Received: 6 February 2021 | Revised: 12 July 2021 | Accepted: 23 August 2021

DOI: 10.4218/etrij.2021-0044

ORIGINAL ARTICLE

ETRI Journal WILEY

A new method to detect attacks on the Internet of Things (IoT) using adaptive learning based on cellular learning automata

Javad Dogani[1] | **Mahdieh Farahmand**[2] | **Hassan Daryanavard**[1]

[1]Department of Electrical and Computer Engineering, University of Hormozgan, Bandar Abbas, Hormozgan, Iran

[2]Department of Mechanics, Electrical and Computer, Islamic Azad University Science and Research Branch, Tehran, Iran

Correspondence
Javad Dogani, Department of Electrical and Computer Engineering, University of Hormozgan, Bandar Abbas, Hormozgan, Iran.
Email: j.dogani@shirazu.ac.ir

Abstract

The Internet of Things (IoT) is a new paradigm that connects physical and virtual objects from various domains such as home automation, industrial processes, human health, and monitoring. IoT sensors receive information from their environment and forward it to their neighboring nodes. However, the large amounts of exchanged data are vulnerable to attacks that reduce the network performance. Most of the previous security methods for IoT have neglected the energy consumption of IoT, thereby affecting the performance and reducing the network lifetime. This paper presents a new multistep routing protocol based on cellular learning automata. The network lifetime is improved by a performance-based adaptive reward and fine parameters. Nodes can vote on the reliability of their neighbors, achieving network reliability and a reasonable level of security. Overall, the proposed method balances the security and reliability with the energy consumption of the network.

KEYWORDS

adaptive learning, cellular learning automata (CLA), energy consumption, Internet of Things (IoT), network lifetime, reliability, security

1 | INTRODUCTION

The Internet of Things (IoT) is a new concept in technology and communication [1]. IoT has emerged as a groundbreaking platform on which any human, animal, or object can send information through communication channels such as the internet or intranet [2,3]. IoT connects all objects and devices to a virtual world [4] with all-digital tools. This paradigm is both advantageous and disadvantageous [5,6]. "Internet of Things," a term coined by Kevin Ashton in 1999 [7], can generate, analyze, and make decisions about connected objects; accordingly, it has received much attention in recent years and is projected to connect up to 100 billion devices by 2025 [8].

IoT integrates the physical and internet objects from various fields such as home automation, industrial processes, human health, and environmental monitoring [9]. It also connects the devices used in daily living to the internet. Protecting the vast numbers of IoT devices from threats is among the biggest challenges of technology-based companies. IoT sensors receive information from their environment and forward it to their neighboring nodes. Attacks such as Denial-of-Services (DoS), closed-loop attacks, and black hole and worm attacks disrupt the routing process and reduce the network performance

[9]. Therefore, secure and reliable IoT routing is fundamental to the smooth running of IoT. Intrusion detection systems (IDSs) that usually form part of other security systems or software have now been applied to information systems. IDS security and authorized access control mechanisms provide a double line of defense against intrusion [10,11]. An IDS aims to automate the protection of information systems [12]. Although IDS is widely used in network security and has been extensively developed in recent years, intrusion detection technology remains incomplete and far from ideal [13].

For peer-to-peer communications between machines, sensors, and hardware, IoT relies on IP-based networks to send the data collected from connected devices to gateways, cloud platforms, or middleware. IoT is a growing part of future 5G networks providing free access and new services to users and businesses [14]. However, as the resources of front-end IoT devices are limited, many security mechanisms cannot protect IoT networks. When encryption fails, an IDS efficiently prevents intrusion and enforces the security of IoT networks [15]. IDSs have played a pivotal role in protecting networks and information systems for more than two decades [16]. Nevertheless, the particular characteristics of IoT, such as constrained-resource devices, specific protocol stacks, and standards [17], are not always well handled by traditional IDS techniques.

The network nodes in most applications face security challenges. Such challenges must be met with innovative methods that establish a secure information-exchange environment. Advanced learning techniques are anticipated to tackle the existing and anticipated security and privacy issues of IoT [18]. Each IoT sensor collects information from the environment and sends its collected data to the central node. Along with security, quality of service is a fundamental requirement of IoT. The quality of IoT services is evaluated by various parameters, such as the transmission range of nodes, the optimal number of active nodes, the network lifetime, and energy consumption of the network [12]. IoT sensors are cheap, inexpensive, and low power consumers. However, many IoT applications must run on batteries in an alienated environment for many years, and the overall energy consumption is reduced to improve the performance. Therefore, a power-efficient solution with high efficiency and extended lifetime is highly demanded [19].

In this paper, the quality of IoT service is improved by an intelligent method that deploys cellular learning automata (CLA) [20]. This new multistep routing protocol based on CLA is intended to improve the security and reliability of IoT. Most of the above methods provide protection while ignoring the energy consumption. Incorporating the energy consumption would maintain the performance and extend the lifetime of the network. This paper aims to resolve these limitations while achieving an appropriate security level. The new protocol is both energy conscious and energy efficient and establishes a proper balance between the network reliability and energy consumption criteria. To achieve this balance, the CLA finds the most suitable radio board for each node while considering the network's energy consumption.

By creating a CLA for each node in the network, applying the load distribution technique, and optimizing the use of all nodes, we try to balance the energy consumptions of the nodes and ultimately increase the network lifetime. First, the statistical parameters are calculated for each node. Communications within the network are then established based on the neighbors' reliability levels, which are calculated at each node. Our proposed protocol includes three phases: (1) topology management, (2) route identification, and (3) traffic distribution and route maintenance. The first phase selects the best transmission range for all nodes. The second phase creates a routing table for each node and identifies all individual paths between each source–destination node pair. Our intrusion detection method uses learning automata with adaptive reward or penalty parameters. Whether a route is rewarded or penalized depends on the comprehensive communication quality of the nodes. The third phase prevents premature death by distributing the traffic to all nodes in the network. If there is no active route between the source and destination, multiple paths are re-identified after the failure of the initial routes. In the CLA with adaptive parameters, the nodes vote on the reliability of their neighbors. The comprehensive communication quality reflects the forwarding behaviors of nodes. The main contributions of this article are summarized below.

- We improve the security and reliability of IoT networks by a new multistep routing protocol based on CLA. The protocol establishes a proper balance between the two essential criteria of IoT: network reliability and energy consumption.
- We create a CLA for every node in the network and employ the adaptive reward and penalty technique to balance the energy consumption of the nodes and ultimately increase the network's lifetime.
- We calculate the reliability levels of nodes based on the communication quality. The nodes then vote on their neighbors' reliability. Based on these reliability levels, we establish a communication in the network.
- We simulate the network simulation and collect information on the network status. The effectiveness of the proposed method is compared with those of two well-established previous studies.

2 | RELATED WORKS

Pajouh et al. [21] studied intrusion detection in an IoT case study of wireless sensor networks. IDSs can detect any unauthorized use of the system causing abuse or harm by internal and external users. For this purpose, they employed a k-nearest neighbor classification algorithm. Nearest neighbor optimization finds and categorizes the nearest points in metric spaces.

Rani et al. [22] introduced a method that assesses trust in the IoT environment and access control. The trust was evaluated using game theory and related techniques. A trust model for IoT was built by mapping the trust assessment and controlling access to game concepts and players. For accurate measurements of trust level, the authors defined a profit function based on performance goals such as energy consumption and data transfer rate. After presenting a system model that examines trust in various high trust, moderate trust, and low trust cases, the authors concluded that e-commerce in the IoT network can help to increase trust.

Nguyen et al. [23] presented DIoT, an autonomous self-learning distributed system that detects compromised IoT devices. DIoT builds effectively on device-type-specific communication profiles without any human intervention or labeled data. Subsequently, the system detects abnormal deviations in the devices' communication behavior, which may be caused by malicious adversaries. Furthermore, DIoT utilizes a federated learning approach for efficient aggregation of behavior profiles. This system was the first to employ a federated learning approach to anomaly-detection-based intrusion detection. Consequently, DIoT can cope with upcoming and unknown attacks.

A selective forwarding attack detection method was proposed in [24]. This method uses adaptive learning automata and communication quality to distinguish malicious packet dropping from normal packet loss. The method eliminates the impact of normal packet loss on selective forwarding attack detection. It also detects both ordinary and special cases of selective forwarding attacks. The nodes' current comprehensive communication qualities reflect their short- and long-term forwarding behaviors. The method accounts for normal packet loss caused by unstable channels and medium-access-control layer collisions. The adaptive reward and penalty parameters of the detection learning automata are determined by the comprehensive communication quality of the node and the nodes' votes on the reliability of their neighbors. Normal nodes are rewarded and malicious ones are punished.

Data routing in the IoT network is a particularly worrying security problem. Large-scale data exchange between devices is an easy target for attackers. Under such an attack, the data paths are compromised. Airehrour et al. [25] provided a secure routing communication framework called SecTrust, a framework on which trust between the nodes in IoT can be calculated, assessed, and built. The SecTrust framework creates a direct connection between connected nodes. Each node calculates the reliability of its immediate neighbors based on the direct and recommended trust values. The neighbors with high trust values are selected for safe routing, whereas the low trust nodes are considered either as malicious nodes or selfish nodes that conserve their resources (such as battery power). SecTrust implements five main processes: trust calculation, trust monitoring, identification and separation of malicious nodes, trust rating, and backup/recovery.

SecTrust tool is a promising framework for practical IoT systems, as it detects and isolates malicious actors, manages and maintains the trust and advisory systems in IoT networks, and secures IoT routing using a trust-based mechanism. The SecTrust system has shown more promising performance results than other trust-based systems.

The method presented in [26] analyzes the intrusion detection requirements of IoT networks. A uniform intrusion detection method based on an automated model was developed for large heterogeneous IoT networks. This method automatically detects and reports potential IoT attacks in three ways. Using an extension of labeled transition systems, it provides a consistent description of IoT systems and detects intrusions by comparing the abstracted action flows. Besides designing the intrusion detection approach, Fu et al. [26] constructed Event databases and implemented the Event Analyzer as an IDS.

Among the massive amount of data generated by the surge of IoT devices, attacks or untrustworthy data are nearly impossible to detect. In [27], a new hinge-classification algorithm based on minibatch gradient descent with an adaptive learning rate and momentum (HCA-MBGDALRM) was designed to minimize the effects of security attacks. Deep networks trained with this algorithm significantly outperform traditional neural networks, decision trees, and logistic regression in terms of scale and speed. We have solved the data skew problem in the shuffle phase and implemented HCA-MBGDALRM on a parallel framework that accelerates the processing speed of massive traffic datasets.

Wu and Wang [28] developed a game-theoretical analysis framework for collaborative security detection that considers defender–attacker confrontations. First, this framework analyzes the existence and uniqueness of the Nash equilibrium in a game model with complete consensus. The Nash equilibrium is then determined by an iterative learning-based calculation method. Last but

not least is a quantitative analysis of the relationship between the Nash equilibria of the game models in complete and incomplete consensus with infinite and finite numbers of iterations.

Gu et al. [29] presented a reinforced learning-based attack detection model that adapts to new characteristics in IoT attacks by automatically learning and recognizing transformations in the attack pattern. This method first learns the crucial features of IoT traffic and detects both high-rate and low-rate IoT attacks using entropy-based metrics. Leveraging the reinforcement learning technique, it continuously adjusts the attack detection threshold based on detection feedback, thereby optimizing both the detection rate and the false alarm rate.

Adaptive hybrid IDS with a timed automata controller [30] can overcome the challenges introduced by real-time service changes. The hybrid IDS obtains additional knowledge on frequent multimedia file formats and uses this knowledge in a comprehensive analysis of packets carrying multimedia files.

Deep learning (DL) is commonly used in big data analysis and has attracted special interest as a cybersecurity technique. With their self-teaching and compression capabilities, DL architectures can discover hidden patterns in the training data that discriminate attacks from benign traffic. Diro and Chilamkurti [31] developed a new DL cybersecurity approach that detects attacks on the social IoT. The authors compared the performances of the deep model and traditional machine learning and competed distributed attack detection against the centralized detection system. Neural networks are popularly employed in network intrusion detection. Because they learn complex patterns and behaviors, they are expected to differentiate between regular traffic and network attacks. Rezvy et al. [32] detected intrusion and attacks in 5G and IoT networks using a deep auto-encoded dense neural network algorithm. We evaluated this algorithm on the benchmark dataset Aegean Wi-Fi Intrusion. Hussain et al. [33] proposed a consolidated framework based on deep convolutional neural networks trained on real network datasets. Their framework enables early detection of distributed denial-of-service (DDoS) attacks orchestrated by a botnet that controls malicious devices. Puppet devices individually perform silent calls, signaling, short-message service (SMS) spamming, or a blend of these attacks in targeted calls, internet messaging, SMS, or a combination of these services, respectively, causing coordinated DDoS attacks in a cell that can disrupt cyber-physical systems operations. In evaluations, this framework achieved a high standard and detected attacked cells with high accuracy. In these studies, attack detection in IoT was evaluated in terms of error detection, accuracy, and similar parameters.

Although these methods provide security, they do not consider the energy consumption in the IoT network. As energy consumption is a crucial factor in IoT and affects the network performance, this article proposes the reduction of power consumption while improving the network security and reliability. The proposed method attempts to balance the two essential criteria of IoT: network reliability and energy consumption.

3 | CELLULAR LEARNING AUTOMATA

In this section, we briefly review the learning automata and introduce the CLA [34]. A learning machine performs a finite number of operations. Each selected action is evaluated in a possible environment, and the evaluation result is given to the automata as a positive or negative signal. This response influences the following action selected by the automata [34]. Ultimately, the automata attempt to learn the best action among all actions. The best practice will maximize the likelihood of receiving a reward from the environment. Figure 1 shows the interaction between the learning automata and their environment, and Table 1 outlines the notations used in this paper.

The environment is represented by a triplet $E \equiv \{\alpha, \beta, C\}$ in which $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_r\}$ is the set of environmental inputs, $\beta = \{\beta_1, \beta_2, ..., \beta_r\}$ is the set of environmental outputs, and $C = \{c_1, c_2, ..., c_r\}$ is the set of penalty probabilities [30]. The environment inputs one of r selected automatic functions and outputs a response β_i to action i. If β_i is a binary response, the environment is called a Model P-type environment. In such an environment, undesirable (failed) and desired (successful) responses are specified as $\beta_i(n) = 1$ and $\beta_i(n) = 0$, respectively. In a Q-type environment $\beta_i(n)$, the model can contain a limited discrete number of values in the range [0, 1], whereas in an S-type environment, the value of $\beta_i(n)$ is a random variable in the range [0 and 1]. Meanwhile, c_i is the probability of an adverse outcome of

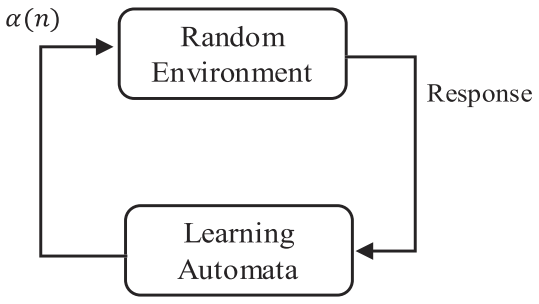

FIGURE 1 Descriptive diagram of the cellular learning automata

TABLE 1 Notations used in this paper

Notation	Definition
α	Set of CLA operations
β	Set of CLA inputs
φ	New state generation function
P	Probability vector of operation selection in CLA
$P_i(n)$	Response from the environment in step n of the CLA
T	Automata learning algorithm
a	Adaptive reward parameter
b	Adaptive penalty parameter
A_{inc}	Increase value of radio range
A_{dec}	Decrease value of radio range
R_t	Transmittance range
ELevel_j	Security level of neighboring node j
HopCount_i	Number of steps for sending information to neighboring node i
Neighbor#	Total number of neighboring nodes
ω	Weight on two levels of security and the number of step factors
$\text{Slev}_{i,j}(t)$	Security level of node i along path j at time t
AvgEnergy_i	Average energy of the first nodes of other routes from node i to the destination
MaxHop	Maximum number of links from the node receiving the ACK packet to the destination
μ_1	Minimum acceptable reward value
μ_2	Minimum acceptable of penalty value
Quality_j	Comprehensive quality of a neighboring node of *node j*
age_i	Age factor of *node i*
$\lvert W \rvert$	Window length
TR_j	Trust ratio

Abbreviation: CLA, cellular learning automata.

action α_i. In a static environment, the values of c_i remain unchanged; in a nonstatic environment, they change over time [30].

Static automata are represented by a fixed structure $\{\alpha, \beta, G, \varphi\}$ in which $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ is action set of operations, $\beta = \{\beta_1, \beta_2, ..., \beta_m\}$ is a set of inputs, and $\varphi = \{\varphi_1, \varphi_2, ..., \varphi_s\}$ is the set of internal states. The new state generation function is $F \equiv \varphi \times \beta \rightarrow \varphi$ and the output function of the automata is $G \equiv \varphi \rightarrow \alpha$. The function G writes the current state of the automata to the next output. A learning automata with a variable structure can be represented by $\{\alpha, \beta, P, T\}$, where $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ is a set of operations, $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ is a set of inputs, $P \equiv \{P_1, P_2, ..., P_r\}$ is the vector of selection probabilities for each operation, and $P(n+1) \equiv T[\alpha(n), \beta(n), P(n)]$ is the automata learning algorithm. For example, in the following linear learning algorithm, an action is presumed to be selected in step n. The optimal response from the environment is

$$\begin{aligned} P_i(n+1) &= P_i(n) + \alpha[1 - P_i(n)], \\ P_j(n+1) &= (1-\alpha)P_j(n) \quad \forall j j \neq i \end{aligned}. \tag{1}$$

An adverse response from the environment is

$$\begin{aligned} P_i(n+1) &= (1-b)P_i(n) \\ P_j(n+1) &= \frac{b}{r-1} + (1-\alpha)P_j(n) \quad \forall j j \neq i \end{aligned} \tag{2}$$

In Equations (1) and (2), a and b are the adaptive reward and penalty parameters, respectively. Depending on the values of a and b, three states of the above relations are possible. If b and a are equal, the learning automata are denoted as L_{RP}. If b is equal to the buffer, the learning automata are denoted as L_{RP}. If $b \ll a$, the learning automata are designated as L_{RIP}.

CLA is a mathematical model for systems with simple components. The behavior of each component is determined and modified according to the behaviors and past experiences of its neighbors [31]. The simple components of this model react and interact to yield complex behaviors typical of many problems. Each cell in the cellular automata of one CLA is equipped with one or more learning automata that determine the cell's state. Like cellular automata, the environment is governed by natural laws that determine whether the action chosen by the automata in the cell should be rewarded or fined. The structure of the CLA is adaptively rewarded and penalized to achieve a specific goal. The developed CLA is a random CLA that uses the learning automata to calculate the state transfer in random cellular automata. CLAs can be either asynchronous or synchronous. In a synchronous model, all cells are synchronized with a global clock and run simultaneously.

4 | PROPOSED METHOD

Using the CLA, our protocol increases the security of network nodes and hence extends the life of the network. Our proposed protocol is implemented in three phases:

1. Topology control (including two learning phases and selection of the most suitable radio board for each node)

2. Route identification
3. Traffic distribution and route maintenance

4.1 | Network definition

In the first phase, all nodes are assigned by the automata, which randomly select an automaton operation (for example, A_1, A_2, A_3). The automata update the node radio board by choosing an action and sending a help message containing the sensor identification number to the node neighbors. The number of responses to the signals indicates the number of neighbors per node or the degree of each node. In the next step of Phase 1, the selected action is awarded by an adaptive penalty or reward based on the number of responses received, and the radio board is updated. Guided by the learning automata, the nodes eventually choose the most suitable board. In the second phase, routing tables are created for each node, and all separate paths between each source–destination node pair are identified. In this way, the source node sends a confirmation message of the main route to its best neighbor toward the destination, and another confirmation message of the secondary backup route to another neighbor with lower priority than the neighbor on the main track.

This method finds N paths from sources to the destination. If the primary path is lost for any reason, the route switches to the backup path. Each node selects a path among the different paths at any time using its learning automaton. An unsafe chosen route is penalized whereas a suitable route is rewarded. The adaptive penalizing and rewarding of paths increases or decreases the probability of their selection. In the third phase, traffic is distributed among all network nodes to prevent the continued use and energy depletion of the nodes along a particular path, thereby preventing premature death of the network. Moreover, to maintain the paths in our network, several paths are rediscovered when the original path fails and when no active path exists between the source and the destination.

4.2 | Neighbors recognition and operators of the cellular automata

In the first phase, each learning automaton can undertake three operations designated A_1, A_2, and A_3 expressed by (3), (4), and (5), respectively. Operation A_1 increases the radio range of the node by a fixed value (A_{inc}), A_2 decreases the radio range of the node by a fixed value (A_{dec}), and A_3 maintains the previous radio board.

$$A_1 = (\text{New Node Radius}) = (\text{Node Radius} + A_{\text{inc}}), \quad (3)$$

$$A_2 = (\text{New Node Radius}) = (\text{Node Radius} - A_{\text{dec}}), \quad (4)$$

$$A_3 = (\text{New Node Radius}) = (\text{Node Radius} + 0). \quad (5)$$

Initially, the selection probabilities of the three operations are assumed equal. Later, the probability is calculated by (6), where m is the number of automated operations:

$$\forall i,\ i \le m\ \ P_i = \frac{1}{m}. \quad (6)$$

The automaton selects the value of the transmitted range R_t based on the network densities of all nodes and sends information to each node. At the beginning of the process, all nodes are randomly and simultaneously governed by one of the automatic operations: increasing the radio range of the node by A_{inc}, decreasing the radio range of the node by A_{dec}, or maintaining a constant radio range. The node radio board is then updated accordingly. In the following stages, the probabilities of these three actions change as penalties and rewards are issued.

The nodes propagate the HELLO packet in all broadcasts within their range. Each node that receives the HELLO packet sends a packet called ACK to the sender node. The number of neighbors is equal to the number of ACKs reaching the sender node. If the number of neighbors is less than the minimum-neighbor threshold or if the node has selected its maximum radio power and the number of neighbors exceeds the threshold, the automaton of the selected operation is fined b. Otherwise, the automaton is rewarded a for the chosen action. In both cases, the node's radio board is updated.

The destination node creates the FLOOD packet and publishes it on the network, where it is accessible to all neighboring nodes. The FLOOD package includes three fields: the sender node number, number of steps, and the sender-node energy level. The destination node fills these fields before releasing the FLOOD packet. The sender node number is set to the number (destination node), the number of steps is set to zero, and the energy level is set to the energy level (destination node). All nodes receiving the FLOOD packet return the feedback packet, and the receiving nodes of the feedback packet update their routing tables. The nodes receiving the FLOOD packet send the entire FLOOD packet through the broadcast. This process continues until the FLOOD package reaches all groups on the network.

Here, the set of CLAs at each node equals the number of identified neighbors of that node. For each neighbor node, a possible selection value is determined. These values are initially identical and are later updated

based on the penalty and reward assignments. If the number of paths to the destination is zero, the related CLA is fined by b and the neighbor search operation is repeated.

4.3 | Criteria for calculating the β signal determining the reward and penalty of each neighbor

In this paper, the node holding the information selects an action from its probability vectors (paths to destination) and sets that action as its primary path. This choice probability is initially uniform and is later determined by the probability of choosing each path. After selecting the route for information-sending, the neighbor's penalty and reward amount should be calculated to maximize the effectiveness of the following steps. In our method, the reward and penalty parameters are dynamic and adaptive and depend on the quality of the selected node. The β signal is calculated based on the quality of the chosen path as follows:

$$\text{Signal}_\beta(i) = \omega \frac{\text{SLevel}_i}{\sum_{j=1}^{\text{Neighbor\#}} \text{ELevel}_j} + (1-\omega) \frac{\text{HopCount}_i}{\sum_{j=1}^{\text{Neighbor\#}} \text{HopCount}_j} \quad \text{Neighbor\#} 1,2,3,4. \tag{7}$$

In this relation, $\text{Signal}_\beta(i)$ represents the value of the β signal calculated for the ith neighbor, ELevel_j is the security level of the neighboring node j that sends the packet, HopCount_i is the number of steps for sending information via neighboring node i to the sink node, and Neighbor# is the total number of adjacent nodes. ω is a user-determined parameter in the range [0, 1] that weights the two security levels and the number of steps. After this step, each node in the network has different possible vectors for each of its neighbors in the cellular automata model. $\text{Slev}_{i,j}(t)$ refers to the security level of *node i* along path j at time t during the path discovery process between a source node and the sink. It is calculated as

$$\text{Slev}_i = \frac{\text{DeliveryRatio}_i}{\text{SendingRatio}_i}. \tag{8}$$

Here, DeliveryRatio_i is the number of acknowledgments received by *node i* among all messages sent through *node i* and SendingRatio_i is the total number of packets sent while discovering the path between the source and destination nodes through *node i*.

4.4 | Route-selection fines and rewards along each route depending on the β signal

The node holding the information selects an action among its probability vectors. Each selection is a route through which data are sent through a nearby neighbor, and the probabilities of all neighbors sum to 1. The selection probabilities of all nodes are initially equal because the nature of the network is initially unknown, so all neighbors are given the same chance of sending information. The entries in the probability vector of neighbors are initially set to $1/\text{Neighbor}_\#$ where $\text{Neighbor}_\#$ represents the number of neighbors of a CLA. In the following steps, the probability vectors are used for node selection and their values depend on the quality of the information along the transmission paths in the previous steps, which accords with the number of adaptive fines and rewards. The CLA with adaptive parameters receives the optimal choice that improves the network efficiency.

After calculating the new probability vector, the node with the most probability is selected as the information sender. In the network routing process, the node holding the information sends a data packet to its best neighbor on the path to the destination.

After sending data, each node receives a data packet, creates an ACK packet, and sends it to the packet-sender node. After receiving an ACK, the node calculates signal_β using (7) and calculates the path quality from the energy of the selected neighbor node and the number of hops in the data path. Depending on the value of signal_β, one of the following is implemented:

- If signal_β along the receiving path of ACK exceeds the threshold specified by Inequality (9), reward the action and update the action vector's probability using the learning algorithm.

$$\text{Signal}_\beta \geq 0.5 \text{ Then} \quad a = \mu_1 + \theta_1 \frac{\gamma * \text{Elevel}_i + \text{MaxHop} - \text{HopCount}_i}{\gamma * \text{AvgEnergy}_i + \text{MaxHop}}. \tag{9}$$

- If signal_β is below the threshold, the action selected by the transmitter node is not appropriate and should be punished by Inequality (10):

$$\text{Signal}_\beta \leq 0.5 \text{ Then} \quad b = \mu_2 + \theta_2 \frac{\gamma(\text{AvgEnergy}_i - \text{Elevel}_i) + \text{HopCount}_i}{\gamma * \text{AvgEnergy}_i + \text{MaxHop}}. \tag{10}$$

In Inequalities (9) and (10), Elevel_i is the energy level of the node sending the ACK packet. AvgEnergy_i is the average energy of the first nodes of the other routes from *node i* to the destination. MaxHop is the maximum number of links from the node receiving the ACK packet to the destination. μ_1 and μ_2 denote the minimum acceptable values of the reward and penalty parameters, respectively. As the number of steps and the amount of remaining energy differ in scale, we specify a parameter γ that approximately equalizes the scales of both terms. Finally, the values of α and β are upper limited by setting $E \equiv \{\alpha, \beta, C\}$ and θ_2. From the verification received from the nodes during this step, the system can learn the best available routes for energy consumption and hop counting. After selecting and sending the data, the sending operation is complete if the package is received at the destination. Otherwise, if the original path is lost and there exist one or more backup paths, it selects one backup path for data sending.

To ensure that the data packet is received at the destination, the destination node sends an ACK packet to the sending node. As these data transmissions consume the energy of nodes, the number of messages between nodes should be reduced as far as possible. For this purpose, we introduce a parameter p. Each node selects each path and sends p data packets along it. Also, each node sends only one ACK packet to the node from which it received the data packets. This mechanism reduces the number of ACK packets exchanged in the network and consequently reduces the power consumption.

4.5 | Qualitative evaluation of comprehensive communications

The quality stability of a node, which determines the attack resistance of the node, can be determined from the quality history. The quality of the current communication is a short-term value focused on the current moment. Hence, it is not by itself a reliable indicator. In this research, the quality of a node is determined from the quality of the node communications, similarly to [24]. Using a sliding window, the proposed method interrogates each node on the quality of its communications with each of its neighbors. The nodes' opinions are pooled and whether an attack is normal or not is judged by voting. The knot is done. At the time of evaluation, a sliding time window with a dynamic age factor is applied. A sliding time window contains time units, and each window records the current quality of neighboring nodes of a CLA per unit time. The time window moves forward by one time unit, and the quality of all neighbors of the CLA recorded at the end of each time window gives the quality history of that node. The age coefficient age specifies the importance of each unit of time in the slider time window and prioritizes the recent data over the earlier data. The age of neighbor i, denoted age_i, depends on the number of time windows in which neighbor i communicates with CLA. The quality of the comprehensive connection of a node, which indicates the comprehensive quality of a neighboring node of node j, is called Quality_j. It is calculated as

$$\text{Quality}_j = \sum_{i=1}^{|W|} \text{age}_i \times \left(1 - \text{quality}_j^i\right) \times \text{quality}_j^i, \qquad (11)$$

where $\text{Quality}_j \in [0, 1]$, $|W|$ is the window length, quality_j^i denotes the quality of *node j* in the ith time window, and $\text{age}_i \in [0, 1]$ with $\text{age}_1 < \text{age}_2 < \ldots < \text{age}_{|w|}$ is the age factor. Here, the age factor is expressed by the following exponential function: $\text{age}_i = \gamma^{|W|}$ and $\gamma \in [0, 1]$.

5 | RECOGNITION AND PUNISHISMENT OF SELECTIVE ATTACK BEHAVIORS

Once the CLA has selected *node j* as the next step based on the probability of operation, it receives a response from the environment based on the quality of the comprehensive communication of node j. *Node j* is considered suspicious when the quality of the comprehensive quality_j^i connection evaluated by *node i* is less than a predefined Quality_Threshold, which depends on the network requirements. If every node j adjacent to *node i* is suspicious, then *node i* sends the identification number of that node to the cluster head or sink to evaluate a voting process on *node j* according to the communication qualities with other nodes in the network. In this study, each *node i* in the CLA suspects a neighboring node j if Quality_j value is below 0.5.

The outcome of the vote determines whether the judged node is trusted or distrusted. The trust ratio TR_j is defined as the number of nodes that distrust node j divided by the number of nodes participating in the vote. Whether a node is trusted or suspected is determined as follows:

1. If $\text{TR}_j < 25\%$, more than 75% of the nodes participating in the voting distrust *node j*. The suspicious *node j* is then removed from the routing. In addition, the cluster head announces all neighbors of *node j* to join its blocklist. The action probability in each CLA is set to 0, and the CLA that selected the malicious node reselects the next step based on the probabilities of the other actions.

2. If $25\% \le \mathrm{TR}_j \le 75\%$, more than 25% but less than 75% of the participating nodes distrust *node j*. The environment gives an unfavorable response to the CLA, that is, $\beta = 1$; this action is fined in the CLA with the dynamic parameter $b = (1 - \mathrm{TR}_j)/2$ (LR-P model).
3. If $\mathrm{TR}_j > 75\%$, less than 25% of the participating nodes distrust node *j*. The environment gives favorable feedback to the CLA and β is set to 0. This action of the automaton is rewarded with the dynamic parameter $a = \mathrm{Mid}_{\mathrm{vote}} \times \mathrm{TR}_j$ where $\mathrm{Mid}_{\mathrm{vote}}$ represents the average vote value of the nodes in the voting process.

6 | SIMULATION MODEL

The proposed method was simulated in MATLAB R2016a software. The simulation was run on a computer system with a corei7 processor having 4 GB of main memory. The operating system was Windows 7. The network was simulated for 60 min and some network communication statistics were collected. The number of network nodes was 100, including 20 malicious nodes. The information required for the research was collected while monitoring the network performance. To build a forward cell attack detection system based on CLAs, 24 000 and 3000 data patterns were used in the training and validation phases, respectively, and 5000 patterns were retained as the test data. Based on the CLA, each pattern was categorized as attack or non-attack. Attacks were identified in the data extracted from access and data transfer in the IoT.

The accuracy of the proposed system was evaluated by cross-validation. In classification applications, cross-validation determines the usefulness of the model in practical scenarios. In each category, the answers fall into one of four categories:

True Positive (TP): Records in this category are positive and correctly identified as positive by the model.

True Negative (TN): Records in this category are negative and correctly identified as negative by the model.

False Positive (FP): Records in this category are negative but are incorrectly identified as positive by the model.

False Negative (FN): Records in this category are positive but are incorrectly identified as negative by the model.

To compare the accuracy of the classifier based on the proposed method, we adopted the Precision and Recall measures, respectively, defined as

$$\begin{aligned} \mathrm{Precision} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \\ \mathrm{Recall} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}. \end{aligned} \tag{12}$$

The Precision indicates the proportion of positive predictions among the categorical positive predictions. The Recall is the ratio of the number of correctly identified positive predictions to the total number of available positive predictions.

7 | EXPERIMENTS AND DISCUSSION

In this section, the performance of the proposed method is compared with those of existing methods, specifically with two highly regarded and well-known methods for detecting attacks on IoT. The proposed method was simulated with different input parameters and evaluated on the above performance criteria. Figures 2, 3, and 4 respectively plot the true positive rate (TPR), Precision metric, and Recall rate as functions of node number in the attack detection system. The results of the proposed system are compared with those of the automata model [26] and SecTrust [27]. The CLA-based IoT IDSs achieved higher TPR and Precision criteria than the existing methods. The improved correct positive rate indicates that our method better detects attacks than the other two methods and detects a higher percentage of actual attacks.

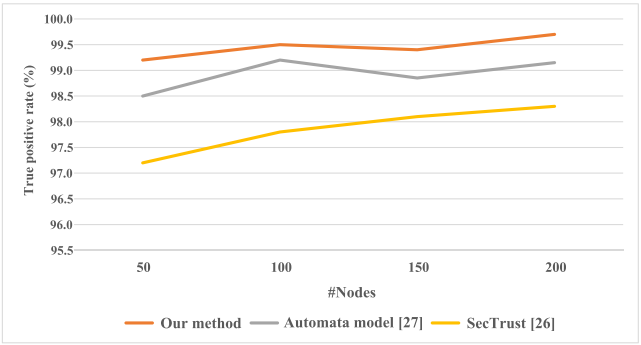

FIGURE 2 True positive rate versus number of nodes

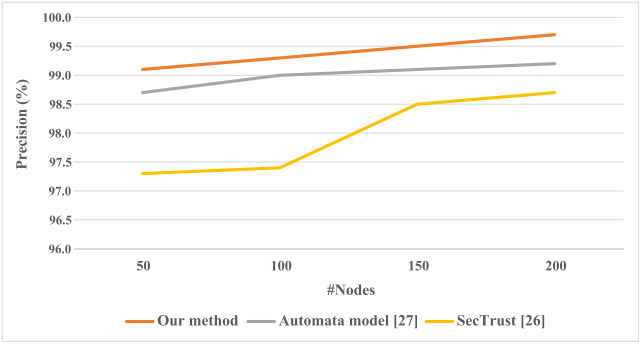

FIGURE 3 Precision versus number of nodes

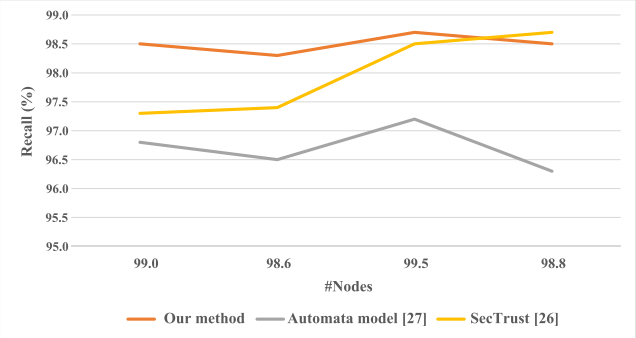

FIGURE 4 Recall versus number of nodes

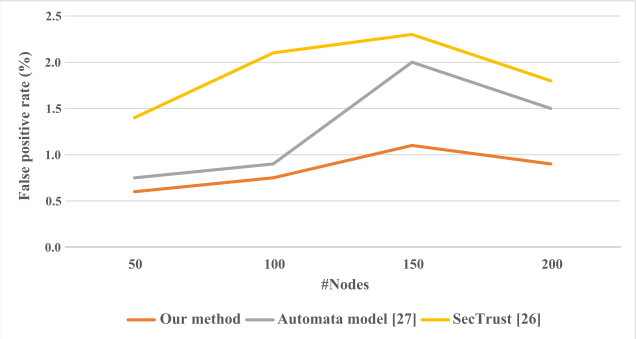

FIGURE 5 False positive rate versus number of nodes

The Recall criterion represents the proportion of records that were identified as attacks in practice; as such, it is an essential qualitative criterion for intrusion attack detection systems. Ideally, the Recall should be low. A high Recall means that some attempted attack accesses are incorrectly identified as normal. As this criterion represents the percentage of attacks that have been mistaken for normal accesses, a network with high Recall can be sabotaged by intruder and hostile nodes, with negative consequences and possibly severe damage.

As shown in Figure 5, the false positive rates (FPRs) were consistently lower in the proposed method than in the other studies, confirming that the network correctly detected most attacks and misclassified only a few normal records as attacks. Note that the FPR was higher in the 150-node network than in networks with more and fewer than 150 nodes. In the FPR, the numerator is the number of false positive detections and the denominator is the total number of network nodes. In the 200-node network, the proposed method operated properly and the number of normal records incorrectly identified as an attack was not significantly increased, meaning that the form of the retention did not change significantly. The decrease in FPR in the 200-node network was entirely attributed to the increase in the denominator (an increase of 50) from that of the 150-node network. In the 200-node simulation, the amount of communication in the network was high and the attack detection was efficient. The effectiveness of attack detection can be attributed to the comprehensive communication quality and the voting decisions of nodes regarding the reliability of their neighbors. In this case, the number of attacks was no higher than in the 150-node network. An increase in the number of network nodes with no increase in false positive detection will reduce the FPR; therefore, the proposed method better suppresses the vulnerability of the network than the existing methods.

Further comparisons between the proposed and existing methods are discussed below. In the first experiment, the number of attackers in the 100-node network was increased from 2 to 30. Figures 6 and 7 respectively plot the packet loss rates and throughputs of the three methods as functions of number of attacks. As shown in Figure 6, the number of lost packets initially rose steeply with number of attacks, and the package delivery rate declined. As the number of attacks increased, the attacker nodes blocked the packet forwarding and disrupted the network routing. By calculating the comprehensive communication quality of each node, the proposed method detected the attacking nodes and efficiently eliminated them from the list of node neighbors. As the attacking nodes were identified, the quality of the network routing increased; therefore, the number

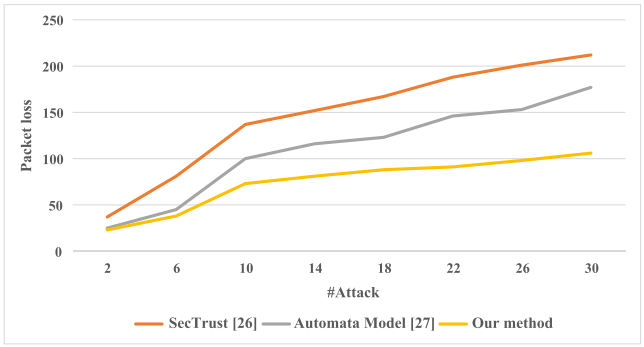

FIGURE 6 Packet loss versus number of attacks

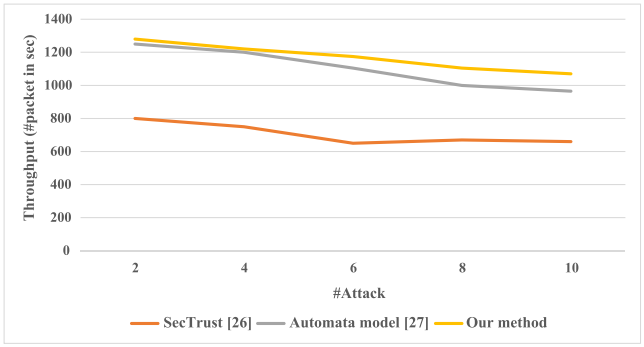

FIGURE 7 Throughput versus number of attacks

of lost packets was robust against further increase in number of attacks. Overall, the packet loss was lower in the proposed method than in the previous studies (Figure 7), because the CLA rapidly identifies and isolates attacks. Through collaborative exchange of trust values between active nodes, the proposed method detects attackers with higher accuracy than the methods presented in [25,26].

Because energy consumption in IoT and wireless sensor nodes critically affects the network performance, we attempted to increase the network security and reliability while maintaining low energy consumption. Accordingly, our method considers the balance between network reliability and energy consumption. The energy consumptions of the proposed and previous methods are compared in Figure 8. The power consumption was clearly lower in our method than in the previous methods, and the difference grew with increasing number of nodes. Here, the number of network nodes was varied from 100 to 200. The much lower average energy consumption in the network run by our method will prolong the network life.

As demonstrated in the above results, the proposed method improved the energy consumption in networks with different numbers of network nodes. High energy consumption is known to reduce the lifetime of a network. In the proposed method, adaptive fines and rewards based on energy consumption in the IoT create an energy-load balance between all nodes. The reduced power consumption improved the efficiency of the proposed CLA-based method and extended the network lifetime. Energy efficiency is crucial for maximizing the lifetime of sensor nodes in IoT, as nodes typically draw power from a limited-capacity battery source. Furthermore, most IoT applications require secure and long-term operation of the sensor nodes.

Figure 9 plots the network lifetimes of the three methods as functions of network nodes. In this simulation, the transmitted traffic rate was 10 packets per second. The network lifetime was defined as the time between the start of the simulation and turn-off of the first sensor. As shown in the figure, the proposed protocol extended the network lifetime from those of the previous studies. The proposed method dynamically learns the traffic patterns and selects the most efficient paths based on CLA; consequently, the energy consumption is reduced and the lifetime is lengthened because each node predicts the traffic and forwarding data via several paths; moreover, the workload is distributed across different paths rather than concentrated along the most used paths.

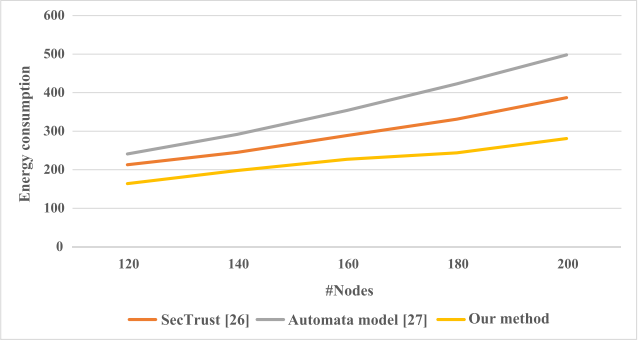

FIGURE 8 Energy consumption versus number of nodes

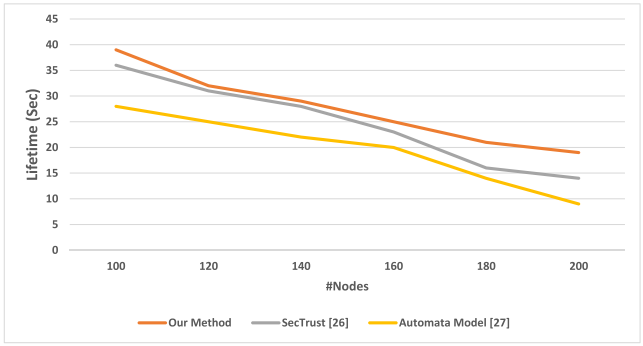

FIGURE 9 Network lifetime versus number of nodes

As the number of nodes increased, the proposed method and automaton-based method consistently consumed the least and most energy, respectively. The low power consumption directly explains the long lifetime of the proposed method (note that the lifetime trends echo the energy consumption trends). These results prove that lowering the power consumption extends the lifetime of network nodes.

Increasing the number of network nodes increased the number of packets exchanged in the network, thereby increasing traffic and congestion and reducing the network life. This inference is reasonably expected.

8 | CONCLUSION

This paper proposes a binary adaptive learning model for detection of IoT attacks. The method uses CLA with parameters that adapt to energy consumption, the comprehensive communication quality, and voting by neighbors. A new CLA-based architecture for attack detection was then described. Comparative simulation results confirmed the high quality of the proposed method. The proposed model accurately detects IoT attacks while increasing the network efficiency. The existing methods for IoT-attack detection ignore the energy consumption, which is a crucial determiner of network performance. The proposed

method increases both the security and reliability of the network and establishes a balance between network reliability and energy consumption. However, our method requires full synchronization of the network nodes. In future work, we will develop a fully distributed cluster-based structure for IoT that removes the limitation of complete coordination between the network nodes.

CONFLICT OF INTEREST

The authors declare that there are no conflicts of interest.

ORCID

Javad Dogani https://orcid.org/0000-0002-0466-2939

REFERENCES

1. L. Lee and K. Lee, *The Internet of Things (IoT): Applications, investments, and challenges for enterprises*, Bus. Horiz. **4** (2015), 431–440.
2. M. M. Noor and W. H. Hassan, *Current research on Internet of Things (IoT) security: A survey*, Comput. Netw. **148** (2019), 283–294.
3. E. B. Priyanka, C. Maheswari, and S. Thangavel, *A smart-integrated IoT module for intelligent transportation in oil industry*, Int. J. Numer. Model. **1** (2020), article no. e2731.
4. L. García et al., *IoT-based smart irrigation systems: An overview on the recent trends on sensors and IoT systems for irrigation in precision agriculture*, Sensors **20** (2020), article no. 1042.
5. F. Mehmood et al., *A novel approach towards the design and implementation of virtual network based on controller in future IoT applications*, Electronics **9** (2020), article no. 604.
6. F. John Dian, R. Vahidnia, and A. Rahmati, *Wearables and the Internet of Things (IoT), applications, opportunities, and challenges: A survey*, IEEE Access **8** (2020), 69200–69211.
7. K. Ashton, *That 'Internet of Things' thing*, 2009, Available from: http://www.rfidjournal.com/articles/view?4986
8. K. Shafique et al., *Internet of Things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios*, IEEE Access **8** (2020), 23022–23040.
9. L. Greco et al., *Trends in IoT based solutions for health care: Moving AI to the edge*, Pattern Recognit. Lett. **135** (2020), 346–353.
10. F. Al-Turjman, H. Zahmatkesh, and R. Shahroze, *An overview of security and privacy in smart cities IoT communications*, Trans. Emerg. Telecommun. Technol. (2019), e3677.
11. V. Hassija et al., *A survey on IoT security: Application areas, security threats, and solution architectures*, IEEE Access **7** (2019), 82721–82743.
12. N. Chaabouni et al., *Network intrusion detection for IoT security based on learning techniques*, IEEE Commun. Surv. Tutor. **21** (2019), 2671–2701.
13. M. Gajewski et al., *Anomaly traffic detection and correlation in smart home automation IoT systems*, Trans. Emerg. Telecommun. Technol. **1** (2020), article no. e4053.
14. L. Chettri and R. Bera, *A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems*, IEEE Internet Things J. **7** (2020), 16–32.
15. M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, *Intrusion detection systems for IoT-based smart environments: A survey*, J. Cloud Comput. **7** (2018), 1–20.
16. M. D. S. S. Romeo, *Intrusion detection system (IDS) in Internet of Things (IoT) devices for smart home*, Int. J. Psychosoc. Rehabil. **23** (2019), 1217–1227.
17. S. N. Mohanty et al., *An efficient lightweight integrated blockchain (ELIB) model for IoT security and privacy*, Future Gener. Comput. Syst. **102** (2020), 1027–1037.
18. Y. Maleh and A. Ezzati, *Towards an efficient datagram transport layer security for constrained applications in Internet of Things*, Int. Rev. Comput. Softw. **11** (2016), 611–621.
19. P. Sudhakaran, *Energy efficient distributed lightweight authentication and encryption technique for IoT security*, Int. J. Commun. Syst. (2019), article no. e4198.
20. J. Kari, *Theory of cellular automata: A survey*, Theor. Comput. Sci. **334** (2005), 3–33.
21. H. H. Pajouh et al., *A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks*, IEEE Trans. Emerg. Topics Comput. Secur. **7** (2019), no. 2, 314–323.
22. R. Rani, S. Kumar, and U. Dohare, *Trust evaluation for light weight security in sensor enabled Internet of Things: Game theory oriented approach*, IEEE Internet Things J. **6** (2019), 8421–8432.
23. T. D. Nguyen et al., *DI OT: A federated self-learning anomaly detection system for IoT*, in Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS), (Dallas, TX, USA), July 2019.
24. H. Zhu et al., *Detection of selective forwarding attacks based on adaptive learning automata and communication quality in wireless sensor networks*, Int. J. Distrib. Sens. Netw. **14** (2018), no. 11, 1–15.
25. D. Airehrour, J. Gutierrez, and S. Kumar Ray, *A lightweight trust design for IoT routing*, in Proc. IEEE Int. Conf. Dependable, Auton. Secure Comput. & Int. Conf. Pervasive Intell. Comput. & Int. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech), (Auckland, New Zealand), Aug. 2016.
26. Y. Fu et al., *An automata based intrusion detection method for Internet of Things*, Mob. Inf. Syst. **2017** (2017), 1–13.
27. X. Yan et al., *Trustworthy network anomaly detection based on an adaptive learning rate and momentum in IIoT*, IEEE Trans. Industr. Inform. **16** (2020), no. 9, 6182–6192.
28. H. Wu and W. Wang, *A game theory based collaborative security detection method for Internet of Things systems*, IEEE Trans. Inf. Forensics Secur. **13** (2018), no. 6, 1432–1445.
29. T. Gu et al., *Towards learning-automation IoT attack detection through reinforcement learning*, in Proc. IEEE Int. Symp. World Wirel. Mob. Multimed. Netw. (WoWMoM), (Cork, Ireland), Aug. 2020.
30. S. Venkatraman and B. Surendiran, *Adaptive hybrid intrusion detection system for crowd sourced multimedia Internet of Things systems*, Multimed. Tools Appl. **79** (2020), 3993–4010.

31. A. A. Diro and N. Chilamkurti, *Distributed attack detection scheme using deep learning approach for Internet of Things*, Future Gener. Comput. Syst. **82** (2018), 761–768.
32. S. Rezvy et al., *An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks*, in Proc. Annu. Conf. Inf. Sci. Syst. (CISS), (Baltimore, MD, USA), Mar. 2019, 1–6.
33. B. Hussain et al., *Deep learning-based DDoS-attack detection for cyber–physical system over 5G network*, IEEE Trans. Industr. Inform. **17** (2021), no. 2, 860–870.
34. W. Q. Li, Q. Yu, and L. X. Ma, *Cellular automata-based WSN energy saving technology*, Adv. Mat. Res. **546–547** (2012), 1334–1339.

AUTHOR BIOGRAPHIES

Javad Dogani received his B.Sc. degree in software engineering from Technical and Vocational University, Shiraz, Iran, in 2010 and an M.Sc. degree in software engineering from Shiraz University, Shiraz, Iran, in 2012. He has been an assistant professor in the Department of Electrical and Computer Engineering at the University of Hormozgan from 2014 to 2018. His main research interests include wireless networks, Internet of Things systems, big data, cloud computing, and deep learning.

Mahdieh Farahmand received her B.Sc. degree in software engineering from the Shahid Bahonar University of Kerman, Kerman, Iran, in 2014 and an M.Sc. degree in Artificial Intelligence from the Science and Research Branch, Islamic Azad University, Tehran, Iran, in 2018. Her main research interests include wireless networks, Internet of Things systems, distributed systems, and deep learning.

Hassan Daryanavard received his B.Sc. degree in electrical engineering from Shahid Rajaei University, Tehran, Iran, in 2008 and M.Sc. and Ph.D. degrees in Digital Electronics from the University of Tabriz and Shahid Beheshti University in 2010 and 2015, respectively. He is now an assistant professor in the Department of Electrical and Computer Engineering at the University of Hormozgan. His main research interests include FPGA embedded system design and Internet of Things systems.

How to cite this article: J. Dogani, M. Farahmand, and H. Daryanavard, *A new method to detect attacks on the Internet of Things (IoT) using adaptive learning based on cellular learning automata*, ETRI Journal **44** (2022), 155–167. https://doi.org/10.4218/etrij.2021-0044