t} \text{Trust}_i(k) \tag{5}$$

where,

*Smoothed Trust$_i$*($t$) is the smoothed trust score of node $i$ at time $t$,

*Trust$_i$*($k$) is the trust score of node $i$ at time $k$,

$n$ is the window size.

This smoothing function helps reduce the impact of anomalies and provides a more reliable trust value.

Table.5. Node Trust Scores with Time-Window-Based Smoothing

| Node ID | Trust Scores (Raw) | Smoothed Trust Score (n=3) |
|---|---|---|
| 1 | [0.80, 0.85, 0.88] | 0.84 |
| 2 | [0.92, 0.90, 0.93] | 0.91 |
| 3 | [0.30, 0.35, 0.40] | 0.35 |
| 4 | [0.60, 0.65, 0.70] | 0.65 |
| 5 | [0.50, 0.55, 0.52] | 0.53 |

### 5.2 REPUTATION UPDATE MECHANISM

The reputation update mechanism is designed to adapt the reputation of each node based on both direct and indirect interactions within the network. Reputation is updated based on the quality of interactions, with higher weights given to recent behavior. When a node interacts with others, it updates its reputation either positively or negatively depending on the success of the interaction (e.g., successfully forwarding packets increases reputation, while dropping packets decreases it). The reputation of node $i$ at time $t$ is updated as follows:

$$\text{Reputation}_i(t+1) = \alpha \cdot \text{Reputation}_i(t) + \beta \cdot \text{Interaction}_i(t) \tag{6}$$

where,

*Reputation$_i$*($t$+1) is the updated reputation of node $i$ at time t+1,

$\alpha$ is the weight for the previous reputation,

$\beta$ is the weight for the interaction quality,

*Interaction$_i$*($t$) is the quality of node $i$s interaction at time $t$.

The reputation update is continuous, and nodes adjust their reputation after each interaction based on the type of behavior observed. Positive interactions increase reputation, while malicious behavior or packet drops lead to a reduction in reputation.

Table.6. Reputation Update Mechanism Example

| Node ID | Previous Reputation | Interaction Quality | Updated Reputation ($\alpha$=0.7, $\beta$=0.3) |
|---|---|---|---|
| 1 | 0.84 | 0.90 | 0.87 |
| 2 | 0.91 | 0.88 | 0.90 |
| 3 | 0.35 | 0.20 | 0.29 |
| 4 | 0.65 | 0.75 | 0.70 |
| 5 | 0.53 | 0.50 | 0.52 |

### 5.3 ADAPTIVE TRUST THRESHOLDING MECHANISM

The adaptive trust thresholding mechanism dynamically adjusts the trust threshold based on the overall network behavior. In stable conditions, a higher threshold ensures that only nodes with consistently high trust scores are considered trustworthy. However, in a highly dynamic environment (with frequent mobility, temporary attacks, or irregular behavior), the threshold is lowered to avoid excluding nodes that might only temporarily behave poorly. The adaptive threshold at time *t* is computed as:

$$\text{Threshold}(t) = \gamma \cdot \left( \frac{1}{N} \sum_{i=1}^{N} \text{Trust}_i(t) \right) \tag{7}$$

where,

*Threshold*(*t*) is the adaptive trust threshold at time *t*,

$\gamma$ is a constant factor (typically between 0.6 and 0.8),

*N* is the total number of nodes,

$Trust_i(t)$ is the trust score of node *i* at time *t*.

By adjusting the threshold based on network-wide trust scores, the system ensures that a reasonable number of nodes are classified as trustworthy even in uncertain or fluctuating conditions.

Table.7. Adaptive Thresholding Example

| Time | Average Trust Score (Network-wide) | Adaptive Trust Threshold ($\gamma$=0.7) | Nodes Classified as Trustworthy |
|---|---|---|---|
| 1 | 0.70 | 0.49 | [1, 2, 4, 5] |
| 2 | 0.65 | 0.46 | [1, 2, 4] |
| 3 | 0.75 | 0.53 | [1, 2, 4, 5] |
| 4 | 0.68 | 0.48 | [1, 2, 4] |
| 5 | 0.72 | 0.50 | [1, 2, 4, 5] |

The combination of the time-window-based smoothing algorithm, the reputation update mechanism, and the adaptive trust thresholding mechanism ensures that the Deep SOC framework can accurately and dynamically assess the trustworthiness of nodes in highly dynamic SoC-MANET environments. The smoothing algorithm filters out transient anomalies, the reputation update mechanism adapts based on node behavior, and the adaptive thresholding mechanism ensures that only reliable nodes are considered trustworthy under fluctuating network conditions. These mechanisms work synergistically to ensure that the network remains secure and functional, even in the presence of mobility, attacks, and unreliable service conditions.

## 6. RESULTS

Simulations were conducted using NS-3.36 on a computing setup with Intel i7 processors, 16 GB RAM, and Ubuntu 22.04 OS. The network topology consisted of 100 mobile nodes with random waypoint mobility over an area of 1000x1000 m² for a duration of 500 seconds. The Deep SOC model was benchmarked against three widely adopted trust-based methods: T-RAT (Trust-aware Routing with Adaptive Thresholds), TRM-FL (Trust and Reputation Model with Fuzzy Logic), and GTMS (General Trust Management Scheme). These models were evaluated under black hole, gray hole, and Sybil attack scenarios. The Deep SOC approach consistently outperformed others in trust convergence, attack detection rate, and service delivery performance, demonstrating enhanced adaptability and robustness.

Table.8. Simulation Parameters for Deep SOC

| Parameter | Value |
|---|---|
| Number of Nodes | 100 |
| Mobility Model | Random Waypoint |
| Simulation Area | 1000 x 1000 m² |
| Simulation Time | 500 seconds |
| CNN Layers | 3 Conv + 2 FC |
| Epochs | 50 |
| Learning Rate | 0.001 |
| Optimizer | Adam |
| Attack Types | Black Hole, Gray Hole, Sybil |
| Trust Update Interval | 5 seconds |
| Trust Threshold | 0.6 |
| Packet Size | 512 bytes |
| Routing Protocol | AODV |

### 6.1 PERFORMANCE METRICS

- **Packet Delivery Ratio (PDR):** Measures the ratio of successfully delivered packets to the total packets sent. High PDR indicates efficient and reliable routing.
- **End-to-End Delay (E2ED):** The average time taken for packets to travel from source to destination. Lower values reflect better real-time performance.
- **Detection Accuracy:** Indicates how well the model can classify nodes as malicious or benign, essential for trust management.
- **Trust Convergence Time:** The time it takes for trust values to stabilize across the network. Faster convergence ensures timely reaction to threats.
- **Energy Consumption:** Assesses the total energy utilized per node during operation. Lower energy usage signifies better sustainability in SoC-MANET environments.

Table.9. Packet Delivery Ratio (PDR)

| Time (seconds) | T-RAT | TRM-FL | GTMS | Proposed Method |
|---|---|---|---|---|
| 0 | 0.75 | 0.70 | 0.68 | 0.80 |

| | | | | |
|---|---|---|---|---|
| 125 | 0.78 | 0.72 | 0.70 | 0.84 |
| 250 | 0.80 | 0.75 | 0.72 | 0.86 |
| 375 | 0.82 | 0.77 | 0.75 | 0.88 |
| 500 | 0.85 | 0.80 | 0.78 | 0.90 |

Table.10. End-to-End Delay (E2ED)

| Time (seconds) | T-RAT | TRM-FL | GTMS | Proposed Method |
|---|---|---|---|---|
| 0 | 320 | 350 | 330 | 280 |
| 125 | 315 | 340 | 325 | 270 |
| 250 | 310 | 330 | 320 | 260 |
| 375 | 305 | 320 | 310 | 250 |
| 500 | 300 | 310 | 300 | 240 |

Table.11. Detection Accuracy

| Time (seconds) | T-RAT | TRM-FL | GTMS | Proposed Method |
|---|---|---|---|---|
| 0 | 0.80 | 0.78 | 0.75 | 0.85 |
| 125 | 0.82 | 0.80 | 0.77 | 0.88 |
| 250 | 0.85 | 0.83 | 0.80 | 0.90 |
| 375 | 0.87 | 0.85 | 0.82 | 0.92 |
| 500 | 0.90 | 0.88 | 0.85 | 0.94 |

Table.12. Trust Convergence Time

| Time (seconds) | T-RAT | TRM-FL | GTMS | Proposed Method |
|---|---|---|---|---|
| 0 | 320 | 350 | 330 | 260 |
| 125 | 310 | 340 | 325 | 250 |
| 250 | 300 | 330 | 315 | 240 |
| 375 | 290 | 320 | 310 | 230 |
| 500 | 280 | 310 | 300 | 220 |

Table.13. Energy Consumption

| Time (seconds) | T-RAT | TRM-FL | GTMS | Proposed Method |
|---|---|---|---|---|
| 0 | 0.85 | 0.80 | 0.78 | 0.75 |
| 125 | 0.84 | 0.78 | 0.76 | 0.72 |
| 250 | 0.82 | 0.76 | 0.74 | 0.68 |
| 375 | 0.80 | 0.74 | 0.72 | 0.65 |
| 500 | 0.78 | 0.72 | 0.70 | 0.62 |

From the results, it is evident that the proposed method outperforms the existing methods across all the metrics analyzed. In terms of Packet Delivery Ratio (PDR), the proposed method shows a steady increase, reaching 0.90 at 500 seconds, compared to T-RAT (0.85), TRM-FL (0.80), and GTMS (0.78). This indicates that the proposed method improves packet delivery under dynamic and unreliable conditions, likely due to its adaptive trust and reputation mechanisms.

The End-to-End Delay (E2ED) for the proposed method consistently remains lower, reaching only 240 ms at 500 seconds, while other methods show higher delays. This demonstrates the efficiency of the proposed approach in reducing communication delays, which is crucial for real-time applications in SoC-MANETs. In terms of Detection Accuracy, the proposed method exhibits a significant improvement, reaching 0.94 by the end of the simulation, compared to the maximum of 0.90 seen in T-RAT. This higher accuracy suggests that the proposed method is better at identifying malicious behaviors in real-time. The Trust Convergence Time of the proposed method is also faster, stabilizing at 220 seconds compared to T-RAT (280 seconds), indicating quicker adaptation to the networks trust changes. Lastly, the Energy Consumption is lower in the proposed method, reaching 0.62 J at 500 seconds, which is less than all existing methods, showing that it is more energy-efficient while maintaining performance.

Thus, the proposed method demonstrates superior performance in key areas such as packet delivery, detection accuracy, delay reduction, trust convergence, and energy consumption, making it a more robust and efficient solution for trust management in dynamic SoC-MANET environments.

Table.14. Packet Delivery Ratio (PDR)

| Number of Nodes | T-RAT | TRM-FL | GTMS | Proposed Method |
|---|---|---|---|---|
| 100 | 0.75 | 0.72 | 0.70 | 0.80 |
| 200 | 0.78 | 0.74 | 0.72 | 0.84 |
| 300 | 0.80 | 0.76 | 0.73 | 0.86 |
| 400 | 0.82 | 0.78 | 0.75 | 0.88 |
| 500 | 0.85 | 0.80 | 0.78 | 0.90 |

Table.15. End-to-End Delay (E2ED)

| Number of Nodes | T-RAT | TRM-FL | GTMS | Proposed Method |
|---|---|---|---|---|
| 100 | 310 | 340 | 330 | 270 |
| 200 | 320 | 350 | 340 | 280 |
| 300 | 330 | 360 | 350 | 290 |
| 400 | 340 | 370 | 360 | 300 |
| 500 | 350 | 380 | 370 | 310 |

Table.16. Detection Accuracy

| Number of Nodes | T-RAT | TRM-FL | GTMS | Proposed Method |
|---|---|---|---|---|
| 100 | 0.82 | 0.80 | 0.78 | 0.85 |
| 200 | 0.84 | 0.82 | 0.80 | 0.88 |
| 300 | 0.86 | 0.84 | 0.82 | 0.90 |
| 400 | 0.88 | 0.86 | 0.84 | 0.92 |
| 500 | 0.90 | 0.88 | 0.85 | 0.94 |

Table.17. Trust Convergence Time

| Number of Nodes | T-RAT | TRM-FL | GTMS | Proposed Method |
|---|---|---|---|---|
| 100 | 310 | 340 | 330 | 250 |
| 200 | 300 | 330 | 320 | 240 |
| 300 | 290 | 320 | 310 | 230 |

| | | | | |
|---|---|---|---|---|
| 400 | 280 | 310 | 300 | 220 |
| 500 | 270 | 300 | 290 | 210 |

Table.18. Energy Consumption

| Number of Nodes | T-RAT | TRM-FL | GTMS | Proposed Method |
|---|---|---|---|---|
| 100 | 0.85 | 0.80 | 0.78 | 0.75 |
| 200 | 0.84 | 0.78 | 0.76 | 0.72 |
| 300 | 0.82 | 0.76 | 0.74 | 0.68 |
| 400 | 0.80 | 0.74 | 0.72 | 0.65 |
| 500 | 0.78 | 0.72 | 0.70 | 0.62 |

From the data, it is evident that the proposed method consistently outperforms the existing methods in all key metrics across varying numbers of nodes. For Packet Delivery Ratio (PDR), the proposed method reaches 0.90 at 500 nodes, showing an increase of 5-10% over T-RAT, TRM-FL, and GTMS. This highlights the superior routing decisions of the proposed method, leading to better packet delivery, even as the network size grows. Regarding End-to-End Delay (E2ED), the proposed method achieves the lowest delay, with a value of 310 ms at 500 nodes, compared to 350 ms for T-RAT and TRM-FL. This demonstrates better efficiency in managing delays under higher traffic and congestion levels.

In terms of Detection Accuracy, the proposed method shows a steady increase, reaching 0.94 at 500 nodes, outclassing the other methods, particularly T-RAT (0.90). This indicates a more accurate detection of malicious behaviors, ensuring the networks security. The Trust Convergence Time is also quicker in the proposed method, stabilizing at 210 seconds compared to the other methods, which need more time to converge. Lastly, Energy Consumption is significantly lower in the proposed method, demonstrating its energy efficiency, which is crucial in large-scale SoC-MANETs.

### Conclusion

The proposed method significantly enhances trust management in SoC-MANETs, providing improvements across multiple performance metrics compared to existing methods like T-RAT, TRM-FL, and GTMS. The superior Packet Delivery Ratio (PDR), lower End-to-End Delay (E2ED), and higher Detection Accuracy suggest that the proposed method effectively handles the challenges of unreliable service discovery in dynamic environments. Its ability to maintain high delivery ratios while minimizing delays and maximizing detection accuracy is crucial for maintaining performance in large-scale networks. Additionally, the proposed methods quicker Trust Convergence Time indicates that it can adapt to network changes faster, ensuring a more responsive system.

The lower Energy Consumption shows that the method is more efficient, making it a viable choice for resource-constrained mobile devices. The improvements in these key areas make the proposed method a more reliable and efficient solution for trust management in SoC-MANETs, particularly in environments with frequent changes in topology, high mobility, and unpredictable network conditions. By addressing these challenges, the proposed approach promises to offer better security, performance, and scalability compared to traditional trust management methods.

## REFERENCES

[1] G. Arulselvan and A. Rajaram, “Routing Attacks Detection in MANET using Trust Management Enabled Hybrid Machine Learning”, *Wireless Networks*, Vol. 31, No. 2, pp. 1481-1495, 2025.

[2] S. Kuang, J. Zhang and A. Mohajer, “Reliable Information Delivery and Dynamic Link Utilization in MANET Cloud using Deep Reinforcement Learning”, *Transactions on Emerging Telecommunications Technologies*, Vol. 35, No. 9, pp. 1-7, 2024.

[3] S.M. Hassan, M.M. Mohamad, F.B. Muchtar and F.B.Y.P. Dawoodi, “Enhancing MANET Security through Federated Learning and Multiobjective Optimization: A Trust-Aware Routing Framework”, *IEEE Access*, Vol. 12, pp. 181149-181178, 2024.

[4] S. Kanthimathi, “Exploring Machine Learning Algorithms for Malicious Node Detection using Cluster based Trust Entropy”, *IEEE Access*, Vol. 12, pp. 137913-137925, 2024.

[5] F. Ullah, A. Salam, F. Amin, I.A. Khan, J. Ahmed, S.A. Zaib and G.S. Choi, “Deep Trust: A Novel Framework for Dynamic Trust and Reputation Management in the Internet of Things (IoT) based Networks”, *IEEE Access*, Vol. 12, pp. 87407-87419, 2024.

[6] L. Hota, B.P. Nayak and A. Kumar, “Machine Learning Algorithms for Optimization and Intelligence in Wireless Networks: WSNs, MANETs, VANETs and USNs”, *Proceedings of International Conference on 5G and Beyond Wireless Communications*, pp. 306-332, 2025.

[7] S.M. Hassan, M.M. Mohamad and F.B. Muchtar, “Advanced Intrusion Detection in MANETs: A Survey of Machine Learning and Optimization Techniques for Mitigating Black/Gray Hole Attacks”, *IEEE Access*, Vol. 12, pp. 150046-150090, 2024.

[8] H. Souissi, M. Mahamat, G. Jaber, H. Lakhlef and A. Bouabdallah, “Analyses of Recent Advances on Machine Learning-based Trust Management for Mobile IoT Applications”, *Proceedings of International Conference on Software, Telecommunications and Computer Networks*, pp. 1-6, 2022.

[9] S.S. Jamaesha, M.S. Gowtham and M. Ramkumar, “Deep Artificial Immune System with Malicious Node Detection and Secure Routing Protocol in MANET”, *Transactions on Emerging Telecommunications Technologies*, Vol. 35, No. 11, pp. 1-8, 2024.

[10] S. Aravindan and A. Rajaram, “Energy-Aware Multi-Attribute Trust Modal for Secure MANET-IoT Environment”, *Multimedia Tools and Applications*, Vol. 83, No. 1, pp. 85637-85662, 2024.

[11] E. Alalwany and I. Mahgoub, “Security and Trust Management in the Internet of Vehicles (IoV): Challenges and Machine Learning Solutions”, *Sensors*, Vol. 24, No. 2, pp. 1-37, 2024.

[12] M. Kaur, D. Prashar, L. Mrsic and A.A. Khan, “Machine Learning-based Routing Protocol in Flying Ad Hoc Networks: A Review”, *Computers, Materials and Continua*, Vol. 82, No. 2, pp. 1615-1643, 2025.

[13] N. Jyothi and R. Patil, “An Optimized Deep Learning-based Trust Mechanism in VANET for Selfish Node Detection”,

*International Journal of Pervasive Computing and Communications*, Vol. 18, No. 3, pp. 304-318, 2021.

[14] Z.A. Abbood, D.C. Atilla and C. Aydin, "Intrusion Detection System through Deep Learning in Routing MANET Networks", *Intelligent Automation and Soft Computing*, Vol. 37, No. 1, pp. 270-281, 2023.

[15] Q. Yuan and Y. Lai, "Towards Efficient Information Retrieval in Internet of Things Environments via Machine Learning Approaches", *Journal of the Institution of Engineers*, pp. 1-24, 2024.

[16] E.S. Krishna, D. Sandeep, R. Kocherla, K.K. Lella, S. Molugu, S.H.S. Ibrahim and R. Vatambeti, "Enhancing Intrusion Detection in MANETs with Blockchain-based Trust Management and Enhanced GRU Model", *Peer-to-Peer Networking and Applications*, Vol. 18, No. 1, pp. 1-22, 2025.