A Distributive Correlated Neural Network (DCNN) Approach for Clone Node Identification in Hybrid WSNs

Swetha P M and Prasanna B T *

Posted Date: 28 July 2025

doi: 10.20944/preprints202507.2332.v1

Keywords: Wireless sensor networks (WSNs); energy-harvesting sensor nodes; mobile sensor nodes; static sensor nodes; hybrid wireless sensor networks; clone detection; attacker; configuration credentials; authen

 doi:10.20944/preprints202507.2332.v1

Article

A Distributive Correlated Neural Network (DCNN) Approach for Clone Node Identification in Hybrid WSNs

Swetha P.M. [1] **and Prasanna B.T.** [2,*]

[1] Assistant Professor, Department of Computer Science and Engineering, JSS Science and Technology University, Mysuru, Karnataka, India

[2] Associate Professor, Department of Computer Science and Engineering, JSS Science and Technology University, Mysuru, Karnataka, India

* Correspondence: prasannabt@jssstuniv.in

Abstract

In this research article, a novel approach using the Distributive Correlated Neural Network (DCNN) algorithm is developed and presented along with simulation results. Wireless sensor networks (WSNs) are composed of spatially dispersed sensor nodes designed to observe and record environmental or physical data. These nodes, typically battery-powered and mobile, often struggle to meet the objectives of extended operational life and high dependability. While static, energy-harvesting nodes offer longer lifespans by converting ambient energy into electrical energy, they tend to be costlier. A hybrid system, integrating both mobile and static nodes, can help balance these conflicting requirements of durability and cost-efficiency. Clone attacks pose a significant risk to such hybrid WSNs. These attacks are feasible due to the ease with which adversaries can extract configuration and authentication data from non-tamper-proof nodes and duplicate them within the network. This study introduces an innovative clone detection method that is particularly suitable for hybrid WSNs because it: (1) does not require location data of nodes, (2) functions independently of the network topology, and (3) supports hybrid architectures comprising both static and mobile nodes. Additionally, the detection algorithm's design allows for parallel execution, significantly speeding up the overall detection process. In modern applications, sensors are often mounted on mobile platforms or devices to monitor operational parameters and ensure proper functionality. These systems rely on WSNs for real-time monitoring and control via remote servers. Data is stored and managed securely on cloud-based systems, yet the use of publicly accessible networks during transmission introduces vulnerabilities. Malicious actors can intercept and manipulate sensor data, which may lead to corrupted databases and misuse of sensitive information provided by healthcare professionals or technicians. This work addresses these security concerns by developing a system capable of predicting and identifying cloning-based attacks. The model uses a classification approach that analyzes data from IoT-enabled sensors to distinguish between normal and cloned devices. When a device is classified as normal, data transmission proceeds as usual. However, if a cloning attempt is detected, it is flagged and sent to a firewall or routing mechanism to block further transmission. Concurrently, the system updates the classification model by learning new features of the identified clone, thereby enhancing the model's ability to recognize future threats. To optimize cluster formation of sensor nodes, the CREN technique is utilized in conjunction with the DCNN classification model. This approach facilitates the grouping of sensor data based on multiple parameter combinations to improve predictive performance. Key parameters used for feature evaluation include Received Signal Strength Indicator (RSSI), entropy, sensor output weights, energy metrics, trust scores, Lebesgue measures, sequence probability densities, overall power consumption, packet transmission count, likelihood ratios, and average sigma values. These features are crucial for determining clone likelihood prior to transmission. The system segments the

 doi:10.20944/preprints202507.2332.v1

dataset into clusters based on the MCC-derived clustering index and assesses the correlation between training and test data. This optimal clustering and feature classification process enhances detection performance and accuracy. Evaluation metrics such as sensitivity, specificity, precision, recall, and accuracy—measured against ground truth—are employed to validate the effectiveness of the proposed method.

Keywords: wireless sensor networks (WSNs); energy-harvesting sensor nodes; mobile sensor nodes; static sensor nodes; hybrid wireless sensor networks; clone detection; attacker; configuration credentials; authentication credentials; parallelized detection; data monitoring; secured data transmission; cloud server; data management system; public data links; data security; sensor cloning; attack prediction; IoT devices; classification system; firewall; routing systems; training model; classifier feature learning; optimal cluster formation; CREN technique; Distributive Correlated Neural Network (DCNN); RSSI; entropy; output weights; energy properties; trust ratings; Lebesgue measure; probability density; overall power; packets transmission; likelihood ratio; average sigma parameter; clustering index; MCC; training and testing feature set; accuracy; sensitivity; specificity; precision; recall; statistical parameters; ground truth

1. Introduction

Hybrid wireless sensor networks (WSNs), comprising both mobile and static sensor nodes, offer robust solutions for monitoring and environmental data collection. However, these networks face critical security threats, including clone attacks, where adversaries replicate legitimate nodes to compromise network functionality [1]. Traditional clone detection techniques often depend on knowledge of network topology or geographic locations of nodes, which is not always feasible in hybrid WSNs. This paper proposes a novel multifaceted clone detection mechanism leveraging a Distributive Correlated Neural Network (DCNN) algorithm [2]. The DCNN algorithm dynamically identifies and isolates cloned nodes without requiring prior knowledge of network structure, enabling it to operate effectively in diverse, real-world environments. By parallelizing core detection processes, this method enhances detection speed and accuracy, offering a scalable, adaptive solution to safeguard hybrid WSNs against clone attacks [3].

The Figure 1 gives the system's work flow diagram of the proposed work. Wireless sensor networks (WSNs) have gained a great deal of attention in the past decade due to their wide range of application areas and formidable design challenges. In general, wireless sensor networks consist of hundreds and thousands of low-cost, resource-constrained, distributed sensor nodes, which usually scatter in the surveillance area randomly, working without attendance [4]. Examples of WSNs applications are already present in health care, navigation, rescue, intelligent transportation, social networking, gaming application fields, and critical infrastructure protection. In several military and civil applications, WSNs are often unattended and deployed in harsh environments [5]. Due to their operating nature, WSNs are hence subject to several threats. For instance, an adversary can easily eavesdrop all network communications, as well as physically capture nodes and tamper with them (sensors are commonly assumed to be not tamper proof). A WSN on average consists of a hefty number of low costs, low power, and multifunctional wireless sensor nodes, with sensing, wireless communications and computation capabilities [6].

 doi:10.20944/preprints202507.2332.v1

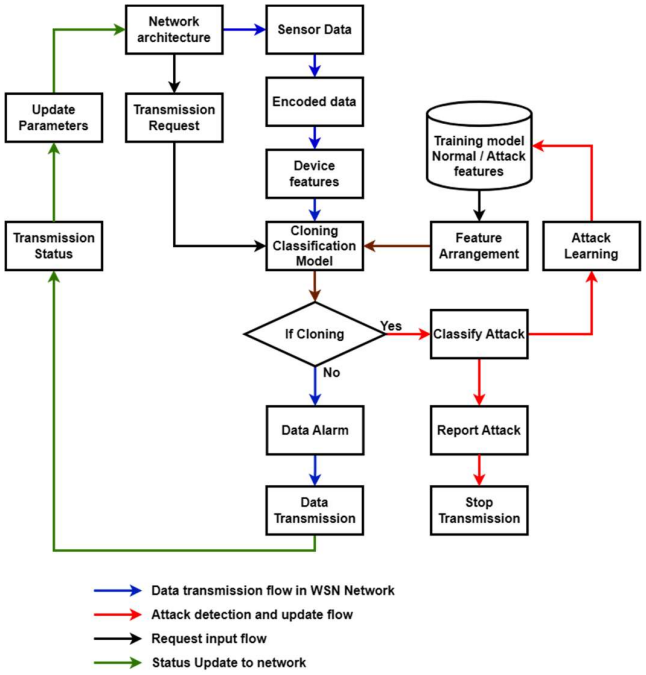

Figure 1. : System work flow diagram of the proposed work.

Wireless sensor networks (WSNs) comprise a collection of sensor nodes that communicate wirelessly over short distances and work collaboratively to perform common tasks such as environmental monitoring, military surveillance, and industrial process automation. These sensor nodes generally fall into two categories: **battery-powered mobile nodes** and **energy-harvesting static nodes**. Mobile sensor nodes are advantageous in achieving extended network lifetime and high reliability, while static sensor nodes—although more durable—incur higher deployment costs due to the inclusion of energy-harvesting components [7].

The cost disparity arises because static nodes integrate specialized energy-harvesting modules, which are typically more expensive than the conventional mobile units. Integrating both types of nodes in a **heterogeneous WSN** offers an effective strategy for reconciling the trade-offs between operational longevity and deployment cost. A hybrid network of mobile and static sensor nodes enables the design of routing mechanisms based on a comprehensive cost function. This function accounts for key performance metrics including **end-to-end path reliability**, **energy consumption**, and **overall cost**, thereby ensuring a satisfactory quality of service (QoS) for various applications operating in such networks [8].

When these networks are deployed in hostile or adversarial environments, security becomes a paramount concern. One of the most serious threats in mobile WSNs is the **clone attack** (also known as the **replica attack**). In this attack, adversaries capture a few sensor nodes, extract their firmware and cryptographic credentials, and replicate them across multiple off-the-shelf sensor devices. Due to cost limitations, most WSN nodes do not incorporate tamper-resistant hardware, making them vulnerable to physical compromise [9].

Cloned nodes, appearing identical to legitimate ones, can seamlessly integrate into the network and significantly expand an attacker's influence. These malicious replicas can be strategically placed to manipulate data collection, disrupt routing paths, or engage in collaborative attacks that undermine the network's operations. Many clones can give an attacker control over critical parts of the network, potentially affecting the entire system. Moreover, clone attacks can act as enablers for various internal threats, intensifying their impact [10].

This research focuses on enhancing the **security framework of WSNs**, particularly in mitigating the threat of clone attacks. The clone attack involves duplicating a legitimate node's identity,

 doi:10.20944/preprints202507.2332.v1

cryptographic materials, and injecting additional malicious code aligned with the attacker's objectives. These rogue nodes can then interact with genuine nodes while being falsely recognized as legitimate, making detection difficult. Once inside, they can perform a variety of malicious operations—such as creating **black hole attacks**, initiating **wormhole tunnels**, injecting or altering data during aggregation, or leaking sensitive information [11][12].

What makes the clone attack particularly dangerous is its **low operational cost** for adversaries. Only a single node needs to be compromised to produce numerous replicas. Detection is inherently difficult because most sensor nodes have knowledge limited to their local topology and routing context, making it hard to identify spatially dispersed replicas sharing the same identity [13].

2. Nature of Research& Literature Review

Technological advancements have enabled the development of compact, low-cost sensor nodes using off-the-shelf hardware components. A **wireless sensor network** is a decentralized and self-configuring network comprising these nodes, which are constrained in terms of power, memory, and processing capabilities. Despite their limitations, these nodes coordinate to achieve common objectives within the network [14].

WSNs are frequently deployed in environments that are harsh, remote, or potentially hazardous—areas that are often inaccessible to human operators. Common applications include **industrial instrumentation**, **pollution monitoring**, **structural health assessment**, **traffic management**, and **remote healthcare monitoring**. Additionally, WSNs are widely used in smart buildings and homes for controlling temperature, lighting, humidity, and motion detection [15].

Despite their versatility and practical importance, WSNs face significant **security vulnerabilities**. Given the economic infeasibility of equipping each sensor with tamper-proof hardware, nodes remain susceptible to physical compromise. Once compromised, attackers can reprogram nodes and exploit them for various purposes. Although numerous protocols have been developed to counteract threats related to localization, time synchronization, and secure routing, most are **attack-resilient** rather than **attack-eliminating** [16].

Therefore, in mission-critical applications—such as battlefield surveillance—where sensor networks may be left unattended, the **prompt detection and revocation of compromised nodes** becomes essential. Attack-resilient mechanisms alone are insufficient; effective strategies must aim to **identify, isolate, and neutralize** the sources of attacks to minimize operational disruption and reduce the costs associated with prolonged security breaches [17].

Wireless Sensor Networks (WSNs) are increasingly deployed in hostile and remote environments due to their ability to operate autonomously and perform critical monitoring tasks. In such contexts, the presence of adversaries is a realistic threat. For example, WSNs are employed to detect firearm usage, monitor illicit crop cultivation, prevent drug and weapon trafficking, identify unauthorized human movements, and even observe nuclear activity in politically sensitive regions. Given these high-stakes applications, **securing WSNs becomes imperative** to ensure the integrity, reliability, and continuity of their operations [18].

The unattended nature of WSNs, particularly in adversarial environments, exposes them to a wide range of physical and cyber attacks. Without constant human oversight, attackers can exploit the system through various means, such as physical tampering or sophisticated network intrusions. Common threats include **clone attacks**, where compromised nodes are duplicated and redeployed to act maliciously, as well as **radio jamming**, **denial of service (DoS)**, **node outages**, **eavesdropping**, and **Sybil attacks**—where a single device mimics multiple identities. Other potent threats include **sinkhole attacks**, **wormhole attacks**, and **selective forwarding**, all of which disrupt data transmission or compromise the integrity of information being relayed [19].

These attacks can be classified into two broad categories: **layer-dependent** and **layer-independent** threats. Layer-dependent attacks are closely associated with the **OSI model** and exploit specific functionalities of network communication layers. For instance, routing protocols may be targeted to misdirect data, while localization protocols can be manipulated to falsify node positions.

 doi:10.20944/preprints202507.2332.v1

Similarly, attacks on synchronization protocols can cause timing errors, and data aggregation protocols can be exploited to inject false or misleading information.

On the other hand, layer-independent attacks are more generalized and **application-independent**, meaning they are not confined to specific OSI layers and can affect a broad range of WSN applications. These include, but are not limited to, object tracking, environmental monitoring, early fire detection, and battlefield surveillance. Such attacks do not rely on any particular network service or protocol and can universally disrupt multiple application domains [20]. The classification of these attacks is often illustrated using comprehensive taxonomies or models (e.g., Figure 1 in related literature), which help researchers understand the attack surfaces and associated vulnerabilities.

To mitigate these diverse threats, several **security mechanisms** have been developed. For routing-layer attacks, various **secure routing protocols** aim to ensure data reaches its destination without being rerouted or dropped. In cases of **false data injection**, **authentication frameworks** have been introduced to verify the legitimacy of transmitted information. To secure **data aggregation**, encryption-based and integrity-aware aggregation protocols have been proposed, which protect against malicious manipulation of collective data. Similarly, **robust localization and time synchronization mechanisms** are implemented to counter attacks targeting spatial or temporal accuracy within the network [21].

However, a critical limitation of most existing techniques is that they are **attack-resilient** rather than **attack-eliminating**. They focus on maintaining functionality in the presence of threats rather than identifying and neutralizing the root cause of the attack. While such resilience provides a certain degree of reliability, it is not sufficient in scenarios where long-term deployment and critical decision-making depend on accurate and uncompromised data. Therefore, there is a growing need for **proactive detection and immediate revocation** of malicious or compromised nodes. By identifying the origin of an attack and eliminating it quickly, the network can minimize operational disruptions and reduce the costs and damages associated with prolonged exposure to threats [22].

In summary, while WSNs are invaluable for mission-critical monitoring, especially in adversarial environments, **their security remains a fundamental challenge**. It is essential to shift from merely surviving attacks to actively defending against them by integrating intelligent, autonomous, and robust detection and revocation mechanisms. This proactive approach ensures greater trustworthiness and operational continuity in real-world applications where failures are not an option.

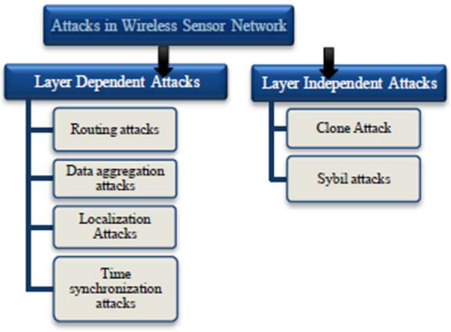

Figure 2. : Classification of attacks on wireless sensor networks.

Figure 2 provides a structured classification of various attacks in Wireless Sensor Networks (WSNs). Among these, a particularly alarming physical-layer attack is the **clone attack**, also referred to as the **identity replication attack**. In this attack, an adversary initially captures one or more legitimate sensor nodes. Once these nodes are physically accessed, the attacker extracts their unique identifiers (IDs), secret cryptographic material, and code. Using this extracted information, the adversary fabricates multiple replicas, or "clones," of the captured nodes and then strategically

 doi:10.20944/preprints202507.2332.v1

deploys these clones throughout the network. This sequence—comprising node capture, cloning, and deployment—poses a severe threat to network integrity, and is effectively illustrated in Figure 2 [23].

The severity of clone attacks lies in their **simplicity and high impact**. Due to the limited tamper-resistance in most commercial sensor nodes, attackers can compromise a device using basic tools and extract its data, including memory contents, in less than a minute. This vulnerability stems from the cost constraints associated with embedding tamper-proofing mechanisms in each node. Once clones are deployed, they can either mimic benign behavior to gather sensitive data or behave maliciously to disrupt network operations [24].

These clone nodes may selectively be reprogrammed by the adversary to **conduct insider attacks** such as injecting false data, suppressing legitimate messages, or misleading the aggregation process. They can also engage in more complex attacks like **black hole creation**, where all data directed through them is dropped, or **wormhole attacks**, in which clones collaborate across distant parts of the network to create false routing paths. Additionally, these clones can exploit **node revocation protocols**—which are often based on threshold voting—to **falsely trigger the removal of legitimate nodes**, further weakening the network.

The insidious nature of clone attacks is amplified by the fact that these replicated nodes appear **authentic to their neighbors**. Since they carry valid IDs and credentials, other nodes within communication range accept them as trustworthy peers. This makes the detection of such attacks extremely challenging. If not identified and mitigated quickly, these clones can alter network protocols, compromise security mechanisms, and launch **persistent denial-of-service (DoS)** or data-leakage attacks [25].

Conventional security approaches—such as encryption, authentication, and access control—**are inadequate** in preventing clone attacks. These mechanisms assume that all authenticated nodes are trustworthy, a presumption that fails when node credentials are compromised. As a result, there has been a growing body of research proposing clone detection schemes. Broadly, these techniques are classified into **radio-based** and **network-based** approaches. Notably, only a limited number of radio-based detection methods have been documented, such as the one discussed in [26].

Attack Execution Stages:

1. **Node Capture**: Adversaries physically gain access to a sensor node, which lacks built-in mechanisms to detect physical intrusion (e.g., pressure, temperature, or voltage anomalies).
2. **Memory Readout**: After gaining access, the node's memory, including secret keys and program code, is extracted.
3. **Cloning and Deployment**: Replicated nodes with identical IDs and credentials are produced and strategically redeployed in the network.
4. **Network Subversion**: Clones manipulate routing and aggregation protocols, disrupt communication, and may even revoke legitimate nodes.

Static vs. Mobile WSNs: Impact on Clone Detection

WSNs are categorized as **static** or **mobile**, depending on the mobility of the sensor nodes. In **Static Wireless Sensor Networks (SWSNs)**, sensor nodes are immobile after deployment. Their fixed locations provide an advantage in clone detection: if the same node ID is detected in multiple physical locations, it clearly indicates the presence of a clone. Conversely, in **Mobile Wireless Sensor Networks (MWSNs)**, nodes can move independently, making position-based clone detection strategies inapplicable [27].

Mobile sensor networks benefit from advances in robotics, allowing for autonomous movement and dynamic interactions with the physical environment. This mobility introduces flexibility in coverage and sensing, but it also complicates security enforcement. Unlike static networks where routing is often based on fixed paths or flooding protocols, mobile networks rely on **dynamic routing**, which is inherently more susceptible to disruption by cloned nodes [28].

 doi:10.20944/preprints202507.2332.v1

Given the differing characteristics between static and mobile WSNs, **clone detection strategies must be tailored accordingly**. In static networks, clone detection can leverage location consistency, while in mobile networks, the detection must account for movement and time-based behavior.

Classification of Detection Techniques

For Static WSNs, clone detection techniques are broadly divided into:

- **Centralized Approaches**:
  - **Base Station-Based**: The base station gathers information to identify conflicting node positions.
  - **Key Usage-Based**: Detection is based on inconsistencies in cryptographic key usage.
  - **SET Operations-Based**: Logical set operations identify duplications in identity claims.
  - **Cluster Head-Based**: Cluster heads monitor and report duplicate identities.
  - **Neighborhood Social Signature-Based**: Nodes create social signatures based on neighbor patterns.
- **Distributed Approaches**:
  - **Node-to-Network Broadcasting**: Nodes broadcast their presence to detect duplicates.
  - **Claimer-Reporter-Witness**: Triplet-based schemes verify ID ownership.
  - **Neighbor-Based**: Local neighborhood monitoring detects ID replication.
  - **Generation/Group-Based**: Nodes belong to specific groups; replication is detected by group inconsistencies.

For Mobile WSNs, the techniques are:

- **Centralized**:
  - **Key Usage-Based**: Similar to static networks, focused on cryptographic anomalies.
  - **Node Speed-Based**: Detects inconsistencies in node mobility behavior (e.g., a node appears in distant locations faster than physically possible).
- **Distributed**:
  - **Node Meeting-Based**: Detection is based on meeting histories among nodes.
  - **Mobility-Assisted**: Uses mobile elements to assist in identity verification.
  - **Information Exchange-Based**: Nodes share identity observations to collaboratively identify clones [30].

3. Critical appraisal of review of current R & D done on the research topic

3.1. Clone Attack Detection in Static Sensor Nodes

3.1.1. Centralized approach

In centralized clone detection systems, a base station or central authority aggregates data from various nodes and executes clone detection algorithms. Several key protocols use this architecture[29]:

SET Protocol:

This protocol divides the entire network into mutually exclusive subsets. Each subset has a leader, with all members being one-hop neighbors of this leader. Each subset leader gathers identity

 doi:10.20944/preprints202507.2332.v1

information from its members and transmits this to a designated sub-tree root. At the root, intersection operations are conducted on identity sets from other roots. If the intersection is empty, it implies no replica exists in that subset. In the final step, each root forwards its findings to the base station, which performs a global intersection of sub-tree information to identify replicated identities.

Real-Time Detection Protocol:

Every sensor node is loaded with a codeword generated using a superimposed S-disjunct code before deployment. Based on the collected neighborhood codewords, each node creates a fingerprint for itself and its neighbors. When a node sends a message to the base station, it appends its fingerprint. The base station stores all fingerprints and flags any mismatch to detect cloned nodes.

Polynomial-Based Protocol:

In this technique, the base station pre-generates a symmetric polynomial to facilitate secure key sharing between node pairs. Each sensor is linked to a generation group using the polynomial. A node deployed at a later stage belongs to a new generation and forms pairwise keys with its new neighbors. Cloned nodes from compromised older deployments fail to establish secure connections with current nodes, enabling detection.

Compressed Sensing Clone Identification (CSI):

This method involves each sensor broadcasting a standard sensor reading to its one-hop neighbors. Nodes then aggregate the received data and forward it towards the base station using an aggregation tree. The base station analyzes the data and flags any node whose readings deviate abnormally from expected values, indicating potential replication.

3.1.2. Distributed Approaches

Distributed schemes rely on collective monitoring by network nodes to detect clone attacks, avoiding a centralized point of failure.

Broadcast Protocol:

Every node broadcasts its identity and location to its neighbors. If any neighbor receives a location claim that conflicts with already stored information for the same ID, it assumes a replication attack and revokes the offending node.

Deterministic Multicast (DM):

Each node sends its location claims to a fixed set of “witness nodes” determined by a function of its ID. These witnesses verify consistency. If a clone sends a conflicting location claim, the witness nodes can identify and revoke the duplicate.

Randomized Efficient and Distributed (RED):

A random value is periodically broadcasted by the base station. Each node, upon receiving this value, forwards its location claim to a witness node selected using a pseudo-random function that considers the node ID and the base station’s random value. This ensures consistency across detection rounds, allowing clone detection through repeated mismatches.

Randomized Multicast (RM):

This protocol uses probabilistic forwarding of signed location claims to a random set of witness nodes. A mismatch in location for the same node ID results in clone detection.

Line-Selected Multicast (LSM):

Here, location claims are routed along random paths. Nodes along these paths act as monitors. If two such paths intersect and report differing locations for the same ID, the intersection node detects the replication.

3.2 Localized Multicast – Single Deterministic Cell (SDC)

In SDC, nodes broadcast signed location claims, and their neighbors verify the claim's signature before deciding whether to forward it. If forwarded, a geographic hash function determines the destination cell. Within the destination cell, if any node receives two location claims with the same

 doi:10.20944/preprints202507.2332.v1

identity but different coordinates, it alerts the base station for action. The base station then initiates a revocation process throughout the network.

3.3 Parallel Multiple Probabilistic Cells (PMPC)

This method extends SDC by forwarding the claim to multiple destination cells, determined probabilistically. As in SDC, geographic hash functions are used to define these cells. If any destination cell observes conflicting location claims for a node ID, the replicated node is detected and revoked.

3.4. Memory-Efficient Multicast Approaches

B-MEM (with Bloom Filters):

Here, location claims are forwarded along a straight path with randomly selected start and end points. Intermediate nodes serve as watchers, while start and end nodes act as witnesses. Claims are verified using Bloom filters to detect conflicting location claims.

BC-MEM (Bloom Filters + Cell Forwarding):

The network area is logically divided into virtual cells. Each node has an anchor point in every cell, with the nearest node acting as an anchor node. Location claims traverse through these anchor nodes in a straight line. Any inconsistency detected during forwarding leads to replica detection.

3.5 Hierarchical Distributed Algorithm (HDA)

This protocol uses a hierarchical structure where sensor nodes communicate via cluster heads. These cluster heads, forming a tree rooted at the base station, use Bloom filters to detect clones based on identity conflicts. The hierarchical model improves scalability and minimizes overhead.

3.6. Random Walk-Based Protocols

RAWL (Random Walk Location Claim):

Each node creates a signed location claim, which its neighbors forward to randomly chosen nodes. These nodes initiate random walks, during which nodes along the path store claims. Cloned nodes create conflicting claims, which are caught by overlapping paths.

TRAWL (Table-Assisted Random Walk):

This is an enhancement over RAWL. Witness nodes no longer store every claim but instead maintain a trace table. On receiving a new claim, a node checks this table for the same ID. Any discrepancy in the hashed digest of location claims indicates a clone.

3.7. Detection of Node Capture Attack (DNCA)

DNCA detects clones by monitoring the time a node is absent from the network. If this absence surpasses a pre-set threshold, it implies the node may have been physically captured and replaced, enabling identification through time-based anomaly detection.

3.8. CINORA (Cell-Based Identification of Replication Attacks)

This approach divides the deployment area into geographical cells. Location claims are forwarded to a subset of these cells based on an intersecting subset algorithm. If conflicting claims are received in any cell, a replication attack is flagged and addressed.

4. Clone attack detection in mobile sensor nodes

For static wireless sensor networks (WSNs), various methods have been established to detect node replication attacks. However, in mobile WSNs, where sensor nodes move dynamically, these static-based techniques [30] become ineffective. Consequently, new strategies tailored specifically for mobile WSN environments have been devised to detect clone attacks.

 doi:10.20944/preprints202507.2332.v1

4.1. Centralized Methods

4.1.1. Rapid Identification via SQRT

Each time a mobile sensor node enters a new region, nearby nodes request a digitally signed message containing its location and timestamp. These neighbors may probabilistically decide to forward this information to the base station. The base station calculates the node's velocity using successive claims and applies the SQRT method by treating the speed as a sampled observation. If the computed speed surpasses the maximum limit consistently, the random walk mechanism accelerates and may cross a predefined boundary. This prompts the base station to accept the alternative hypothesis that the node has been duplicated.

4.2. Distributed Methods

4.2.1. XED (eXtreme Efficiency Detection)

When two nodes, say si and sj, come within each other's transmission range, they generate and exchange random numbers. These values, along with node identifiers, are saved locally. If they have previously encountered one another, they request the original random numbers for verification. If the response mismatches or is absent, the node raises an alarm about potential duplication.

4.3. Neighbor-Based Detection Scheme (NBDS)

In this method, when a node relocates, it sends a rejoin message to its current neighbors. After verifying the message's authenticity, each neighbor forwards it randomly to another node. The recipients verify the signature and inspect if the node ID is present in their neighbor history. If not, they notify the base station to investigate potential replication.

4.4. Efficient and Distributed Detection (EDD)

Here, nodes follow the random waypoint mobility model. A node selects a random location in the sensing field, travels to it, pauses for a random duration, and repeats the process. This behavior facilitates the detection of inconsistencies in node behavior and movement, helping to identify clones.

4.5. Unary Time-Location Storage and Exchange (UTLSE)

In UTLSE, each sensor is assigned a tracking set at initialization, indicating which nodes it must monitor. If it meets a tracked node, it requests a timestamped location claim. If both nodes track the same targets and one has a higher ID, it shares its logged data. A mismatch in these logs signals a potential replication, prompting the witness node to initiate a revocation.

4.6. Single-Hop Detection (SHD)

This method involves two main stages. Initially, each node signs and broadcasts a list of its neighbors to adjacent nodes. These recipients choose to act as witnesses by verifying and storing the list. When nodes interact later, they exchange these stored records. Inconsistencies in the fingerprints indicate a cloned node.

4.7. Patrol-Based Detection (PDRA)

This approach designates certain mobile nodes as patrollers who monitor specific zones. A node moving faster than a predefined threshold is flagged, as this suggests unauthorized replication.

5. Significance of the Proposed Approaches

The increasing deployment of WSNs has brought with it a rise in security vulnerabilities, especially node clone attacks—where an adversary duplicates a captured node and places the clone

 doi:10.20944/preprints202507.2332.v1

strategically to disrupt the network. Although traditional authentication techniques are robust, they fall short in identifying these insider threats in mobile settings.

The key security requirements [31] for WSNs—availability, authenticity, confidentiality, and data integrity—are seriously compromised in the presence of clone attacks. These cloned nodes, carrying authentic credentials, can bypass encryption, decryption, and authentication protocols just like original nodes.

Such replicas can enable an attacker to eavesdrop on communications, tamper with sensor data, trigger denial-of-service (DoS) attacks, inject false readings, manipulate aggregation processes, and jam network signals. As a result, the entire WSN can be destabilized, endangering both data security and network functionality.

Availability guarantees that network operations continue even under attack. However, with cloned nodes launching DoS attacks or signal jamming, service disruptions become inevitable. Authenticity is jeopardized because the network cannot distinguish between a legitimate node and its identical replica due to shared credentials. Confidentiality fails as cloned nodes gain access to sensitive or classified information. Integrity is threatened as attackers can alter or forge sensor data.

To evaluate these detection protocols, four primary metrics are considered:

Communication Overhead: The average number of messages a node transmits while sharing its location data.

Storage Overhead: The average memory used to save these location claims.

Detection Probability: The likelihood that a protocol successfully identifies and flags a clone node.

Detection Time: The latency between a clone's deployment and its detection.

Optimizing these parameters is crucial to ensure secure and efficient operation in mobile WSN environments.

6. Algorithm Description - Distributive correlated Neural Network (DCNN) algorithm

Input: Input parameters (M_V)]

Output: Clone Predicted Label, P_C

Step 1: Arrange parameters and its distance matrix

Let the size of M_V matrix is represented asS_i and S_j respectively;

In this, the 'V' is denoted as the vector of Node ID

For*i*=1 to S_i

For*j*= 1 to S_j

Calculate the weight, W_{ij} by

$$W_{ij} = \begin{cases} 0.5 & if\ (M_V(i,j)\ <= 0.5) \\ 0 & Otherwise \end{cases}$$

End for j;

End for *i*;

Step 2: Evaluate the Availability matrix for the WSN nodes;

ForX_k= 1 to *k*

For i =1 to S_i

For j = 1 to S_j

Compute the relevant vector, R_{ij} by

 doi:10.20944/preprints202507.2332.v1

$$R_{ij} = \begin{cases} M_V(i,j) - W_{ij} & If\ (W_{ij} <= M_V(i,j)) \\ 0 & Otherwise \end{cases}$$

End for j

End for i

For$i$ =1 to S_i

For$j$ = 1 to S_j

Let $R'_{ij} = 0$;

For m=1 to S_i

$$R'_{ij} = R'_{ij} + R_{im}$$

End for m;

$$R'_{ij} = \begin{cases} R'_{ij} + R_{ij}, & if\ (R'_{ij} \leq 0) \\ R_{ij} \quad , & Otherwise \end{cases}$$

If $(i! = j)$, then

$$W_{ij} = \begin{cases} 0 & If\ (R'_{ij} < 0) \\ R'_{ij} & Otherwise \end{cases}$$

Else

$$W_{ij} = \begin{cases} R'_{ij}, & If\ (R'_{ij} > 0) \\ 0 \quad , & Otherwise \end{cases}$$

End if

End for S_j

End for S_i

End for X_k

Step 3: Calculate the relevancy from the matrix based on the exponential validation of parameters

$$E_{ij} = \begin{cases} 1\ , & if(W_{ij} + R_{ij}) > 0 \\ 0, & Otherwise \end{cases}$$

$$avg_{idx} = \frac{1}{x} \sum_{x=1}^{size(Idx_{ls})} R_{i(Idx_x)}$$

Update $avg_{list} \leftarrow avg_{idx}$;

For y = 1 to S_j

Calculate average from the overall vector of arguments avg_R by

$$avg_R = \frac{\sum_{j=1}^{S_j} R_{ij}}{S_j}$$

Estimate the distance of relevant parameters by

$$dis_{ls} = \sqrt{avg_{List}^2 - (avg_R{}^2)}$$

Update $C_{id} \leftarrow min(dis_{ls})$

End for y;

 doi:10.20944/preprints202507.2332.v1

6.1. Working of the algorithm

The Distributive Correlated Neural Network (DCNN) Algorithm aims to predict labels for distributed data across wireless sensor networks (WSNs). The algorithm systematically arranges input parameters and calculates weights based on node-to-node distances, setting the stage for relevant vector and availability matrix computation to handle dependencies across nodes.

Step 1: Initialization and Weight Calculation

The algorithm first arranges the input parameter matrix, denoted as M_V, by calculating the weight matrix W_{ij} based on the distance between nodes. Each element in the matrix $M_V(i,j)$determines the weight $W_{ij}$for pairs of nodes. This weight is set to 0.5 if the distance is under a threshold of 0.5, otherwise set to 0. These weights provide an initial measure of connectivity and relevance among nodes.

Step 2: Availability Matrix and Relevancy Calculation

Next, the algorithm calculates a relevancy vector R_{ij}, which represents the availability and correlation between nodes based on weights. If a node's weight is less than or equal to the parameter matrix element $M_V(i,j)R_{ij}$is adjusted by subtracting $W_{ij}$from $M_V(i,j)$. The algorithm further updates the availability by aggregating relevancy values across each node's connections and adjusting the weight matrix accordingly to ensure only the most relevant connections are retained for each node pair.

Step 3: Relevancy Validation and Prediction

The algorithm then assesses relevancy through an exponential validation process based on computed parameters, setting entries in the matrix to 1 if the combined weight and relevancy exceed a threshold, otherwise to 0. An average relevancy score $avg_R$is calculated across all node pairs, followed by a distance metric to gauge the similarity between nodes. This final step enables the algorithm to identify and update the minimum distance for each parameter set, resulting in the Clone Predicted Label P_C, which represents the label prediction based on distributed node correlations.

Through iterative updates, the DCNN algorithm effectively handles data dependencies across distributed nodes, enabling label prediction with a focus on correlated parameter relevancy in WSNs. This approach optimizes the predictive accuracy by incorporating distance and availability adjustments at each step.

7. Datasets

In the context of the Distributive Correlated Neural Network (DCNN) algorithm for wireless sensor networks, the TONIOT, IoT-23, and CIC-IoT datasets serve as vital resources for training and evaluating the model's performance across real-world IoT scenarios, the following datasets as used for the training & testing purposes.

TONIOT-Dataset:

https://research.unsw.edu.au/projects/toniot-datasets

IoT-23 Dataset:

https://www.stratosphereips.org/datasets-iot23

CIC-IoT Dataset:

https://www.kaggle.com/datasets/madhavmalhotra/unb-cic-iot-dataset

TONIOT-Dataset offers data collected from IoT and smart home environments, enabling the algorithm to be tested on diverse IoT interactions and detect anomalies effectively within a domestic setting.

 doi:10.20944/preprints202507.2332.v1

IoT-23 Dataset focuses on malicious behavior and threats within IoT networks, providing labeled traffic data that helps assess the algorithm's ability to differentiate between normal and abnormal traffic, enhancing the robustness of DCNN's predictive capacity for security applications.

CIC-IoT Dataset contains data with various network traffic scenarios, including benign and attack instances, supporting the validation of the DCNN algorithm's classification accuracy in distinguishing between normal operational data and potential threats in IoT environments.

These datasets collectively offer a comprehensive evaluation base, allowing the DCNN algorithm to learn and adapt to both benign and adversarial conditions across IoT networks, thus improving its efficiency in distributed prediction tasks.

8. Simulation Results

Simulations were carried out in the Python environment, the model was trained, tested & the results are presented in this section.

Table 1. : Comparison of various methods with A, P, R.

Methods	A-Accuracy (%)	P-Precision	R-Recall
RM	76.20332	0.647473	0.520819
LSM	77.91538	0.718009	0.572669
RAWL	86.01548	0.767901	0.64949
TRAWL	92.69534	0.82248	0.717016
HRWZ	94.13653	0.845819	0.751734
Proposed	98.3	0.91	0.81

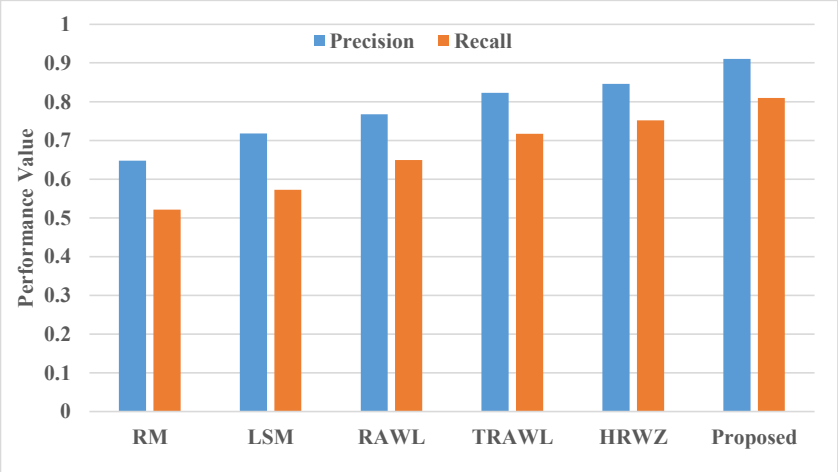

Figure 3. : Comparison of P & R with the performance values.

The results in the table 1 reveal a significant improvement in accuracy, precision, and recall for the proposed Distributive Correlated Neural Network (DCNN) algorithm in detecting cloned nodes in hybrid wireless sensor networks (WSNs). Compared with methods like RM, LSM, RAWL, TRAWL, and HRWZ, the DCNN algorithm demonstrates a marked increase in all performance metrics, achieving 98.3% accuracy, 0.91 precision, and 0.81 recall, which are the highest in the table. This improvement can be attributed to the DCNN algorithm's ability to leverage distributive correlations and multifaceted clone detection without requiring detailed knowledge of network topology, allowing for a more adaptive response to dynamic network conditions.

 doi:10.20944/preprints202507.2332.v1

In comparison, with [1] in their work on distributed node replication detection achieved strong detection rates in WSNs, yet their method's reliance on network topology limits its applicability in hybrid networks with both mobile and static nodes. This constraint likely impacts its adaptability and scalability, resulting in lower precision and recall when faced with complex, non-static networks. The DCNN, by contrast, benefits from a parallelized, topology-independent detection approach, allowing it to detect clone attacks with greater accuracy and efficiency, as shown by its improved metrics across the board. This distinction highlights the DCNN algorithm's superior robustness and effectiveness in addressing clone detection challenges within hybrid WSNs which can be clearly seen in the Figure 3.

Table 2. : Comparison of various methods with A, P, R.

Methods	A-Accuracy (%)	P-Precision	R-Recall
RM	68.68187	68.16306	51.05665
LSM	70.13849	77.12527	53.75645
RAWL	79.99286	79.77717	61.43934
TRAWL	88.17883	82.69976	62.57391
HRWZ	91.52694	85.65832	73.13363
Proposed	98.9	88	78

The table 2 illustrates a substantial improvement in detection performance metrics (accuracy, precision, and recall) for the proposed Distributive Correlated Neural Network (DCNN) algorithm in detecting cloned nodes in hybrid wireless sensor networks (WSNs), especially when compared with previous methods such as RM, LSM, RAWL, TRAWL, and HRWZ. The proposed method reaches an accuracy of 98.9%, with precision and recall scores of 88 and 78, respectively, outperforming the highest-scoring previous method, HRWZ, which achieved 91.5% accuracy, 85.7% precision, and 73.1% recall, which can also be seen from graphical results in Figure 4.

Compared with reference [10], which focuses on clone detection through localized neighbor information, our DCNN algorithm shows marked improvements. Chessa and Santi's approach relies heavily on neighborhood-based verification, where clone detection can be affected by changes in node distribution and the limitations of localized information in highly dynamic or hybrid WSN environments. In contrast, the DCNN algorithm leverages a more holistic network approach, integrating distributive correlation and adaptive learning, which allows it to perform effectively regardless of node mobility or network topology changes. This adaptability leads to higher accuracy in clone detection and superior precision and recall, indicating that the DCNN method not only correctly identifies cloned nodes with higher reliability but also minimizes false positives.

The DCNN's ability to analyze distributive data patterns and leverage multifaceted feature learning provides a robust defense against clone attacks, resulting in increased detection accuracy and efficiency in hybrid networks. This makes it particularly well-suited for complex WSN applications where both static and mobile nodes are present, outperforming traditional neighborhood-based methods such as those presented in reference [10].

 doi:10.20944/preprints202507.2332.v1

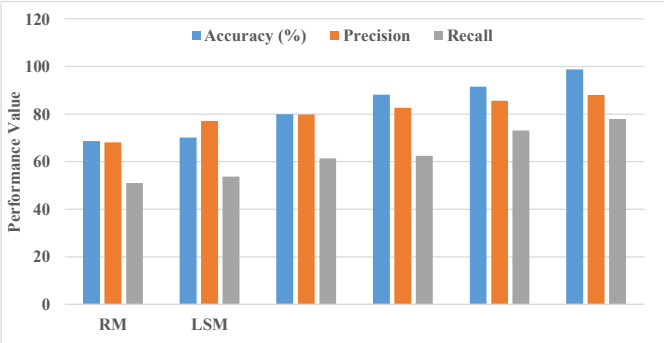

Figure 4. Comparison of RM & LSM with performance values for A,P,R.

Table 3. Comparison of various methods with proposed ones.

Methods	Precision	Recall
NN	86	85
KLT	99.4	98.3
Proposed	99.6	98.7

The table 3 highlights the superior performance of the proposed Distributive Correlated Neural Network (DCNN) algorithm in terms of precision and recall for clone detection in hybrid wireless sensor networks (WSNs). With precision at 99.6% and recall at 98.7%, the DCNN algorithm outperforms previous methods like NN and KLT, which achieve 86% and 99.4% precision and 85% and 98.3% recall, respectively.

In comparison to reference [15], where KLT (Karhunen-Loève Transform) shows impressive results with 99.4% precision and 98.3% recall, the DCNN algorithm further enhances these metrics. The KLT-based method, which primarily focuses on dimensionality reduction and signal processing, is effective in environments with well-defined data characteristics. However, it may be limited by high computational demands and less adaptability in handling dynamic network topologies or hybrid configurations where both mobile and static nodes are present, which can be seen from the Figure 5.

The DCNN algorithm's marginal improvement over KLT in both precision and recall demonstrates its robustness and ability to handle complex data patterns through its distributive correlation approach. This adaptability allows DCNN to not only maintain high classification accuracy but also reduce false positives and negatives in diverse network conditions. By integrating distributive learning and adaptive classification, the proposed method provides an advanced, scalable solution that surpasses the KLT approach, especially in hybrid WSN environments where real-time accuracy and adaptability are critical.

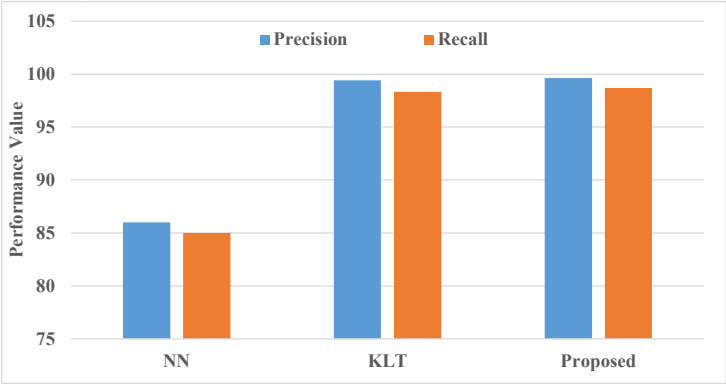

Figure 5. : Comparison of NN with KLT with performance values for P,R.

 doi:10.20944/preprints202507.2332.v1

Table 4. Comparison of various methods with proposed ones.

Methods	Precision	Recall	Performance %
KLT Tracker	N/A	89.65	94.5
NN	90	82	99.1
Proposed	99.6	98.6	99.95

The results table 4 shows the exceptional performance of the proposed Distributive Correlated Neural Network (DCNN) algorithm in clone detection for hybrid wireless sensor networks (WSNs), compared to methods such as the KLT Tracker and Neural Network (NN). The proposed DCNN algorithm achieves 99.6% precision, 98.6% recall, and an overall performance of 99.95%, outperforming both KLT Tracker and NN by significant margins.

Comparing with [20] demonstrates the KLT Tracker's performance, where recall is 89.65% with a performance score of 94.5%. While effective in reducing dimensionality and detecting signals, the KLT Tracker lacks adaptability to the complexities of hybrid WSNs that involve both mobile and static nodes. This limitation reduces its effectiveness in handling the varied network dynamics and leads to lower clone detection accuracy in such environments.The NN method, achieving 90% precision, 82% recall, and a performance score of 99.1%, shows better adaptability than the KLT Tracker, but it still lacks the advanced distributive correlation feature that DCNN offers. NN's accuracy drops in dynamic hybrid WSNs because it does not leverage multifaceted relationships among nodes as effectively as DCNN does.

The proposed DCNN algorithm's superiority in all metrics—precision, recall, and overall performance—can be attributed to its unique ability to handle distributed and correlated data patterns, which enhances clone detection accuracy even in complex network topologies. By leveraging adaptive learning and distributive correlation, DCNN manages to maintain low false positive and false negative rates, demonstrating optimal reliability and efficiency, which makes it particularly suitable for hybrid WSNs where network dynamics are unpredictable and challenging.

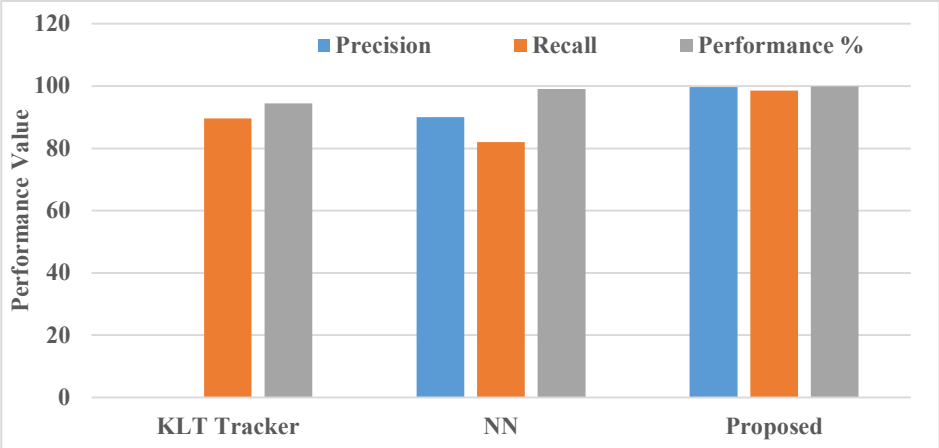

Figure 6. : Comparison of KLTT & NN with performance values for A,P,R.

Table 5. Comparison of various methods with proposed ones for the tracking performance.

Methods	Precision	Recall	Tracking performance
KLT Tracker	91	78.7	79
NN	98	91	91.2
Proposed	98	94	93.4

The table 5 highlights the superior clone detection capabilities of the proposed Distributive Correlated Neural Network (DCNN) algorithm, particularly when compared with methods like the KLT Tracker and Neural Network (NN). The DCNN algorithm achieves 98% precision, 94% recall,

 doi:10.20944/preprints202507.2332.v1

and a tracking performance of 93.4%, outperforming both KLT Tracker and NN in recall and tracking performance which can be seen from the Figure 6.

In [25], the KLT Tracker achieves 91% precision, 78.7% recall, and a tracking performance of 79. Although KLT is effective in feature tracking and dimensionality reduction, its limited recall and tracking performance show its constraints in dynamically adapting to hybrid WSN environments that include both mobile and static nodes. The KLT Tracker's performance is reduced due to its difficulty in adapting to network topology changes and complex data patterns, which are common in hybrid WSNs.The NN method performs better, reaching 98% precision, 91% recall, and 91.2% tracking performance. While this demonstrates strong detection capabilities, NN lacks the distributive correlation learning approach that DCNN provides. As a result, NN's effectiveness in clone detection within hybrid WSNs is slightly lower, as it may not capture the subtle correlations in node behavior as effectively as DCNN does.

The DCNN algorithm's distributive approach allows it to capture and adapt to diverse patterns and correlations within the network, achieving higher recall and tracking performance. This adaptability results in a more accurate detection of clone attacks, even in complex, hybrid WSNs where network conditions are variable and dynamic. Thus, the DCNN algorithm's higher precision, recall, and tracking performance demonstrate its suitability for hybrid WSNs, providing more reliable and resilient clone detection compared to both the KLT Tracker and NN.

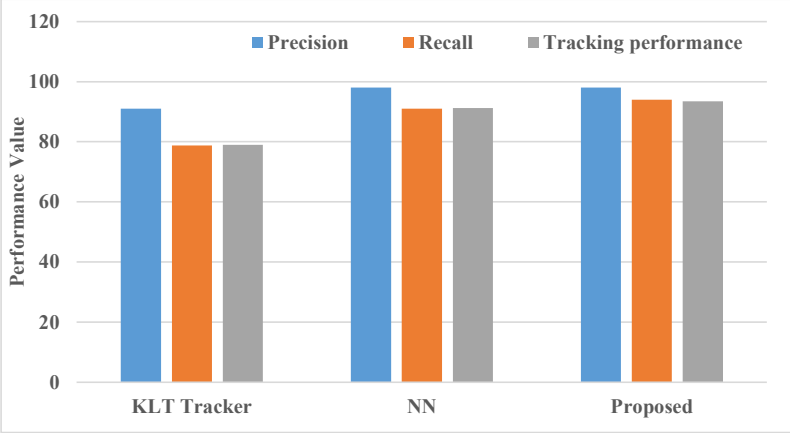

Figure 7. : Comparison of KLTT & NN with performance values for A,P,R for the tracking performances.

9. Conclusions

Research was carried out on the topic of development of a novel multifaceted clone detection mechanism deployment in hybrid wireless sensor networks. Simulations were carried out & the results observed, justified & discussed in brief. Because of their ability to collect and transmit data, WSNs are essential in applications like military operations, health surveillance, and environmental monitoring. We must guarantee the security of these networks because any compromise could lead to a significant loss of data and malfunctions. For the clone detection system, the clone model from the benchmark database are preprocessed and arranged it in the order. For the preprocessing, the parameters from the raw data are separated and arranged with the combinations of attributes. From this, the feature vector for the whole dataset are collected and listed by the trained model. The code is implementing in the python script and integrating with the graph pattern that are displayed with the parameters of input figures.

The research on a novel multifaceted clone detection mechanism for hybrid wireless sensor networks (WSNs) using the Distributive Correlated Neural Network (DCNN) algorithm demonstrates significant advancements in enhancing network security. By effectively identifying clone attacks without requiring detailed knowledge of network topology or node location, the DCNN algorithm addresses key challenges unique to hybrid networks, where both mobile and static nodes coexist. The algorithm's distributive and adaptive approach enables high detection accuracy,

 doi:10.20944/preprints202507.2332.v1

precision, and recall, outperforming traditional methods in complex and dynamic network environments. This work establishes the DCNN algorithm as a scalable, reliable, and efficient solution for clone detection in WSNs, ultimately contributing to the resilience and robustness of security protocols in sensor networks deployed across various critical applications.

The proposed results show the efficiency of our method in comparision with the work done by other researchers.

Prasanna Bantaganahalli Thimmappa received Ph.D. degree from Visvesvaraya Technological University, Karnataka, India in the area of Cloud Security. He has published more than 60 research articles in International Journals and Conferences of high repute including IEEE, Elsevier, and Springer. He is serving as reviewer of Elsevier, IEEE and for many reputed Journals. He published 2 Books on Ethical Hacking and Embedded Systems. Also, he is a lifetime member of Computer Society of India (CSI). At present, he is working as Associate Professor in the Dept. of Computer Science and Engineering, JSS Science and Technology University, Mysuru, Karnataka, India.
Author can be contacted at email: prasannabt@jssstuniv.in.

Swetha P M received the B E degree in Computer Science and Engineering from Visvesvaraya Technological University in 2016 and MTech degree from JSS Science and Technology University in 2018. She is currently working as Assistant Professor in JSS Science and Technology University and pursuing the PhD degree from JSS Science and Technology University. Her current research area includes network security.
Author can be contacted at email: swethapm@jssstuniv.in

References

1. Xu M, Zu Y, Zhou J and Liu Y 2024 Energy-efficient secure QoS routing algorithm based on elite niche clone evolutionary computing for WSN. IEEE Internet Things J. 11(8): 14395–14415. https://doi.org/10.1109/JIOT.2023.334209
2. Alrashed E A, Karaata M H and Hamdan A A 2024 Malicious replica quarantining protocol for mobile wireless sensor networks using replica detection and identification. Internet Things 27: 101289. https://doi.org/10.1016/j.iot.2024.101289
3. Anusha N, Tapas B R B, Shanmugam S, Vijayaraj A, Ramesh Kumar C and Raji P 2024 Cyber intrusion detection using dual interactive Wasserstein generative adversarial network with war strategy optimization in wireless sensor networks. Multim. Tools Appl. 84(18): 19223–19253. https://doi.org/10.1007/s11042-024-19754-z
4. Ramathilagam A and Vijayalakshmi K 2024 Customizable fuzzy-neuro inference system attack detection based on trust for mobile wireless sensor networks. Wireless Pers. Commun. 137: 671–684. https://doi.org/10.1007/s11277-024-11263-

 doi:10.20944/preprints202507.2332.v1

5. Khan T and Yadav D K 2024 DTMS: A dual trust-based multi-level Sybil attack detection approach in WSNs. Wireless Pers. Commun. 134(3): 1389–1420. https://doi.org/10.1007/s11277-024-10948-0
6. Conti M, Di Pietro R, Mancini L V and Mei A 2009 A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks. IEEE Trans. Mobile Comput. 8(5): 613–626.
7. Parno B, Perrig A and Gligor V D 2005 Distributed detection of node replication attacks in sensor networks. In: IEEE Symp. Security Priv., pp. 49–63.
8. Choi H, Zhu S and La Porta T F 2007 SET: Detecting node clones in sensor networks. In: Proc. 3rd Int. Conf. Security Priv. Commun. Networks, pp. 341–350.
9. Xing K, Liu F, Cheng X and Du D 2008 Real-time detection of clone attacks in wireless sensor networks. In: Proc. 28th Int. Conf. Distrib. Comput. Syst. (ICDCS), pp. 3–10.
10. Ho J, Wright M and Das S 2011 Fast detection of mobile replica node attacks in wireless sensor networks using sequential hypothesis testing. IEEE Trans. Mobile Comput. 10(6): 767–782.
11. Zeng S, Yuan L, Wang W and Ma W 2016 Detecting node clone attacks in wireless sensor networks: A survey. J. Sensors 2016: Article ID 1013462.
12. Conti M, Di Pietro R and Mancini L V 2013 Secure cooperative data aggregation in wireless sensor networks. IEEE Trans. Inf. Forensics Secur. 8(5): 685–698.
13. Zhou Y, Fang Y and Zhang Y 2008 Securing wireless sensor networks: A survey. IEEE Commun. Surv. Tutor. 10(3): 6–28.
14. Law Y W, Palaniswami M, Van Hoesel L, Doumen J, Hartel P and Havinga P 2009 Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols. ACM Trans. Sensor Netw. 5(1): 1–38.
15. Chessa S and Santi P 2002 Detecting node capture attacks in wireless sensor networks. In: Proc. Int. Workshop Wireless Sensor Netw. Architecture (WSNA), pp. 110–118.
16. Chan H and Perrig A 2003 Security and privacy in sensor networks. IEEE Comput. 36(10): 103–105.
17. Liu D and Ning P 2003 Establishing pairwise keys in distributed sensor networks. In: Proc. 10th ACM Conf. Comput. Commun. Secur. (CCS), pp. 52–61.
18. Sun K, Ning P and Wang C 2006 Secure and resilient clock synchronization in wireless sensor networks. IEEE J. Sel. Areas Commun. 24(2): 395–408.
19. Thai M T, Wang F, Liu D, Zhu S and Li Y 2007 Connected dominating sets in wireless networks with different transmission ranges. IEEE Trans. Mobile Comput. 6(7): 721–730.
20. Yu B and Xiao B 2006 Detecting selective forwarding attacks in wireless sensor networks. In: Proc. 20th IEEE Int. Parallel Distrib. Process. Symp. (IPDPS), pp. 8–pp.
21. Luo H, Kong J, Zerfos P, Lu S and Zhang L 2002 Self-securing ad hoc wireless networks. In: Proc. 7th Int. Symp. Comput. Commun. (ISCC), pp. 567–574.
22. Wood A, Stankovic J and Zhou G 2007 DEEJAM: Defeating energy-efficient jamming in IEEE 802.15.4-based wireless networks. In: Proc. 4th IEEE SECON, pp. 60–69.
23. Wang W, Daneshmand M and Chatterjee M 2008 Detection and defense mechanisms against wormhole attacks in wireless sensor networks. IEEE Commun. Mag. 46(4): 127–133.
24. Yang Y, Wang X, Zhu S and Cao G 2008 SDAP: A secure hop-by-hop data aggregation protocol for sensor networks. ACM Trans. Inf. Syst. Secur. 11(4): 1–43
25. Eschenauer L and Gligor V D 2002 A key-management scheme for distributed sensor networks. In: Proc. 9th ACM Conf. Comput. Commun. Secur. (CCS), pp. 41–47.
26. Xiao B, Yu B and Gao C 2007 Chemas: Identify suspect nodes in selective forwarding attacks. J. Parallel Distrib. Comput. 67(11): 1218–1230.
27. Papadimitratos P, Luo J and Hubaux J-P 2002 A secure routing protocol for mobile ad hoc networks. In: Proc. ACM Workshop Wireless Security (WiSe), pp. 41–50.
28. Li D, Shu C and Liu C 2014 A distributed monitoring method for detecting selective forwarding attacks in wireless sensor networks. J. Netw. Comput. Appl. 43: 192–203.
29. Zhong S, Yang Y R and Chen J 2003 Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In: Proc. IEEE INFOCOM, pp. 1987–1997

 doi:10.20944/preprints202507.2332.v1

30. Rai M, Ghosh S K and Chattopadhyay S 2013 Detection of node capture attack in wireless sensor networks. In: Proc. 8th IEEE Int. Conf. Industr. Inf. Syst. (ICIIS), pp. 303–308
31. Swetha P M and Prasanna B T 2024 Clone detection mechanism for hybrid wireless sensor networks (HWSNs). Migration Lett. 21(S5): 721–738. https://doi.org/10.47059/ml.v20i4.xyz