# A Realtime Adaptive Trust Model Based on Artificial Neural Networks for Wireless Sensor Networks

Khaled Mohammed Ali Hassan*, Mohamed Ashraf Madkour and Sayed Abd El Hady Nouh

*Al-Azhar University, Faculty of Engineering, Computers and Systems Engineering Department, Cairo, Egypt*
*E-mail: khaledhassan22@azhar.edu.eg; mamadkour@azhar.edu.eg; Snouh@azhar.edu.eg*
**Corresponding Author*

Received 04 December 2022; Accepted 25 March 2023;
Publication 21 June 2023

## Abstract

Wireless sensor networks (WSNs) are vulnerable to security attacks due to the unbounded nature of the wireless medium, restricted node resources, and cooperative routing. Standard cryptography and authentication mechanisms help protect against external attacks, but a compromised node can easily bypass them. This work aims to protect WSNs against internal attacks, which are mostly launched from compromised nodes to disrupt the network's operation and/or reduce its performance. The trust and reputation management framework provides a routing cost function for selecting the best secure next hop. Tuning the trust weights is essential to cope with the constant changes in the network environment, such as the sensor nodes' behaviours and locations. To allow real-time operation, the proposed framework introduces an artificial neural network (ANN) in each sensor node that automatically adjusts the weights of the considered trust metrics according to the WSN state. A large dataset is generated to train and test the ANN using a multitude of simulated

*Journal of Cyber Security and Mobility, Vol. 12_4,* 519–546.
doi: 10.13052/jcsm2245-1439.1244

cases. A prototype is developed and tested using the J-Sim simulator to show the performance gain resulting from applying the adaptive trust model. The experimental results showed that the adaptive model has robust performance and has achieved an improved packet delivery ratio with reduced power consumption and reduced average packet loss. The results showed that when sensor nodes were static and malicious nodes were present, the average accuracy was 99.6%, while when they were in motion, it was 88.1%.

**Keywords:** WSNs, trust and reputation management models, ATSR, ML, ANN, backpropagation algorithm, routing cost function (RCF), and internal attacks.

## 1 Introduction

WSNs are a cost-effective and viable solution for a variety of applications due to their distributed, low-cost, fault-tolerant, self-organizing, and scalable nature [1–5]. Low-powered, small-sized sensor nodes can work together efficiently to monitor and collect data from any environment and send it to a destination [6, 7]. However, WSNs are constrained by energy, processing, memory, topology, mobility, and lifetime [8]. While WSNs are responsible for information sensing and aggregation in important application areas such as big data, cloud computing, and the internet of things (IOTs), they are vulnerable to malicious attacks due to their characteristics, such as being deployed in open and harsh environments, using open mediums, and having resource limitations [4]. So, they require a high level of security to protect information and resources from threats and inappropriate behaviour [9]. These attacks can be classified as internal or external [10, 11]. In any case, WSNs are vulnerable to a large set of internal attacks, such as grey holes, black holes, sinkholes, replayed routing information, wormholes, hello floods, acknowledgment spoofing, Sybil attacks, and so on [4, 12–17]. Even though WSNs need to establish a secure path for data from source to destination, they have limited resources and communication bandwidth, which makes it hard to defend against routing attacks [8, 18, 19]. Actually, traditional security mechanisms are not suitable for WSNs due to higher computational costs, high processing speeds, large memory, and communication overheads. Furthermore, traditional security mechanisms cannot effectively resolve internal or misbehaviour node attacks caused by captured sensor nodes [2, 4, 14, 17, 20]. So, trust and reputation management systems have been suggested to help WSNs detect abnormal activities and improve

their security. They are effective for detecting malicious nodes and ensuring security and can be used to solve security issues for routing protocols in WSNs [2, 4, 11, 13, 14, 21, 22]. Although geographic routing algorithms that combine trust and reputation information with location information are based on the greedy perimeter stateless routing (GGPSR) [16, 23, 24], the ambient trust sensor routing (ATSR) model takes a distance metric into account. The ATSR model is a location-based and trust-aware routing protocol to support scalability and mobility in WSNs [25–36]. So, the ATSR is a fully distributed trust and reputation management system that relies on both direct and indirect trust information to calculate a routing cost function (RCF). It uses a static weighted sum approach, where direct historical interaction information and indirect recommendation information are added together in a weighted way to get the total trust value. Then, the total trust and the distance metric are added together in a weighted way to get the RCF [25–36]. The traditional ATSR model has used static weights, which are neither practical nor flexible in a dynamic network environment. Thus, it is hard to be sure that the ATSR trust evaluation is always correct. This means the model cannot keep up with changes in the network environment, reducing performance measures. In the present work, a novel integrated framework for the ATSR model based on artificial neural networks (ANNs) is proposed to overcome the problems mentioned above. An ANN is a type of supervised learning that aims to mimic human thinking and resolve complicated problems automatically without human intervention or reprogramming [40, 44, 46]. A trained ANN is developed and employed to continuously provide a plausible estimation of the weights used in computing the RCF based on the instantaneous network state as measured by selected metrics. This makes the evaluation of the RCF both intelligent and accurate. ANNs are a good choice for WSNs because of their efficiency, robustness, parallelism, and noise tolerance, which are important in these kinds of environments [12, 37].

In essence, we have explored a methodology to automatically calculate trust, reputation, and distance metrics for sensor nodes in the ATSR model. We have improved the traditional ATSR model by using a backpropagation artificial neural network (ATSR-ANN) to predict the RCF of sensor nodes and choose a neighbour with the highest RCF as the best secure next-hop for packet forwarding.

A prototype is developed and tested using the J-Sim simulator to show the performance gain resulting from applying the adaptive trust model as compared with the traditional trust model that uses fixed trust weights. The experimental results showed that the adaptive model has robust performance

and has achieved an improved packet delivery ratio with reduced power consumption and reduced average packet loss, especially when the sensor nodes are in motion.

The remainder of this paper is structured as follows: Section 2 surveys the related work. Section 3 briefly describes the original static ATSR model, showing its merits and limitations, while Section 4 specifies the work objective and the problem statement. Section 5 gives the development steps to design and train an ANN for dynamically setting the trust weights of the ATSR model for WSNs. Simulations and results are given in Section 6, and Section 7 concludes the paper.

## 2 Related Works

Smart trust and reputation management models are briefly mentioned in this section. Ideas from these models were found useful in designing the enhanced ATSR model.

- The research [38] has proposed a novel machine-learning misbehaviour detection methodology in vehicular ad hoc networks (VANET). The suggested model consists of four main phases: data acquisition, data sharing, analysis, and decision-making. In order to efficiently identify the misbehaviour data, new features are derived that represent the misbehaviour, environment, and communication state. An effective misbehaviour classifier is trained based on historical data that includes both attacker and normal traffic data by using ANN methods, which include the feed-forward and backpropagation algorithms. The results showed that the detection rate for all vehicles was 99%, while the false positive rate was 0.23%. F-Measure suggests that the proposed model is effective, with an average F-measure of 98%.
- The authors of this paper [39] have implemented a proposed algorithm for a routing protocol based on ANNs, with their approach focused on improving clustering performances. Their solution was based on the ANNs tool and the LEACH routing protocol. They have introduced the criterion of the consumed energy for the process of electing the cluster head (CH), where the sensor node with the highest level of energy is selected to be the cluster head. The results showed that their LEACHNN performed better than LEACH, saving energy and extending the network's life. The network's performance was around 11% in terms of power consumption.

- A method to apply ANNs to solve the trust problem in ad-hoc networks was developed in [40]. This research aims to demonstrate that ANNs can be used to evaluate trust in ad-hoc networks, specifically for detecting untrusted nodes and estimating trust levels. The packet delivery ratio (PDR) is used as a metric of the trust value. Simulation experiments showed that an ANN can perform a regression analysis by estimating the PDR value of every node in a given network. The used ANN is trained to give two possible types of outputs for every node. The first type is a binary value to indicate whether a given node is trusted. The second type is a continuous number between 0 and 1, representing the node's predicted trust value. A series of simulation experiments were done to evaluate the performance of the proposed method. The classification problem had an average accuracy of 98%, and the regression problem had an accuracy of 94%.
- A novel dynamic trust model is proposed in [41] as one type of decision support approach, using a radial basis function artificial neural network to decide the trust level and mitigate the number of unreliable downloads. The recommended trust model is applied broadly to help peers download from reliable providers. The results showed that the RBF neural network model was 92% accurate.
- An enhanced trust model that uses a radial base artificial neural network (RBANN) is proposed in [37] to predict the future behaviour of each node. The prediction is based on the node's weighted direct and indirect behaviours and provides a trust model that aids in the detection and elimination of malicious nodes within a WSN.

## 3 The Original Static ATSR Model

The ATSR is a routing model for WSNs that combines a fully distributed trust and reputation management model with a location-based routing approach to protect against routing attacks [28]. The following briefly describes the ATSR algorithm, showing its merits and limitations.

### 3.1 ATSR Operation

The ATSR algorithm can detect malicious nodes and react by avoiding using them in routing. It uses a static (fixed) weighted sum approach to compute the RCF for each of its neighbour nodes based on location coordinates, trust, reputation, and remaining energy. The ATSR model uses a watchdog

mechanism to periodically monitor neighbouring nodes' activities to collect observations (direct information) and calculates node trust and reputation metrics using a beta distribution [29].

### 3.2 Direct Trust Quantification

In the ATSR model, each sensor node monitors the behaviour of its one-hop neighbours regarding specific behavioural aspects. Table 1 lists the ATSR trust and reputation metrics that a sensor node uses to compute the direct trust (DT) values for its neighbours. These metrics are used to detect a sensor

**Table 1** The list of the inputs (trust and reputation metrics) that apply to the ANN

| NO. | Trust Metric | Metric Description |
|---|---|---|
| 1 | Packet forwarding | To detect a sensor node that refuses to forward or selectively forward packets (black-hole, gray-hole, denial of service, and selfish behaviour) [26–28]. |
| 2 | Network layer-ACK | To check the successful end-to-end forwarding of packets to detect colluding attackers. That means verifying all kinds of drops for the full path [26–28]. |
| 3 | Packet precision (integrity) | To verify whether a packet was forwarded without unexpected modifications (to detect all types of modification) [26–28]. |
| 4 | Authentication | A sensor node's ability to support authentication. If it does the value is 1, and if it does not the value is 0 [26–28]. |
| 5 | Confidentiality | The ability to encrypt. The value is 1 if a sensor node supports confidentiality, and 0 if it does not [26–28]. |
| 6 | Reputation responses | To identify selfish nodes and test the implementation sincerity of the reputation protocol [26–28]. |
| 7 | Reputation validation | A reputation value is obtained from a third sensor node (observed by a third party). This could be a bad-mouthing attack or a false-praise attack [26–28]. |
| 8 | Remaining energy (Battery lifetime) | A measure of the battery lifetime to avoid a sensor node with a high trust value dying out early and load balancing, such as a traffic analysis attack [26–28]. |
| 9 | Data Link Layer–ACK | The hop-to-hop ACK between neighbours in the data link layer is used to detect sensor nodes trying to stop or delete data link ACKs so that packets have to be resent (new metric). |
| 10 | Confidence Factor | The confidence factor is a threshold in reputation validation and is also used to balance direct and indirect trust to reach the total trust value in the traditional ATSR model [26–28]. |
| 11 | Distance Metric ($D_{min}$/$\mathrm{d_j}$) | The ratio of the Euclidean distance from the nearest neighbour to the destination ($D_{min}$) over the Euclidean distance between neighbour j and the destination ($\mathrm{d_j}$). The shortest distance to the destination maximizes the distance metric value [26–28]. |

node's desire for collaboration in terms of reputation exchange and routing. The aim of each trust metric is to detect and avoid one or more routing attacks. The remaining energy metric is obtained by periodically exchanging BEACON messages and is part of the trust model [25–36]. The ATSR model uses the remaining energy metric to do load balancing in the network to prevent a sensor node with a high trust value from dying out too soon.

- Equation (1) shows the trust value for the remaining energy metric, where $V_{initial}$ and $V_{now}$ represent levels of remaining energy reported by the first and last packets received from a neighbour.

$$T_{RE} = \frac{V_{now}}{V_{initial}} \tag{1}$$

- Both the confidentiality and authentication metrics are set to 1 if a sensor node supports a high-security system, otherwise, they are both set to 0.
- For the other five metrics, Equation (2) gives the trust value for each metric m (denoted as $T_m^{i,j}$) at sensor node i regarding the neighbour node j, where: $S_m^{i,j}$ is the number of successful metric m events that a node i has measured for a node j, and $F_m^{i,j}$ is the number of failed metric m events that a node i has measured for node j.

$$T_m^{i,j} = \frac{S_m^{i,j}}{S_m^{i,j} + F_m^{i,j}} \tag{2}$$

- Equation (3) uses a weighted sum of the computed trust values to get the overall direct trust value for each neighbour, where $W_m$ is the importance (weight) of the trust metric m.

$$DT^{i,j} = \sum_{m=1}^{8} (W_m * T_m^{i,j}) \tag{3}$$

The main problem with the ATSR model is that the weights Wm are set manually for each scenario. This means that the weights have to be changed every time the network environment changes.

### 3.3 Indirect Trust Quantification

Equation (4) uses a weighted approach to compute the indirect trust value $IT^{i,j}$ passed from a neighbor node j to node i, where:

- Nm is neighbouring nodes to node i.

- n is the number of neighbouring nodes that provided reputation responses to sensor node i (the ATSR model uses n = 4 to reduce overhead).
- $DT^{Nm,j}$ is the IT of sensor node j which is provided by nodes Nm.
- $W(DT^{i,Nm})$ is a weighting factor reflecting a node's i DT value of nodes Nm.

$$IT^{i,j} = \sum_{m=1}^{n} W(DT^{i,\ Nm}) * DT^{Nm,j} \tag{4}$$

### 3.4 Total Trust Quantification

In Equation (5), the total trust (TT) for a neighbour j is calculated as the sum of the direct and indirect trust values for that neighbour. The direct and indirect trust values are balanced by a confidence factor $C^{i,j}$.

$$TT^{i,j} = C^{i,j} * DT^{i,j} + (1 - C^{i,j}) * IT^{i,j} \tag{5}$$

The confidence factor $C_{i,j}$ of node i considering a neighbour node j is calculated based on Equation (6), where:

- $N^{i,j}$ is the number of interactions between nodes i and j.
- M is a fixed integer. The ATSR model uses a value of M = 1.

$$C^{i,j} = \frac{N^{i,j}}{N^{i,j} + M} \tag{6}$$

### 3.5 Distance Routing Metric Quantification

The distance metric $D_m^{i,j}$ of each neighbor to the destination is calculated by using the Equation (7), where:

- $d_j$ is the Euclidean distance between neighbour j and the destination.
- $D_{min}$ is the Euclidean distance from the nearest neighbour to the destination.

$$D_m^{i,j} = \frac{D_{min}}{d_j} \tag{7}$$

### 3.6 Routing Cost Function Quantification

Equation (8) computes the RCF between node i and its neighbour node j, where $W_d$ and $W_t$ represent the significance of distance and trust criteria.

The RCF value is in the range of 0 to 1, and the neighbour having the highest RCF value would be selected for routing.

$$RCF^{i,j} = W_d * D_m^{i,j} + W_t * TT^{i,j} \tag{8}$$

## 4 Work Objective and Problem Statement

The current work investigates the possibility of enhancing the traditional ATSR model to improve routing security in WSNs. Instead of the frequent manual adjustment of the protocol's static weights, we aim to find an automatic way to continually adjust the weights to suit the dynamic network environment and, hence, be more confident that the ATSR trust evaluation is always computed using the proper weights. Consequently, the enhanced ATSR model would keep up with changes in the network environment, hoping to improve the performance measures of the model compared with the traditional ATSR model that uses fixed weights.

Section 5 proposes a smart method to continuously adjust the trust weights in real-time based on the instantaneous values of several important WSN metrics using an ANN.

## 5 Realtime ATSR Model for WSNs

The ATSR algorithm has been using a static weighted sum approach, which is neither practical nor flexible. To solve this problem, the present work has examined and evaluated the use of an ANN to give appropriate real-time settings of the trust weights according to the WSN's environmental changes. ANNs are suitable for real-time operation due to their parallelism, efficiency, robustness, fault tolerance, and noise tolerance.

### 5.1 ATSR Model with Embedded ANN

A common scenario in a WSN environment is that a sensor node wants to transmit data packets to a destination node, but there is a risk of packets dropping due to compromised malicious nodes. By including an ANN in real-time trust computations, a routing node can avoid the attacker and pass the packets through trusted paths to reach their destinations safely. An ANN is embedded in each sensor node in the WSN to increase the probability of avoiding

malicious nodes without dropping packets. It is trained by providing examples to help the sensor node select a trusted next-hop neighbour. The ANN knowledge is obtained by supervised learning using the backpropagation algorithm [3, 5, 18, 41, 42, 45, 47].

The training and test datasets are generated by conducting simulations using different scenarios with static ATSR trust weights. After completing the training phase, the trained ANN model is tested using the test dataset. A sigmoid activation function is used to give the ANN's output as a continuous number between 0 and 1 [49, 50], representing the predicted RCF value. To select the most trusted neighbour to send a packet to, each sensor node in the routing process monitors each of its neighbours and feeds the trust metrics of each neighbour into the trained ANN model to get the RCF output for that neighbour. Then, the sensor node chooses a neighbour with the highest RCF value as the best secure next-hop to send the packet to its destination. This process is repeated at every node in the path until the packet reaches its destination.

### 5.2 Design of the ANN

The number of hidden layers in an ANN and the number of neurons in each hidden layer should depend on the complexity of the problem. For ANN applications in WSNs, resource limitations force the designer to use a small number of trainable parameters [51]. The proposed integrated ATSR-ANN model uses three layers: an input layer with 11 neurons, a hidden layer with two neurons, and an output layer with one neuron, as shown in Figure 1. The training, testing, and evaluation of the model are carried out in three steps.

1. Many simulation scenarios were used to simulate a lot of cases for the traditional ATSR model. The results of the simulation were then used to generate training and testing datasets.
2. The generated datasets were used to train and test the proposed ANN in Figure 1.
3. A comparison between the traditional ATSR model and the ATSR-ANN model is made to assess the improved performance.

### 5.3 Generating the Dataset

The J-Sim simulator was used to run a traditional ATSR model and generate two datasets to train and test an ANN. Many different scenarios were used

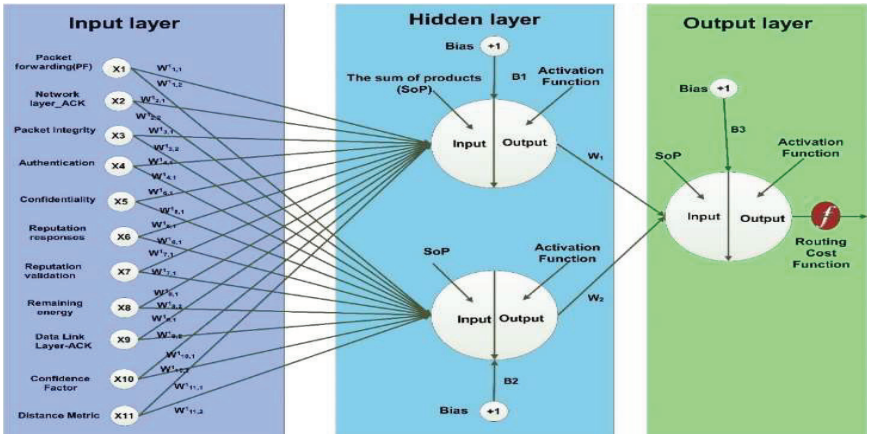


**Figure 1** The ANN architecture.

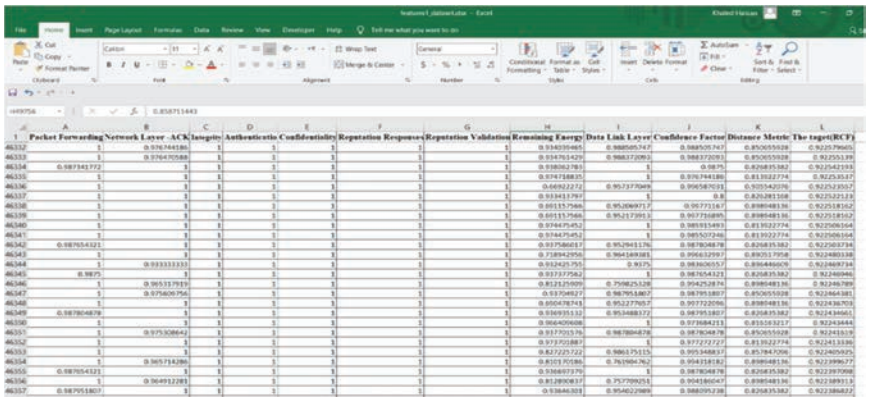

**Figure 2** Spreadsheet snapshot for a part of the generated dataset.

to simulate a WSN network with different types of malicious nodes. In each scenario, the trust static weights were adjusted to fit the attack being considered. Every simulation experiment produced a dataset made up of 11 input metrics and the calculated RCF. These samples were saved in a CSV during the simulation runtime. This created a large dataset that mimics almost every possible case of bad behaviour. The datasets are put together in a 55,000-row CSV file. Each row consists of 12 columns, 11 metrics, and the computed RCF output. The metrics are used as inputs to the ANN, and the RCF value is the target for the supervised learning process. Figure 2 shows a spreadsheet snapshot of a part of the obtained CSV file contents.

### 5.4 The ANN Training Process

A copy of the trained ANN would be embedded in every sensor node to guide the developed ATSR-ANN model in automatically computing RCF values for its neighbours, allowing it to predict trusted and malicious nodes without human intervention.

The backpropagation algorithm is used to train an ANN in a supervised mode by updating the network weights many times iteratively to get an accurate prediction of the target RCF. Each iteration involves two phases: forward calculation and error backpropagation [13, 43, 52]. Algorithm 1 shows the training steps. The learning rate ($\lambda$) is a critical hyperparameter in the training process that controls the speed of learning. There is no optimum

**Algorithm 1** Training the ANN

**Input:**

1. The generated dataset that simulates almost all possible scenarios of malicious behavior. It consists of 50000 training patterns stored as rows in a CSV file. Each row contains 11 input metrics and one output value of the target RCF.
2. The ANN structure with 11 neurons in the input layer, two neurons in the hidden layer, and one neuron in the output layer.

**Output:** The weights and biases of the trained ANN model.

**Begin**

Set the initial weights and biases of the ANN with small random numbers;
Epochs = 5000;
Learning rate($\lambda$) = 0.0001;
iterations = 0;
**while** (iterations < Epochs) **do**
SumOfErrors = 0;
**for** every row p in the training pattern **do**
Apply the 11 metrics of row p to the input layer;
Apply the forward propagation computation stage to calculate the predicted RCF output;
Compute the error cost function: Errp = 0.5 * (the desired target output – predicted output)$^2$;
Apply the Backpropagation stage to minimize the error calculated and update the weights;
SumOfErrors + = Errp;
**end for**;
iterations + = 1;
**end while**;
Save the weights and biases of the trained ANN model;

**End**

value of $\lambda$, but a suitable value can be found by trying simulation experiments using different values. Large values of $\lambda$ increase the rate of weight updating, leading to faster results. This allows the model to learn faster, but at the cost of arriving at a suboptimal final set of ANN weights. while small values slow it down and avoid sudden changes. So, a small value of $\lambda$ may allow the model to learn a more optimal or even globally optimal set of weights, but it may take significantly longer to train [49, 50]. Several simulation experiments were done and a value of $\lambda = 0.0001$ is found to provide the largest ANN accuracy without long training time.

### 5.5 The ANN Testing Process

The trained ANN is tested alone before embedding in the traditional ATSR. The testing dataset consists of 5000 new samples that were not used in the training phase. The trained ANN uses the regression method, which involves predicting a continuous number between 0 and 1, representing the predicted RCF value. The ANN accuracy is computed using the mean absolute error as shown in Figure 3, and the trained ANN achieved a percentage accuracy of 99.60%. After supervised training in the offline mode, all the ANN weights and biases are saved as a trained model, which is included in the routing software of every sensor node.

Consequently, each sensor node will be able to collect eleven-trust metrics for every neighbour node and use its trained ANN copy to predict the RCF values online. Figure 4 gives the pseudocode for selecting the most trusted next hop node, and Figure 5 depicts the real-time computation of the RCF.

```
Begin
    N = number of samples of the testing dataset;
    AccumelatedError = 0;
    for i = 1 to N do
    {
        RelaiveErrori = abs (Target_RCFi – Predicted_RCFi) / Target_RCFi;
        AccumelatedError += RelaiveErrori;
    };
    PercentageAccuracy = 100 * (1 - AccumelatedError / N);

End
```

**Figure 3** Pseudocode for computing the ANN accuracy.

```
Begin
    Ni = number of neighbor nodes of sensor node i ;
    For all neighbor nodes X = 1  to  Ni do
    {
      Collect the 11 trust metrics Mix for node X;
      Appy Mix to the input of the ANN to get the     corresponding output RCFix ;
    };
Select the neighbor node with the highest RCF value to be the most trusted next-hop;

End
```

**Figure 4** Pseudocode for selecting the most trusted next-hop node.

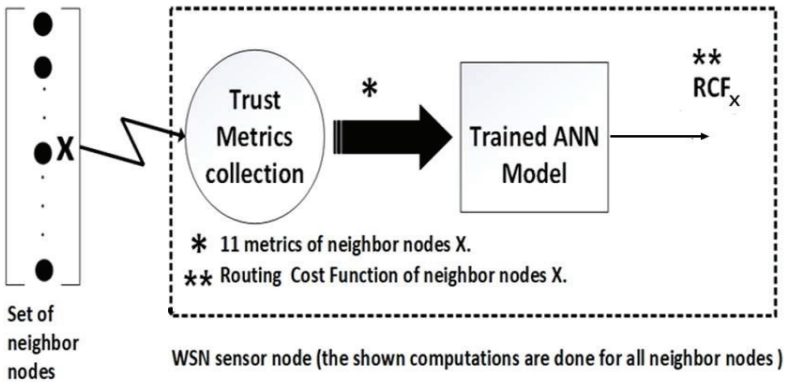


**Figure 5** Realtime computation of the routing cost function (RCF).

### 5.6 Operation of the Integrated ATSR-ANN Model

In the integrated ATSR-ANN model, each sensor node in the WSN loads a copy of the trained ANN and uses it to predict the RCF values of its one-hop neighbours. Each neighbour collects 11 metrics (as given in Table 1) that represent the online values of the specific behavioural aspects of the considered neighbour. The ANN uses only the calculations of the feedforward propagation phase to predict the neighbour's RCF, so even if offline training is slow, the trained ANN can quickly compute its RCF output. Algorithm 2 shows the steps to send a packet from a source to a destination node through a trusted path using the integrated ATSR-ANN model. To prevent infinite looping, a "hop count" field should be included in every transmitted packet and decremented at each routing node. This field should be initialized to a large value, and the packet must be dropped if the hop count falls to zero. This is not shown in Algorithm 2 for simplicity.

**Algorithm 2** The integrated model operation for transmitting through a trusted path

**Input:**

1. Source node S;
2. Destination node D;

**Output:** The trusted path;

**Begin**

CurrentNode = S;
TrustedPath = [S]; // list of nodes
**Repeat**
**for** (all neighbors of CurrentNode) **do**
Collect the 11 trust metrics of the current neighbor;
Apply the forward propagation computation stage to calculate the predicted RCF output;
**end for**;
// the best next-hop for sending packets to the destination would have the highest RCF value.
NextHop = the neighbor that has the highest RCF value;
Transmit the packet to the selected NextHop node;
CurrentNode = NextHop;
TrustedPath = TrustedPath + NextHop; // append to the list
Until NextHop = D;

**End**

## 6 Simulations and Results

A sample WSN is simulated using the J-Sim simulator [53, 54] to evaluate the performance of the integrated ATSR-ANN model. The simulated network topology consists of 100 benign sensor nodes (n0 to n99) placed at fixed locations, forming a $10 \times 10$ grid, and communicating using the IEEE 802.15.4 standard. The packet delivery ratio (PDR) is used to evaluate the model's performance. Numerous experiments for various scenarios are done by adding various types of malicious nodes to the benign WSN to conduct grey-hole and black-hole attacks. The malicious nodes were added to the network at random, thus increasing the total number of sensor nodes in the simulation. All neighbours are given an initial full trust value of 1, which means that initially all sensor nodes in the network can be trusted. During the simulation period, the role of the developed ATSR-ANN model in each sensor node is to monitor the behaviour of its neighbours and correspondingly readjust the trust values of the neighbours. Benign neighbours will continue to have full trust, whereas malicious neighbours would be detected, and their trust values would be accordingly reduced. In this way, less trusted nodes would be identified and avoided in the routing process.

### 6.1 Case Studies

In the simulated WSN, a random waypoint mobility model [53, 54] is used to move sensor nodes continuously at random except for the destination, which is stationary. Different speeds are tested for two scenarios.

(a) A benign scenario for a 100-node network without any malicious nodes.
(b) An attacking scenario with added 15 malicious nodes that are randomly.

placed in the benign network, being 10 gray-hole, and 5 black-hole attackers. The packet delivery ratio (PDR) is considered the essential performance measure and is computed as follows:

$$\text{PDR} = (\text{total number of packets received})/ \\ (\text{total number of packets sent}) * 100.$$

Figure 6 shows the PDR measure for both scenarios at different nodes' speeds. It is noted that PDR decreases in both scenarios when the speed of sensor nodes increases. This happens because a packet may be lost when the selected next-hop node moves outside the transmitting range of the sending node before receiving the packet. Additionally, malicious nodes further reduce the PDR ratio.

Figure 7 shows the remaining battery energy for the network at different sensor nodes' speeds for both the benign and the attacking scenarios. It shows

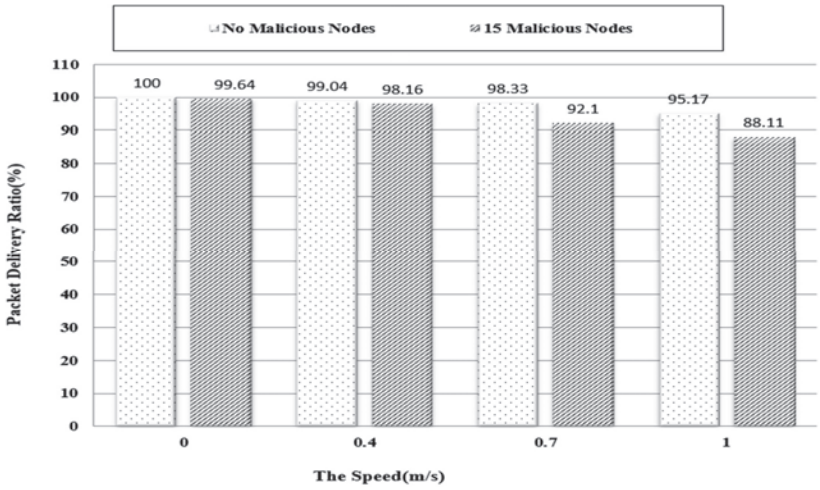


**Figure 6** Packet delivery ratio for both scenarios when sensor nodes move at different speeds.

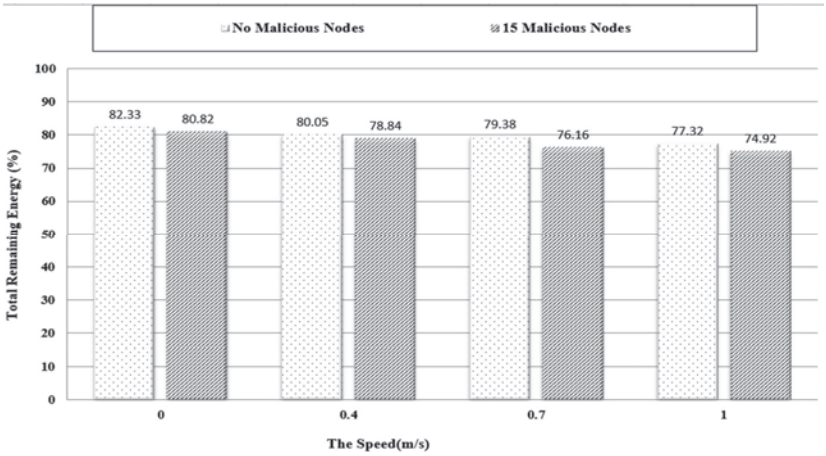


**Figure 7** The total remaining energy for both scenarios when sensor nodes move at different speeds.

that the total energy consumption increases with speed. Whereas, in the attacked WSN, sensor nodes use trusted paths that are typically longer than the shortest paths, leading to increased energy consumption. On the other hand, in the benign WSN, the energy consumption is lower because the sensor nodes use the shortest routes with a smaller number of hops.

## 6.2 Assessing the ATSR-ANN Model

The traditional ATSR model and the ATSR-ANN model were simulated and compared to get a quantitative measure of performance improvement. In all simulation experiments, the ATSR-ANN model showed a great improvement in performance, especially when sensor nodes were moving.

Figure 8 compares the developed ATSR-ANN model with the traditional ATSR model with respect to the PDR measure for the benign WSN scenario. Similarly, Figure 9 compares the same performance measure for both models in the attacking scenario. Tables 2(a) and 2(b) compare the remaining energy as a percentage of initial battery energy for both benign and attacking scenarios, with a slight improvement in the ATSR-ANN model.

The obtained PDR depends on the presence of attackers and the speed sensor nodes. When the speed of sensors increases, the PDR decreases. Because when a routing node transmits a packet to a selected trusted next hop, there is a chance that the next hop will move outside the transmitting range before completing the transmission, increasing the average packet loss. Upon

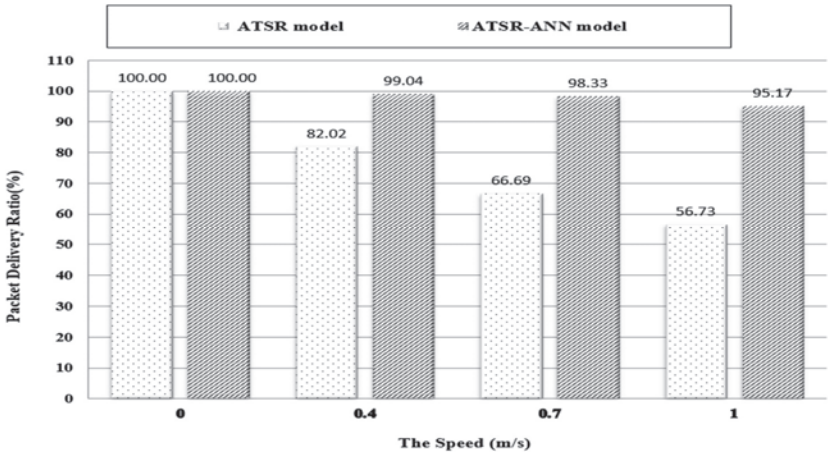


**Figure 8** Comparison between both models in terms of packet delivery ratio in the benign WSN.

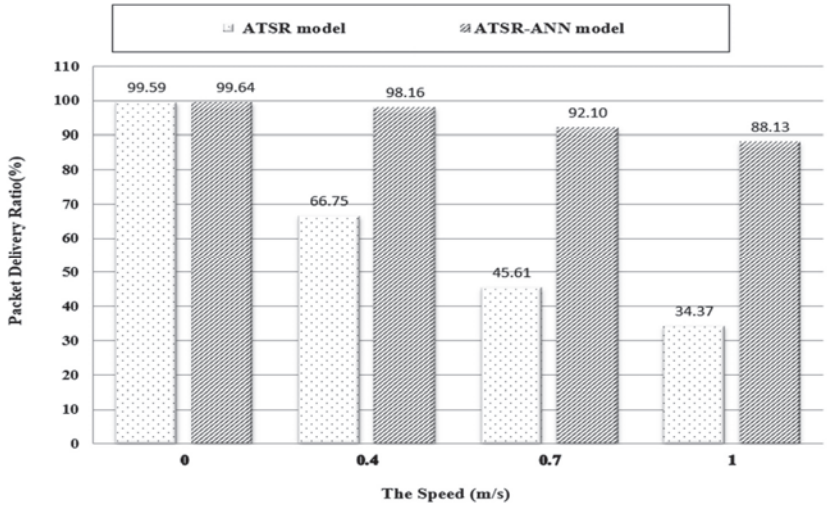


**Figure 9** Comparison between both models in terms of packet delivery ratio in the attacked WSN.

analysing the results shown in the benign scenario of Figure 8, especially when sensor nodes are stationary (Speed = 0), there is no packet loss in both models (PDR = 100%). This is due to the fact that all sensor nodes are considered trusted and only the distance criterion is applied. This leads to selecting the shortest route with the lowest number of hops. It should be

**Table 2(a)** Total remaining energy in the benign WSN

| Speed (m/s) | 0 | 0.4 | 0.7 | 1 |
|---|---|---|---|---|
| Traditional ATSR Model | 81.58 | 78.15 | 77.61 | 75.85 |
| ATSR-ANN Model | 82.33 | 80.05 | 79.38 | 77.32 |

**Table 2(b)** Total remaining energy in the attacked WSN

| Speed (m/s) | 0 | 0.4 | 0.7 | 1 |
|---|---|---|---|---|
| Traditional ATSR Model | 79.53 | 76.74 | 73.18 | 71.60 |
| ATSR-ANN Model | 80.82 | 78.84 | 76.16 | 74.92 |

noted that, in cases of higher moving node speeds, packet loss occurs when selected next-hop nodes move outside the transmitting range before packet transmission is completed.

On the other hand, in the attacking scenario of Figure 9, there is an obvious drop in PDR due to the attacks. However, there are two interesting cases:

- When sensor nodes move at a speed of 1 m/s, the PDR value for the traditional ATSR model is found to be 34.4%. On the other hand, for the ATSR-ANN model under the same conditions, PDR is 88.1%. Such a great improvement in the packet delivery ratio is a remarkable result of using secure routing paths.
- when sensor nodes are stationary (speed $= 0$) and there are malicious nodes in the network, resulting in slight packet loss in both models. The reason is that Initially, every sensor node assumes all neighbours are trusted, and they are given a full trust value of 1. Then, after a few interactions, a sensor node can differentiate between benign and malicious neighbours at the cost of losing some packets. This is because a malicious neighbour will drop some of the packets they receive before the sending sensor node can identify the malicious nature of this neighbour and avoid routing further packets to it.

ATSR-ANN mode is an artificial neural network technique that can generalize training data to deal with different situations. Actually, the decision-making logic in the ATSR-ANN model is based on data rather than on predefined static rules. Sensor nodes continuously monitor the trust and reputation metrics of their neighbour nodes, allowing them to adapt to new situations easily and avoid routing to less trusted nodes. This explains why the ATSR-ANN model showed a significant performance improvement when the WSN was under malicious attack and its environment was continuously

changing. Such changes include the changeable behaviour of the malicious sensor nodes, their interaction patterns, and location changes. So, the traditional ATSR model cannot keep up with dynamic changes in the network environment, resulting in a reduced packet delivery ratio.

## 7 Conclusion

The present work aims to enhance the routing security in wireless sensor networks (WSNs) against internal attacks, which are mostly launched from compromised nodes. Such nodes would inject malicious behaviour into apparently authentic sensor nodes to disrupt the network's operation and/or reduce its performance.

To this end, an adaptive ambient trust sensor routing (ATSR-ANN) model has been developed by integrating the traditional ATSR model with an artificial neural network (ANN) to provide a real-time capability for avoiding routing through malicious nodes. The problem with the traditional ATSR model is that it uses manually adjusted static parameters to identify malicious nodes. This means that it cannot adapt to changes in the WSN environment, especially the misbehaviour of malicious nodes. Alternatively, the developed adaptive integrated ATSR-ANN model has a context-aware characteristic thanks to the integrated ANN. The integrated model builds trust and reputation values based on a well-trained ANN rather than on predefined static rules. Sensor nodes are continuously monitoring the trust and reputation metrics of their neighbour nodes, and hence they can adapt to new situations easily and avoid routing to less trusted nodes. A simple ANN having a single hidden layer with two neurons was used. The input layer contains 11 neurons that accept the monitored trust and reputation metrics, and the output layer contains a single neuron that provides a predicted trust value in the range [0 to 1]. Backpropagation was applied to train and test the ANN using a dataset generated by simulations. The J-Sim simulator was used to simulate the traditional ATSR model and generate a sufficiently large dataset. Many different scenarios were done to simulate a sample WSN network with different types of malicious nodes. In each simulation scenario, the trust weights were carefully adjusted to suit the considered attack. With a very large number of different scenarios, we managed to build a large dataset that mimics almost all possible scenarios of malicious behaviour. Simulation experiments have been done to compare the performance of the integrated model with that of the traditional ATSR model. The obtained results showed an increased packet delivery ratio and reduced node power consumption

compared with the traditional ATSR model. This indicates an overall increase in the number of packets successfully delivered to their destinations. As the results showed, when sensor nodes were static and malicious nodes were present, the average accuracy was 99.6%, while when they were in motion, it was 88.1%. The main point in the developed ATSR-ANN model is that the trained ANN allows each sensor node that participates in the routing to find secure, trusted paths to send packets from any source sensor node to any destination node while avoiding compromised malicious nodes.

## References

[1] C. D. McDermott and A. Petrovski, 'Investigation of computational intelligence techniques for intrusion detection in wireless sensor networks', international journal of computer networks and communications, vol. 9, no. 4, 2017.

[2] Z. Ye, T. Wen, Z. Liu, X. Song, and C. Fu, 'An efficient dynamic trust evaluation model for wireless sensor networks', In Sensors, vol. 2017, 2017.

[3] H. Sharma, A. Haque, and F. Blaabjerg, 'Machine learning in wireless sensor networks for smart cities: a survey', Electronics (Basel), vol. 10, no. 9, p. 1012, 2021.

[4] W. Fang, W. Zhang, W. Chen, T. Pan, Y. Ni, and Y. Yang, 'Trust-based attack and defense in wireless sensor networks: a survey', Wireless Communications and Mobile Computing, vol. 2020, 2020.

[5] C. R. Morales, F. Rangel de Sousa, V. Brusamarello, and N. C. Fernandes, 'Evaluation of Deep Learning Methods in a Dual Prediction Scheme to Reduce Transmission Data in a WSN', Sensors, vol. 21, no. 21, p. 7375, 2021.

[6] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, ' Application of machine learning in wireless networks: Key techniques and open issues', Proc. In IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp., 2019.

[7] YG. Bhatti, ' Machine learning based localization in large-scale wireless sensor networks', Sensors, vol. 18, no. 12, p. 4179, 2018.

[8] F. Sanhaji, H. Satori, and K. Satori, 'Clustering Based on Neural Networks in Wireless Sensor Networks', in: Proceedings of the 2nd International Conference on Computing and Wireless Communication Systems, pp. 1–6, 2017.

[9] P. Hankare, S. Babar, P. Mahalle, 'Trust management approach for detection of malicious devices in siot', ISSN 1846-6168 (Print), ISSN 1848-5588 (Online) Original scientific paper https://doi.org/10.31803/tg-20210204180217, vol. 15, no. 1, pp. 43–50, 2021.

[10] Vasily A. Desnitsky, Igor V. Kotenko, Igor B. Parashchuk,'Neural Network Based Classification of Attacks on Wireless Sensor Networks', IEEE, 2020.

[11] A. Beheshtiasl and A. Ghaffari, 'Secure and trust-aware routing scheme in wireless sensor networks', in Wireless Personal Communications, vol. 107, no. 4, pp. 1799–1814, 2019.

[12] J. Wang, X. Jing, Z. Yan, Y. Fu, W. Pedrycz, and L. T. Yang, 'A survey on trust evaluation based on machine learning', ACM Computing Surveys (CSUR), vol. 53, no. 5, pp. 1–36, 2020.

[13] H. Rathore, 'Case study: A review of security challenges, attacks and trust and reputation models in wireless sensor networks', in Mapping Biological Systems to Network Systems, pp. 117–175, 2016.

[14] F. Ishmanov and Y. bin Zikria, 'Trust mechanisms to secure routing in wireless sensor networks: Current state of the research and open research issues', in Journal of Sensors, vol. 2017.

[15] H. Deng, X. Sun, B. Wang, and Y. Cao, 'Selective forwarding attack detection using watermark in WSNs', ISECS International Colloquium on Computing, Communication, Control, and Management, vol. 3, pp. 109–113, 2009.

[16] Y. Cho and G. Qu, 'Detection and prevention of selective forwarding-based denial-of-service attacks in WSNs', in: International Journal of Distributed Sensor Networks, vol. 9, no. 8, p. 205920, 2013.

[17] H. Alzaid, M. Alfaraj, S. Ries, A. Jøsang, M. Albabtain, and A. Abuhaimed, 'Reputation-based trust systems for wireless sensor networks: A comprehensive review', in IFIP International Conference on Trust Management, pp. 66–82, 2013.

[18] C.B. Vinutha, N. Nalini and B.S. Veeresh, 'Energy Efficient Wireless Sensor Network Using Neural Network Based Smart Sampling and Reliable Routing Protocol', IEEE WiSPNET conference, 2017.

[19] A. Akbas, H. U. Yildiz, A. M. Ozbayoglu, and B. Tavli, 'Neural network based instant parameter prediction for wireless sensor network optimization models', in Wireless Networks, vol. 25, no. 6, pp. 3405–3418, 2019.

[20] B. Jaint, V. Singh, L. K. Tanwar, S. Indu, and N. Pandey, 'An efficient weighted trust method for malicious node detection in clustered

wireless sensor networks', in 2018 2nd IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), pp. 1183–1187, 2018.

[21] Bassam Hasan, Sameer Alani, Mohammed Ayad Saad, 'Secured node detection technique based on artificial neural network for wireless sensor network', International Journal of Electrical and Computer Engineering (IJECE), February 2021.

[22] F. Ishmanov, S. W. Kim, and S. Y. Nam, 'A robust trust establishment scheme for wireless sensor networks', in Sensors, vol. 15, no. 3, pp. 7040–7061, 2015.

[23] F. Ghasemi, 'Secure Geographic Routing in Wireless Sensor Networks', Master of Science University of Gothenburg, Gothenberg, Sweden, 2013.

[24] B. Karp and H.-T. Kung, 'GPSR: Greedy perimeter stateless routing for wireless networks', in Proceedings of the 6th annual international conference on Mobile computing and networking, pp. 243–254, 2000.

[25] T. Zahariadis, H. C. Leligou, S. Voliotis, S. Maniatis, P. Trakadas, and P. Karkazis, 'Energy-aware secure routing for large wireless sensor networks', WSEAS Transactions on Communications, vol. 8, no. 9, pp. 981–991, 2009.

[26] T. Zahariadis et al, 'Design and implementation of a trust-aware routing protocol for large WSNs', in: International Journal of Network Security & Its Applications (IJNSA), vol. 2, no. 3, pp. 52–68, 2010.

[27] H.-C. Leligou, P. Trakadas, S. Maniatis, P. Karkazis, and T. Zahariadis, 'Combining trust with location information for routing in wireless sensor networks', Wireless Communications and Mobile Computing, vol. 12, no. 12, pp. 1091–1103, 2012.

[28] T. Zahariadis, P. Trakadas, H. C. Leligou, S. Maniatis, and P. Karkazis, 'A novel trust-aware geographical routing scheme for wireless sensor networks', Wirel Pers Commun, vol. 69, no. 2, pp. 805–826, 2013.

[29] Theodore Zahariadis, Helen C. Leligou, Panagiotis Trakadas, Stamatis Voliotis, 'Mobile Networks Trust management in wireless sensor networks', Wiley InterScience, Published online 8 April 2010.

[30] T. Zahariadis, P. Trakadas, S. Maniatis, P. Karkazis, H. C. Leligou, and S. Voliotis, 'Efficient detection of routing attacks in wireless sensor networks', in 2009 16th International Conference on Systems, Signals and Image Processing, pp. 1–4, 2009.

[31] S. Voliotis, T. Zahariadis, H. C. Leligou, D. Bargiotas, P. Trakadas, and P. Karkazis, 'A Scalable Geographical Routing approach for Wireless Sensor Networks', IWSSIP, 2010.
[32] P. Trakadas, S. Maniatis, P. Karkazis, T. Zahariadis, H.-C. Leligou, and S. Voliotis, 'A novel flexible trust management system for heterogeneous wireless sensor networks', in 2009 International Symposium on Autonomous Decentralized Systems, pp.1–6, 2009.
[33] Y. Stelios, N. Papayanoulas, P. Trakadas, S. Maniatis, H. C. Leligou, and T. Zahariadis, 'A distributed energy-aware trust management system for secure routing in wireless sensor networks', International Conference on Mobile Lightweight Wireless Systems, pp. 85–92, 2009.
[34] T. Zahariadis, H. Leligou, P. Karkazis, and P. Trakadas, 'Energy efficiency and implementation cost of trust-aware routing solutions in WSNs', in2010 14th Panhellenic Conference on Informatics, pp. 194–198, 2010.
[35] M. García-Otero et al., 'Secure geographic routing in ad hoc and wireless sensor networks', inEURASIP Journal on Wireless Communications and Networking, vol. pp. 1–12, 2010.
[36] H. C. Leligou et al., 'The impact of indirect trust information exchange on network performance and energy consumption in wireless sensor networks', in Proceedings ELMAR-2011, pp. 153–156, 2011.
[37] A. Yasin and K. Sabaneh, 'Enhancing Wireless Sensor Network Security using Artificial Neural Network based Trust Model', Int. J. Adv. Comput. Sci. Appl, vol. 7, no. 9, pp. 222–228, 2016.
[38] Fuad A. Ghaleb, Anazida, Zainal, Murad, A. Rassam, and Fathey Mohammed, 'An Effective Misbehavior Detection Model using Artificial Neural Network for Vehicular Ad hoc Network Applications', IEEE Conference on Application, Information and Network Security (AINS), 2017.
[39] Farah Sanhaji, Hassan Satori, Khalis Satori, 'Cluster Head Selection based on Neural Networks in Wireless Sensor Networks' 978-1-5386-7850-3/19/$31.00, IEEE, 2019.
[40] Y. Trofimova, A. M. Moucha, and P. Tvrdik, 'Application of neural networks for decision making and evaluation of trust in ad-hoc networks', in 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), 'pp. 371–377, 2017.
[41] R. Azmi, M. Hakimi, and Z. Bahmani, 'Dynamic reputation based trust management using neural network approach', in International Journal of Computer Science Issues (IJCSI), vol. 8, no. 5, p. 161, 2011.

[42] J. Krenek, K. Kuca, O. Krejcar, P. Maresova, V. Sobeslav, and P. Blazek, 'Artificial neural network tools for computerised data modeling and processing', in 2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI), pp. 255–260, 2014.

[43] G. Mahalakshmi, E. Uma, M. Vinitha, and M. Aroosiya, "VANET: Trust Evaluation Using Artificial Neural Network", in Advances in Parallel Computing Technologies and Applications, IOS Press, pp. 9–17, 2021.

[44] D. P. Kumar, T. Amgoth, and C. S. R. Annavarapu, 'Machine learning algorithms for wireless sensor networks: A survey', Information Fusion, vol. 49, pp.1–25, 2019.

[45] H. Kaur and S. Sahore, 'A survey on wireless sensor network (wsn) security using AI methods', Int. J. Latest Trends Eng. Technol, vol. 7, no. 4, pp. 234–239, 2016.

[46] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, 'Machine learning in wireless sensor networks: Algorithms, strategies, and applications', in IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 1996–2018, 2015.

[47] G. M. Jinarajadasa and S. R. Liyange, 'A survey on applying machine learning to enhance trust in mobile adhoc networks', in 2020 International Research Conference on Smart Computing and Systems Engineering (SCSE), pp. 195–201, 2020.

[48] Yi Zhao, 'Combination of Wireless sensor network and artificial neural network: A new approach of Modeling', in 2013.

[49] A. F. Gad, A. F. Gad, and S. John, 'Practical computer vision applications using deep learning with CNNs', Springer, 2018.

[50] A. F. Gad and F. E. Jarmouni, 'Introduction to Deep Learning and Neural Networks with PythonTM: A Practical Guide', Academic Press, 2020.

[51] R. v Kulkarni and G. K. Venayagamoorthy, 'Neural network based secure media access control protocol for wireless sensor networks', in 2009 international joint conference on neural networks, pp. 1680–1687, 2009.

[52] B. Rajasekaran and C. Arun, 'Detection of malicious nodes in wireless sensor networks based on features using neural network computing approach', International Journal of Recent Technology and Engineering, vol. 7, no.4, pp. 188–192, 2018.

[53] A. Sobeih et al., 'J-sim: A simulation environment for wireless sensor networks', in 38th Annual Simulation Symposium, pp. 175–187, 2005.

[54] Y. G. G. H. C. H. H. K. L. K. N. L. H. L. A. S. H. T. H. Z. R. Zheng. Wei-peng Chen, 'https://sites.google.com/site/jsimofficial/', 2013.

## Biographies

**Khaled Mohammed Ali Hassan** has received a bachelor's degree in Computer Engineering from the Electronics and Electricity Faculty of Engineering at Aleppo University in Syria. He received his master's degree in Computer Engineering from the Faculty of Engineering at Cairo University in 2015. He is currently working as a researcher on his Ph.D. in the Systems and Computers Engineering Department of the Faculty of Engineering at Al-Azhar University in Cairo, Egypt.

**Mohamed Ashraf Madkour** has got his B. Sc. and M. Sc., degrees in Electrical Engineering in 1968 and 1974, respectively, and got his Ph. D. degree in computer networking from the Electrical Engineering Department, Ain Shams University in 1981. Dr. Madkour is a professor emeritus in the Systems and Computers Engineering Department, Faculty of Engineering, Al-Azhar University, Cairo, Egypt, where he teaches courses and does research on computer networking, internetworking, and systems and data security. Dr. Madkour has published more than 40 research papers in international and local journals and conferences, and his research interests include internetworking and wireless networks, cyber security, artificial intelligence, and data science.

**Sayed Abd El Hady Nouh** is a computer network professor in the Department of Computers and Systems Engineering at Al-Azhar University in Cairo, Egypt. He received his B.Sc. degree in Communications Engineering and his M.Sc. degree in Computer Engineering from Al-Azhar University in 1978 and 1982, respectively. He received his Ph.D. degree in Computer Engineering from AGH University, Cracov, Poland, in 1992. From 2006–2010, he served as the Egyptian Consultant at the African Union, in Addis Ababa, Ethiopia. From 2012 to 2015, he served as the chairman of the Computers and Systems Engineering Department at Al-Azhar University. He is the chairman of the committee for upgrading professors and associate professors. He has been an IEEE member since 1991. He has been involved with research in performance analysis and evaluation of computer networks, Ad-hoc routing protocols, routing and security protocols for wireless sensor networks, mobile computing and wireless networking, modeling and computer simulation techniques, and data communications networks.