Quarterly Journal of
Optimization In Soft Computing

Vol. 1, Issue. 2, Winter 2023

Journal of Optimization of Soft Computing (JOSC)

Vol. 1, Issue 2, pp: (1-17), December-2023

Journal homepage: https://sanad.iau.ir/en/Journal/josc

Research paper

An efficient energy-aware trust-based RPL protocol for Internet of Things

Farzaneh Kaviani[1], Mohammadreza Soltanaghaei[1*], Farsad Zamani Boroujeni[2]

[1]*Department of Computer Eng., Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran.*
[2]*Department of Computer Engineering, Islamic Azad University, Science and Research Branch, Tehran, Iran.*

Article Info

Article History:
Received: 2023/11/15
Revised:
Accepted: 2024/1/7

Keywords:
Internet of Things, Trust models, Energy-aware, RPL protocol, IDS.

**Corresponding Author's Email Address: soltan@khuisf.ac.ir*

Abstract

The Internet of Things (IoT) is considered as one of the newest communication technologies for various applications. On the other hand, it has faced many challenges that one of the most important of which is related to the security. Due to many limitations, IoT is very vulnerable to attacks, and it is highly exposed to attacks due to its sensitive applications. Various studies have been introduced to improve IoT security. Most of methods have focused on improving the security of the RPL protocol (as the IoT routing standard) based on the development of trust models. However, most of these researches have considered behaviors to calculate the value of trust. This way of assessing trust is not enough due to the widespread attacks of malicious nodes. In this paper, an improved method is proposed based on RPL development utilizing trust models with intrusion detection system. The proposed method focuses on three important principles, including establishing secure and reliable routing topology, evaluating trust, and detecting malicious nodes. In the first step, the network routing topology is formed based on the trust and conditions of the nodes. In the second step, in accordance with the data exchanges, the trust of the nodes is evaluated and malicious factors are identified. The simulation results using Cooja indicated the superiority of the proposed method in improving routing reliability and data exchange over previous operations.

1. Introduction

The Internet of Things (IoT) is widely used in a wide range of areas, including transportation, military applications and emergencies [1, 2]. The most important features of these without infrastructure networks are the absence of central power and distribution of variable topology network, many deficiencies, especially in consumption resources, node self-organization and multi-hop data exchange [3, 4]. These special and unique features of these networks, have led to various issues (especially the issue of trust and support) that are different from other wireless networks. This difference, along with the specific and limited nature of IoT, has led to these networks being more vulnerable to attacks than other networks [5]. However, important and sensitive applications, such as military, have made the IoT highly susceptible to various attacks. The discussion of routing and data exchange as the most important element of the IoT is no exception to this rule. Building trust, especially in the field of routing and data exchange, is one of the most important issues of the IoT. Accordingly, extensive researches have been introduced to improve the security and trust of IoT routing and exchange based on trust models.

Trust models are considered as a complementary tool for security systems that they provide good ability to detect malicious nodes [6]. Trust models, in addition to high efficiency to detect malicious nodes, are very compatible with the characteristics of the IoT, and they are considered as one of the most effective techniques for implementing trust in these networks [7]. In these models, the network nodes by monitoring and analyzing behaviors assess trust. Trust relations can be used to make effective decisions at the nodes, for example, selecting preferred parents or providing trusted routing. Therefore, many researches have been presented based on the importance of trust models in improving IoT security. The most purpose of these models is to improve the security of the RPL protocol. However, many researches focus on behaviors to calculate the value of trust. This method of assessing trust is not sufficient due to the widespread attacks of malicious nodes and is particularly vulnerable to intelligent attacks [8, 9].

As mentioned in trust models, the trust value of nodes is calculated based on behaviors. This evaluation method provides a cumulative value of previous behaviors of nodes that it is not sensitive enough to detect malicious elements [9]. In other words, it does not provide the ability to detect suspicious elements and deceptive attacks. In fact, malicious nodes can mask negative behaviors because of good behaviors and not be recognizable in exchange. These deceptive behaviors include a number of attacks that the most important of them are on-off attack, selective forwarding and gray hole. The purpose of this article is to improve the response to these attacks.

This paper proposes an improved method called Energy Aware Trust-based Efficient RPL for IoT (EATE-RPL) based on the development of the RPL protocol using trust models and intrusion detection systems. This method improves the reliability of trust calculations and the detection of deceptive attacks by examining the integrity of node behavior. Based on this, the proposed method identifies malicious agents and enables secure routing by removing and quarantining them.

The RPL routing protocol is accepted as the routing standard for IoT and has been widely used in various applications [10, 11]. RPL provides users with the ability to optimize and define routing according to their needs. This article focuses on the RPL capability and tries to improve it based on trust relationships. The main contribution of this article can be summarized as follows:

This paper proposes a new objective function, known as the energy and trust-aware objective function. EATE-RPL uses it to select preferred parents, which is a result of the trust and conditions of the network nodes. It introduces a mechanism to evaluate the trust of nodes, predicts complementary intrusion detection system for more effective detection of malicious agents and intelligent attacks and implements and EATE-RPL evaluates in different scenarios and compares its performance with previous researches.

In the continuation of the article, the PRL protocol will be briefly introduced. In the third section, previous researches will be reviewed. Details of the proposed EATE-RPL will be provided in Section four. In the fifth section, the proposed protocol based on Cooja software is simulated and its performance will be evaluated. The article will conclude in the sixth section.

2. RPL Routing Protocol

The IoT consists of a large number of sensor nodes and one or more specific nodes called roots. Sensor nodes send all their data and reports to the root node. However, the sensors are unable to communicate directly with the root due to radio range limitations. Accordingly, in the IoT, communications and exchanges of nodes with roots take place in a multi-hop with the participation of other nodes. So, nodes operate on the basis of routing protocols [12, 13]. Routing protocols enable communication between sensors and roots by considering and discovering intermediate routes. Among all types of routing protocols, RPL is accepted as the routing standard for the IoT [14]. This protocol has been introduced by one of the Internet Engineering Task Force (IETF) working groups called Routing Over Low Power and Lossy (ROLL) [15]. The group focuses on the routing of Low power and Lossy Networks (LLN), including IoT.

RPL resolves the needs of IoT routing well because of its unique capabilities, but there are still many

challenges. One of the most important of these challenges is related to the security. The function of this protocol is briefly discussed below.

The RPL protocol generally has three control messages including DIO (DODAG information object), DIS (DODAG information solicitation) and DAO (destination advertisement object). DIO and DAO are used to create and update the DODAG graph, and DIS is used to manage network topology changes (such as adding a new node). The RPL forms the DODAG graph through two messages of DIO and DAO, which the nodes will be able to communicate with the root through this graph [14].

To create a DODAG graph, the root node generates a DIO message and sends it all over the network nodes. Each node on the transmitter radio board receives a DIO. After receiving the DIO, the receiving node makes a decision based on evaluating the objective function (OF) in relation to selecting the sender as a parent and resending the DIO [11]. If the node selects the sender as the parent, it updates the DIO packet and resends it in broadcast. Otherwise, it will not send the packet in order to prevent looping. This process continues until all nodes receive the DIO. If a node receives a DIO from multiple neighbors, it selects its parent from the list of candidate parents according to the criteria and constraints defined by the objective function. After sending the DIOs, the node with the best objective function is selected as the preferred parent and is notified by sending a DAO message. This process continues until DAOs are received by the root. Upon receipt of DAOs by the root, the DODAG graph is created, and the nodes are able to communicate with the root through the generated graph. If a new node is added to the network, this extension is managed by sending a DIS packet. The new node sends DIS to its neighbors for requesting DIO, and after receiving it the node with the best objective function is selected as the preferred parent and a DAO packet is sent to it. This will make the new node as a member of the DODAG graph. A more comprehensive explanation is provided in [14].

3. Related works

As mentioned, the importance of IoT applications has made these networks highly susceptible to various attacks. The purpose of most of these attacks is to disrupt the routing process and data exchange, especially to disrupt the performance of the RPL protocol. Accordingly, the extensive researches have reviewed RPL protocol security and the challenges of this protocol [16 and 17]. In [18] Raoof et al. considered the security features of the RPL and the vulnerabilities of this protocol. They introduced various methods of detecting and counteracting attacks. According to this research, attacks on RPL are divided into two types of attacks from wireless sensor networks and attacks focused on RPL features. The results of this study showed that although several security features have been envisaged for the RPL protocol, this protocol still faces wide-ranging challenges. Research [19] provided a detailed review of the types of attacks on RPLs and counteraction techniques. Verma et al. divided the attacks on RPL into three categories: resource-focused, topology-oriented, and traffic-oriented.

On the other hand, extensive methods have been introduced to deal with attacks, which they are two types of intrusion detection systems (IDS) and trust-based methods. In the following, the types of methods related to each category are introduced separately and important details related to each category are discussed.

3.1. IDS-based methods

In this section, IDS-based attack response techniques will be discussed. IDS is one of the most effective intrusion detection solutions that has been widely used in IoT [16]. References [20, 21] provides comprehensive reviews of applied IDS-based models for IoT. In [22] IDS models are evaluated and classified based on parameters and applied techniques. In [23], IDS models based on learning algorithms have been evaluated and analyzed. In [24-26] methods called Compression Header Analyzer Intrusion Detection System (CHA-IDS), Hybrid Intrusion Detection System (HIDS) and Detection of routing attacks in RPL-based IoT (DETONAR) have been introduced, respectively. Introduced methods focused on intrusion detection techniques that their aim is to detect routing attacks. Although these methods have been successful in detecting attacks, their implementation is highly complex. In [27, 28], the

proposed techniques are classified based on IDS to improve RPL security and the vulnerabilities of these techniques are evaluated. Simoglou et al. [16] discussed about the design requirements of IDSs for the RPL protocol and focused on the problems of these techniques. Arış et al. [29] proposed two techniques for combating version number attacks (VNA). In [30, 31], signature-based intrusion detection methods are introduced. Although these methods are simple and fast, but due to the extent of the attacks, these methods are only resistant to some attacks. Luangoudom et al. [32] and Soni et al. [33], respectively, introduced the methods called svBLOCK: mitigating black hole attack in low-power and lossy networks, and Link Hop Value-based Intrusion Detection System (L-IDS) to detect the black-hole attacks. So, svBLOCK focuses on checking the authenticity of control messages, and L-IDS focuses on evaluating the number of hops. The disadvantage of these methods is the high delay. Mayzaud et al. [34] proposed a distributed monitoring architecture to detect DODAG incompatibility attacks. In [35] the same approach was developed to identify rank attacks. In this solution, the monitoring nodes have the ability to cooperate with each other to detect. References [36-38] proposed the Deep Learning (DL) model as IDS to identify routing attacks. The results show good performance in terms of IDS accuracy. However, the DL fitting time is very long.

Studies on intrusion detection models show that although these methods are successful in detecting attacks, they do not provide the ability to establish trust in routing and exchanges.

3.2. Trust-based methods

In this section, trust-based methods will be discussed. Trust is one of the key tools for dealing with malicious nodes and supporting routing and trusted exchanges. It has been widely used in the IoT.

In recent years, the issue of trust management has become more widely used in the IoT. In [39], Junior et al. considered trust models, especially in relation to data transmission trust, and reviewed the challenges associated with this field. Mohammadi et al. [40] considered trust-based models and these methods are classified into three different layers of the IoT based on application parameters and techniques. In [41], trust management in Social Internet of Things is discussed and the challenges in this area are reviewed and several suggestions are introduced to resolve them. In [42], the use of blockchain technologies to improve trusted exchanges is considered and these techniques are compared with traditional trust models. This study shows that the blockchain can improve trusted exchanges. Pourghebleh et al. [9] classified trust management techniques into four classes including recommendation-based, prediction-based, policy-based, and reputation-based and evaluated them based on trust criteria. Studies show that there are still widespread challenges to trust management, especially trust in routing and exchanges. Researches of [7, 8] provided comprehensive reviews of applied trust management models for IoT. These studies have been performed by considering direct and indirect observations (recommendations) based on distributed, semi-distributed and centralized designs. Verma et al. [19] examined a variety of trust models, especially applied trust models to improve trusted RPL protocol exchanges, and increasing energy consumption and incompatibility with the dynamics of nodes are introduced as the most important challenges in trust models. In reference [6] attacks on RPL and trust models against these attacks have been investigated and suggestions have been made to provide a secure routing model.

In the following, some of the most important models of trust are introduced and examined in a more specialized way.

Most previous researches for trust management in the IoT are based on social trust that the important goal of them is to improve RPL routing and exchanges trust. For example, Karkazis et al. [43] introduced the Packet Forwarding Indication (PFI) criterion as a measure of trust for RPL. In [44, 45], models based on multi-metric evaluation and fuzzy logic are introduced, respectively. The goal of research in [44] is to improve the trust and reliability of routing, and for this purpose it operates based on trust, residual energy and ETX. Fuzzy, dynamic and hierarchical trust model (FDTM-IoT) is integrated as a objective function

in RPL and its goal is to improve unreliability as one of the most important features of trust [45]. Djedjig et al. introduced a new trust-based measure for RPL routing [46, 47]. In these methods, direct trust is evaluated based on energy trust, trust in behavior, trust in link, so that the final trust is obtained based on the result of this evaluation by combining indirect trust. Muzammal et al. [48] improved routing trust and RPL exchanges. In this study, trust is estimated based on two factors of direct and indirect trust with considerations related to the dynamics and energy of the nodes. Hassan et al. [49] proposed a control layer-based trust mechanism for supporting secure routing (CTrust-RPL). CTrust focuses on the high energy consumption of trust models that to improve it, trust calculations have been transferred to the higher layer. In [50], trust is evaluated based on successful exchanges between two nodes, and based on it, decisions are made to identify malicious factors. Hashemi et al. [51]Proposed a Dynamic and comprehensive trust model for IoT and its integration into RPL (DCTM-IoT). DCTM is based on direct and indirect trust. It also includes other criteria such as energy, dynamic and ETX (Expected transmission count) in trust assessments to provide a dynamic trust model. In [52], a method for detecting black hole attacks was presented by focusing on assessing the status of graph paths relative to the ratio of successful delivery to total transmissions. Airehrour et al. [53, 54] proposed the trust-aware RPL routing protocol (SecTrust-RPL). In SecTrust, the final trust of the nodes is calculated based on the direct trust and the recommendations received from the two-hop parents. In [55] RPL trust is improved based on a logistic regression model. Logistic regression is used to predict the behavior of nodes based on the value of integrated trust.

According to what was presented in the previous sections regarding intrusion detection systems and trust models, if IDS-based methods are successful in detecting malicious nodes, these models do not provide the ability to support routing and trusted exchanges. While trust-based models have been successful in supporting routing and trusted exchanges, most of these methods deal with some attacks, including deceptive attacks (such as on-off attacks, selective forwarding, etc). In addition, most of the researches presented have focused on securing topology routing and trusted exchanges. Although this performance covers the needs of trust well, other aspects of routing, including the reliability of routing and exchanges, still need to be considered in order to maintain the quality of exchanges. Focusing only on trust, even in some situations, may lead to increased intermediate route lengths, exchange inefficiency, and reduced service quality because trust is the only criterion for decision making.

In this research, a method called Energy Aware Trust-based Efficient RPL (EATE-RPL) is proposed based on RPL optimization with a focus on models of mutual trust using IDS. EATE-RPL focuses on three important principles including establishing secure and reliable routing topology, evaluating trust, and detecting malicious nodes. On this basis, in addition to covering security needs, the reliability of transactions has also been provided.

4. Proposed EATE-RPL

The purpose of EATE-RPL is to improve RPL protocol routing and exchanges. For this purpose, the EATE-RPL operation is divided into two general steps. In the first step, based on the trust and conditions of the nodes, the focus is on creating a secure and reliable routing topology. In the second step, along with the data exchanges, the trust of the nodes is evaluated and malicious factors are identified. In addition to providing communication trust, EATE-RPL also covers the reliability of data exchanges. Overall, the EATE-RPL focus to improve reliability and trust of RPL can be divided into the following:

1-Creating secure and reliable routing topology.

2-Proposing a new objective function called Trust and Reliability Aware Object Function (TQAOF) that it is evaluated by focusing on residual energy, trust, ETX, and node rankings.

3-Establish trust by focusing on trust-based and IDS-based techniques with the aim of improving detection of malicious nodes.

In the following, first the details of TQAOF evaluation are provided and then how to develop

RPL in order to implement EATE-RPL will be discussed.

4.1. Extended RPL with TQAOF

TQAOF is an objective function that provides a result of trust, reliability, and node rankings. The selection of the preferred parents is based on the evaluation result of this function. Based on previous researches, only communication security has been discussed and supporting the reliability of data exchanges has not been considered. However, TQAOF supports both needs well. This criterion is evaluated in terms of residual energy, ETX, node ranking and reliability.

In general, the RPL protocol has two functions including OF0 and ETX to select the preferred parents that one of which can be used as needed. In the objective function of OF0, the selection of the preferred parents is done with a focus on proximity to the root [56], and in the objective function of ETX, this selection is based on the reliability of the link [57]. TQAOF is an improved objective function that includes the outcome of rank, reliability (residual energy and ETX) and node trust with the aim of creating a secure and reliable routing topology graph.

To detail the EATE-RPL function, in the first section the applied criteria have been formulated for calculating TQAOF, then routing and network topology is discussed for the TQAOF objective function. In the following, the details of malicious detection and isolation are described.

4.2. TQAOF Metrics

The parameters used to evaluate TQAOF reflect the trust, rank, and reliability of the nodes. These parameters are used to find the best parent and are as follows:

4.2.1. Trust

In EATE-RPL, the trust evaluation is based only on direct trust and is not used indirect trust (recommendations) due to attacks and increased overheads. Exchanges of recommendations, on the one hand, have led to an increase in overheads, which is contrary to IoT limitations, and on the other hand, will lead to attacks such as Bad-mouthing and Good-mouthing. So, in EATE-RPL, the assessment of trust has been done only with a focus on direct trust.

Direct trust is related to the behavioral history of nodes that it provides a result of trust in node communications. Nodes in IoT networks often cooperate and communicate with other nodes (neighboring nodes). Considering communication behaviors provides the ability to detect the normal or abnormal operation of nodes. However, it should be noted that due to severe IoT limitations, many causes lead to communication disruptions and data loss. In fact, the cause of failed communication may not be the only malicious behavior, and communication channel disruptions or decrease of node efficiency may be the cause. Therefore, evaluating trust with regard to the history of communication behaviors will be uncertain. In EATE-RPL, improve this issue is based on the concept of uncertainty. But at the same time, this kind of evaluation will not be safe enough, especially in the face of deceptive attacks. In this type of attack, the malicious agents act by changing their behaviors (change in right and wrong behaviors) in such a way that with performing negative behaviors, their trust is greater than the threshold value of malicious detection. This is done with the intention of deception. Considering three concepts can have great effects in improving this issue: 1) The malicious node can't gain high trust in low value transactions and is misused it in high value transactions. This refers to the need to calculate variable trust in the interaction value. 2) Trust is hard to come by and easy to lose. Trust in negative and positive transactions should not have the same effect. The effect of negative transactions on reducing trust should be greater than the effect of positive transactions on increasing trust. 3) The malicious agent can't gain high trust in the old transactions and abuse it in the current transactions. This refers to the higher value of current transactions in the calculation of trust. The EATE-RPL seeks to improve the detection of deceptive and intelligent attacks by considering these three concepts alongside the topic of uncertainty.

The EATE-RPL uses the watchdog mechanism to assess trust and to make direct observations during exchanges. Based on the results of this monitoring, the ability to analyze and detect the behavior and

performance of nodes during exchanges will be provided. Then, based on the results of this study, the trust of the nodes is evaluated based on equation (1). So that $T_{i,j}$ is equal to the trust of node i to j, $NT_{i,j}$ and $PT_{i,j}$is equivalent to the new and past trust of node i to j, θ is value coefficient to the new and old trust of nodes. So that $0 <\theta <1$. The value of this coefficient in EATE-RPL is considered equal to 0.5. $PT_{i,j}$ is the past trust that it is saved in the node table. In the initial moment the trust value of nodes is equal to 0.5 (average trust).

$$T_{i,j} = \theta . NT_{i,j} + (1 - \theta) . PT_{i,j} \quad (1)$$

4.2.2. Energy Remainder (ER)

This metric is one of the reliability criteria and refers to the residual energy of the nodes. The ER criterion is evaluated based on energy consumption to the initial energy of the nodes. In equations (2) and (3), the details of energy consumption of nodes and evaluation of residual energy of nodes are provided, respectively. Thus, tx, rx, LP and cpu are equivalent to transmission, receive, low power mode (such as node sleep mode) and processing mode (when radio components are off), respectively. P_{tx}, P_{rx}, P_{cpu}, and P_{LP} are equivalent to the energy requirements associated with the four modes, respectively. T_{tx}, T_{rx}, T_{cpu}, and T_{LP} are the length of sending, receiving, processing, and low power mode, respectively [58]. In equation (3) EC_i, IE_i and ER_i are equal to the amount of energy consumed, the initial energy and the ratio of the remaining energy of node i, respectively. In IoT networks, power consumption of node is depends on its location in the network. This metric is effective in preventing the selection of low energy nodes and premature depletion of energy of some intermediate nodes.

In EATE-RPL, nodes share their remaining energy periodically in the network with their neighbors.

$$EC_i = T_{tx} \times P_{tx} + T_{rx} \times P_{rx} + T_{cpu} \times P_{cpu} + T_{LP} \times P_{LP} \quad (2)$$

$$ER_i = 1 - \frac{EC_i}{IE_i} \quad (3)$$

4.2.3. Rank

This metric refers to the position of the nodes relative to the root. If the node rank is smaller, the node is closer to the root, and vice versa. In most cases, the rank of the root node is zero, and the rank of the other nodes increases in proportion to the distance to the root. Rank, in addition to preventing looping, is one of the most important criteria for selecting parents with optimal paths. Rank of nodes in the RPL is exchanged between nodes via DIO messages. Equation (4) shows how the rank is calculated. So that R_j is equal to the rank of node j, FR_j is the rank of parent j and RI is equal to the value of the fixed rank redundant.

$$R_j = FR_j + RI \quad (4)$$

4.2.4. Expected Transmission Count (ETX)

This criterion is known as link reliability [17]. ETX refers to the certainty of the desired connection, and it is defined in terms of the probability of successful delivery of data by the receiver (DF) and the probability of successful delivery of ACK by the sender according to equation (5). So that $ETX_{i,j}$ is the reliability of nodes i and j. Using this parameter leads to the selection of routes with higher reliability. If the ETX value is lower, the link reliability is higher, and vice versa.

$$ETX_{i,j} = \frac{1}{Df_{j,j-1} \times Dr_{j,j-1}} \quad (5)$$

4.3. The process of sending DIOs in EATE-RPL

The process of sending DIO messages in EATE-RPL is consistent with sending DIOs in the RPL protocol, except for one minor difference, which is described in detail below.

In EATE-RPL, as in the RPL protocol [14], the root node generates a DIO message when it starts operating and propagates it over the network. Sending DIOs in accordance with the RPL is repeated over specific time periods to update the network topology.

In the RPL protocol, after sending the DIO message by the root, it is shared among the nodes to the extent that all members of the DIO network receive the sent one. The process of sending DIOs in EATE-RPL follows the same rule, except that if the sender of the DIO was from a parent identified

as malicious (node trust is less than the threshold value), receiving node refuses to accept or resend the DIO message. This is done with the aim of punishing malicious agents and preventing their participation in creating the network topology. In fact, when a factor is identified as malicious, it will be excluded from network interactions for a certain period of time to punish. Therefore, no packet is received from the malicious agent during the quarantine period and no packet is sent to it.

In addition, a bit flag is provided in the DIO packet called security flag (SF) that the value of it is one or zero and is specified by the root node. If the value is one, it means the need to implement security and network nodes operate in security mode. However, if it is zero, the network application is normal without security sensitivity. In fact, depending on the application and security needs, the root node specifies the value of this flag.

4.4. Parent Selection

The EATE-RPL objective function specifies how to separate and select preferred parents. OF in EATE-RPL consists of two execution steps. The first step is to start the network, which includes discovering neighbors and creating connections. The network nodes do not know about their neighbors or their trust. At the beginning of network operation, all nodes are equal to 0.5 (average trust) and their energy is the same. Therefore, only criterion used to construct the topology is the rank of the nodes. The parent with the minimum rank value is selected as the preferred parent and the network topology is created accordingly. After creating the initial topology, the second step is called and executed. The second step is to update the network topology and build trust. In this step, if safe mode is not active (or in other words, the SF flag is set to zero), the only remaining energy, rank and ETX are as decision criteria. Based on these parameters, the preferred parents are selected and the network topology is updated. However, if the security mode is enabled, the nodes first assess the trust of parents and distinguish the parents that have the most trust threshold. Then they choose the parent with the highest priority as the preferred parent.

The parent selection mechanism based on TQAOF is presented in Algorithm (1). When a node intends to select or change its parent, EATE-RPL selects the most appropriate option from the set of candidate parents based on the parents' preference. Prioritization and selection are based on the TQAOF objective function. If the priority of two parents is equal, trust is the selection criterion and the parent with high trust is selected as the preferred parent. The parent selection mechanism based on TQAOF and trust criteria is presented in Algorithm (1). The algorithm first creates a set of candidate parents as a list. The network security conditions are checked based on the SE flag check (Line 3). If network conditions are not secure, parental priority is assessed by energy, ETX, and rank (Lines 4 and 5). Otherwise, if the network conditions are in a secure state, first the trust of the parents is measured and the untrusted parents are removed from the list of candidates (lines 7 and 8). Then the priority of the trusted parents considering the energy, ETX, and rank is assessed (lines 11 and 12). The node with the highest priority is then selected as the preferred parent (lines 14 to 16). If the priority of the two parents is equal to each other, the parent with higher trust will be selected as the preferred parent (lines 17 and 18). At the end, the preferred parent is returned as the selected parent by the algorithm (line 22).

Algorithm (1) Preferred parent selection mechanism

Input: DIO Message from nodes;
Output: Preferred parent (P_p) selection of node i;
1: TQAOF = 0;
2: **For** each j ∈ candidate fathers list
3: **If** (SF = 0) then // SF = Security Flag
4: $TQAOF_j = w_1 \times ER_j + w_2 \times \frac{1}{ETX_{i,j}} + w_3 \times \frac{1}{R_j}$;
5: $w_1 + w_2 + w_3 = 1$; // w is metrics weight
6: $w_1 + w_2 + w_3 = 1\ ,\ 0 \le w_1, w_2, w_3 \le 1$
7: **Else**
8: **If** ($T_j \le TH_T$) then
9: Discard node of parent candidate list;
10: **Else**
11: $TQAOF_j = w_1 \times ER_j + w_2 \times \frac{1}{ETX_{i,j}} + w_3 \times \frac{1}{R_j} + w_4 \times T_j$;
12: $w_1 + w_2 + w_3 + w_4 = 1\ ,\ 0 \le w_1, w_2, w_3, w_4 \le 1$
13: **End if**;
14: **If** (TQAOF < $TQAOF_j$) Then
15: $P_p = Father_j$;
16: $TQAOF = TQAOF_j$;
17: **Else if** (TQAOF = = $TQAOF_j$) Then
18: P_p= node with maximum trust;
19: **End if**;

20: **End if**;
21: **End for**
22: **Return** P_p;

4.4.1. Complementary intrusion detection system

In the EATE-RPL, in addition to the discussion of trust, a complementary intrusion detection systems (IDS) are envisaged to increase the accuracy of malicious node detection. This is because some attacks, especially intelligent attacks, may not be detected by high-reliability trust models. In other words, trust is an accumulated value based on the nodes' past behaviors and reflects an overall evaluation of the nodes. This accumulation is not sensitive enough to detect intelligent attacks, because it takes time to reduce accumulated trust. Designing an intrusion detection system based on assessing the ratio of negative behaviors and behavioral changes can be effective in counteracting these attacks.

Designing such an intrusion detection system can affect the behavior of malicious nodes, especially when they are aware of the rules of trust assessment and try to maintain a certain amount of trust value by fluctuating between their behaviors. Therefore, we use the intrusion detection system based on the evaluation of the ratio of negative behaviors with the use of the concept of entropy [59].

Methods designed based on intrusion detection systems (IDS) are based on a set of anomaly detection rules [60, 61]. As mentioned, EATE-RPL uses this system to increase the accuracy of intelligent attack detection. If IDS generates warning for a node, the node will be identified as malicious.

Since most intelligent attacks focus on behavioral changes, the proposed IDSs are based on this. In EATE-RPL, IDS warns when the ratio of negative behaviors and node behavior changes exceeds a certain threshold. Equations (6) to (9) provide details of this detection. Where $IDS_{i,j}$ is a warning symbol. $D_{i,j}$ (t-1, t) is equal to the ratio of the value of incorrect behaviors to the sum of the values of node j behaviors for node i requests in time period t-1 to t. TH_E is equivalent to the energy threshold, TH_D is the intrusion detection threshold and β is the error control index. Energy threshold and intrusion detection in terms of repetition of experiments in the proposed method are considered equal to 0.2 and 0.5, respectively. The error control index (β) is used when the node energy is in the critical state. In this case, since the negative behaviors of a node may be due to a decrease in performance, by considering the error control index, it tries to prevent the misdiagnosis of these nodes as malicious. If the change in node behavior is high during interactions, the probability of IDS warning is high, and vice versa. In fact, this IDS is designed and predicted with a focus on analyzing behavioral changes.

$$IDS_{i,j} = \begin{bmatrix} 1 \text{ If } (ER_j \geq TH_E) \text{ and } (D_{i,j}(t-1,t) > TH_D) \\ 0 \text{ If } (ER_j < TH_E) \text{ and } (D_{i,j}(t-1,t) > TH_D + \beta) \end{bmatrix} \quad (6)$$

$$D_{i,j}(t-1,t) = \frac{VNB_{i,j}(t-1,t)}{VCB_{i,j}(t-1,t) + VNB_{i,j}(t-1,t)} \quad (7)$$

$$VCB_{i,j} = \sum_{a=1}^{\text{No.of transaction}} S(a)_{i,j} \times V(a)_{i,j} \quad (8)$$

$$VNB_{i,j} = \sum_{i=1}^{\text{No.of transaction}} N(a)_{i,j} \times V(a)_{i,j} \quad (9)$$

In equation (8) and (9), $VCB_{i,j}$ and $VNB_{i,j}$are equal to the sum of the value of positive and negative interactions, respectively. No. of transaction is the sum of interactions between nods of i and j in period of time t. $S(a)_{i,j}$ and $N(a)_{i,j}$are equivalent to satisfaction and dissatisfaction with the interaction of (a) (in successful interaction $S(a)_{i,j} = 1, N(a)_{i,j} = 0$ and in case of failure $S(a)_{i,j} = 0, N(a)_{i,j} = 1$). $V(a)_{i,j}$ is equivalent to the value of interactions of (a) for nodes i and j, which will be equal to 1 and 0.5 for control and data messages, respectively. Note that packets sent in the IoT can generally be divided into data and control. Control messages are more valuable than data messages due to their important role in topology formation [47]. Considering the value of interaction in calculations, the node can't gain high trust in low-value transactions and abuse it in high-value transactions.

4.4.2. DODAG Construction and Trust Update

In EATE-RPL, trust updates are reactively and dynamic. Reactive and dynamic trust updates are based on behaviors. In this type of update, nodes are encouraged and punished for their right and wrong behaviors by increasing and decreasing trust. Details of the evaluation and update of trust were discussed earlier. Based on this assessment, if a node's trust falls below the threshold value, the faulty node is identified as malicious and added to the malicious list. In this case, the desired node will be excluded from the network exchanges for a certain period of time.

5. Simulation and experimental results

In this section, the efficiency and performance of EATE-RPL will be evaluated. For this purpose, EATE-RPL is implemented with cooja 2.7 simulator software (simulator designed based on Contiki [62]) and with protocols of RPL [14], CT-RPL [49] and SecTrust [53] has been compared. Experiments were repeated for variable of malicious agent as well as different attacks and scenarios to evaluate the performance of methods. The details of the simulation scenarios are discussed in the next section.

5.1. Simulation setup

As mentioned, open source simulation software of Contiki 2.7 / Cooja simulator was used [63]. The configured scenarios for evaluating the methods include 40 nodes of Sky mote type (TelosB) and one root node and they are located in a network with a size of 200m * 200m. The root node is in the center of the network and the other nodes are randomly placed around it. Each network node has a 16-bit microcontroller of Texas Instruments MSP430 with a frequency of 8 MHz with 10 KB of RAM and 48 KB of flash memory. To evaluate the methods, two types of attacks, black-hole and selective-forwarding, have been considered. The performance of the methods against these two types of attacks has been examined. Also, the number of malicious nodes in different scenarios is considered between 1 to 10 nodes.

The trust threshold is set to 0.5 and the γ coefficient is set to 0.5 for indefinite behaviors. The values of coefficients (w) in SE = 1 are equal to w_1, w_2, w_3, w_4 = 0.25 and in SE = 0 are equal to w_1, w_2, w_3 = 0.333. Other details of the simulation parameters are given in Table (1).

Table 1. Simulation parameters.

Parameter	Value
Simulator	Cooja-Contiki 2.7
Loss Model	Distance loss
Sensors	Skymote
Adaptation	6LoWPAN
Communication protocol	CSMA, ContikiRPL, IPv6
Traffic rate	1 packet sent every 10 seconds by every node
number of nodes	40
Network area	200m*200m
Data packet size	64 bytes
Range of nodes	RX: 50%, TX: 50m, interference: 60m
Number of attacker nodes	1-9
Transmission layer	UDP
Attacks	Blackhole, Selective forwarding
Simulation time	1 h
Radio model	Unit Disk Graph Medium (UDGM)
Trust Threshold	0.5

In experiments, the methods competed with each other despite different attacks and a variable number of malicious nodes in different scenarios. The following metrics have been used to compare the results:

1-Packet Delivery Ratio (PDR): This metric is the result of the ratio of successful received packets by the root to the total of sent packets.

2-Throughput: This metric is evaluated based on the total number of bits received during the time interval *t*.

3- Average Rank Changes (ARC): This metric provides a result of the average number of parent switches.

4-End-to-End Delay (EED): This metric is evaluated based on the average sending time of all packets received correctly by the root.

5- Average Energy Consumption (AEC): This metric presents the result of the average energy.

5.2. Result

This section contains the results of the simulations performed. Each chart is displayed with an average of 20 runs with a 95% confidence interval.

5.2.1. Packet Delivery Ratio (PDR)

Figures 1 and 2 show the PDR results under black-hole and selective forwarding attacks, respectively. The results show that in all four methods, with increasing the number of malicious nodes, PDR decreased. The reason for this is the increase in the

negative effects of the presence of malicious nodes for network exchanges and especially data loss. However, the reduction ratio for the proposed EATE-RPL was lower than for other methods. This is due to the high efficiency of EATE-RPL in supporting trusted exchanges, especially in dealing with malicious nodes, which has been more effective in scenarios with more malicious nodes. In addition to effectively detecting malicious nodes and supporting trust, the proposed method also supports the reliability of routing and exchanges, which has been another to improve successful exchanges. EATE-RPL selects parents with more trust and secure routes for sending data based on measures that it provides to evaluate nodes' trust and reliability. This performance has significant effects on improving exchanges and results the increased PDR. The effects are greater as the number of malicious nodes increases. However, despite selective forwarding attacks, EATE-RPL has provided far better results than other methods. This is due to the effective performance of EATE-RPL in counteracting deceptive behaviors. The other three methods do not provide effective measures to counter these attacks. CT-RPL has been more successful than SecTrust in assessing trust and countering attacks, resulting in better PDR performance. However, like SecTrust and RPL, this method, in addition to being vulnerable to deceptive behaviors, does not provide measures to evaluate QoS metrics but EATE-RPL also solves QoS requirements well.

Under the attack of the black-hole, when the number of malicious node was 1 node, EATE-RPL had about 94% successful delivery, which was 3.7%, 5.1% and 9.6% more successful than that of CT-RPL, SecTrust and RPL. However, in the presence of 9 malicious nodes, the successful delivery of EATE-RPL was 71%, which was 7.8%, 14.3% and 51.5% more successful than that of CT-RPL, SecTrust and RPL. But under selective forwarding attack, when the number of malicious node was 1, EATE-RPL had about 93.2% successful delivery that was 3.2%, 4% and 7.2% more successful than that of CT-RPL, SecTrust and RPL. In the presence of 9 malicious nodes, EATE-RPL successful delivery was 65.6%, which compared to CT-RPL, SecTrust and RPL was 9.7%, 15.9% and 32.5% more successful. These results explain two important points. The first one is that EATE-RPL has a stable operation with increasing number of malicious nodes. Second, the proposed method is well resistant to deceptive behaviors.

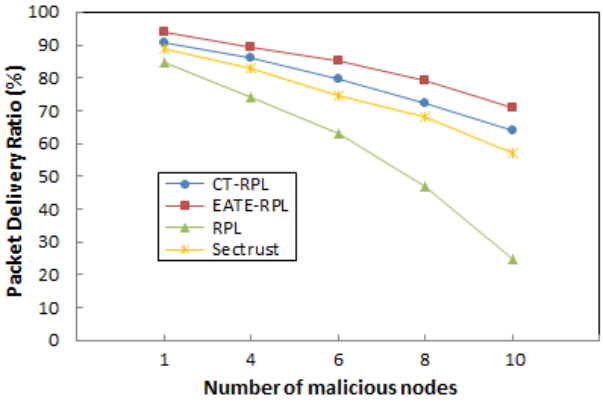

Figure 1. Packet delivery ratio under black hole attack

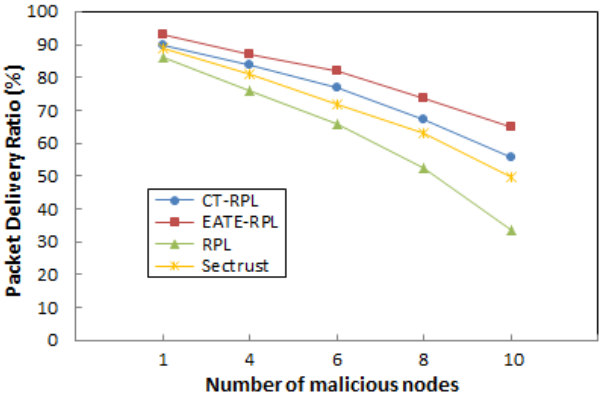

Figure 2. Packet delivery ratio under selective forwarding attack

5.2.2. Throughput

Figures 3 and 4 show the results of the throughput under black hole attacks and selective forwarding, respectively. Because increasing the number of malicious nodes leads to increased topology instability and data loss, with the increase of malicious nodes, the network throughput has decreased in both types of attacks. However, the increase in malicious nodes had less effect on the throughput during EATE-RPL operation. The result of this successful operation is the measures taken by EATE-RPL to establish trust and identify attacks. The effects of these measures on network throughput in the presence of selective forwarding attack are more obvious than other methods. RPL is extremely vulnerable to attacks and in this regard, with increasing malicious nodes, the

throughput for this method had decreased significantly. Although CT-RPL and SecTrust have been successful in building trust, these two methods are particularly inefficient and vulnerable to deceptive behaviors. The existence of this issue has caused a decline in these two methods. According to the results in the presence of black hole attack, EATE-RPL and CT-RPL had better results compared to the other two methods. EATE-RPL and CT-RPL had been more successful in detecting malicious nodes and had provided a more stable network than SecTrust and RPL. Therefore, packet loss for these two methods was reduced and network throughput was increased.

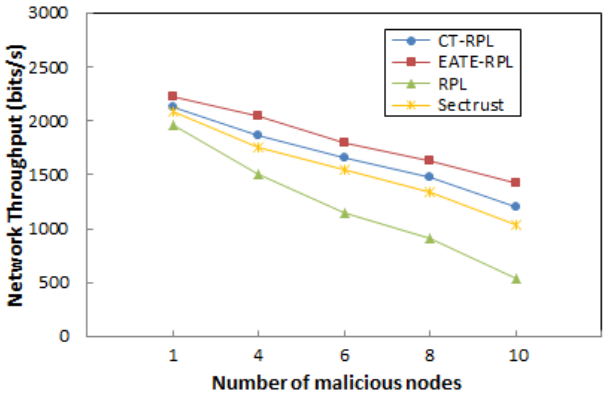

Figure 3. Network throughput under black hole attack

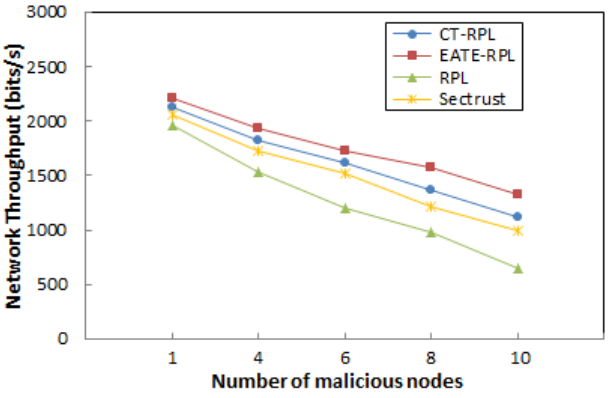

Figure 4. Network throughput under selective forwarding attack

5.2.3. Average Rank Changes (ARC)

Figures 5 and 6 show the ARC results under black hole attacks and selective forwarding, respectively. ARC provides network topology instability ratio. If the network topology is more unstable and the changes are more, the ARC is increased and vice versa. The increasing malicious factors had a direct impact on increasing ARC as it had exacerbated instability. At the same time, this increase had been more pronounced for comparable methods in the presence of selective forwarding attack. The reason for this is that due to the vulnerability of methods against selective forwarding attack, the possibility of choosing parents from malicious nodes is high. According to the results, ARC for RPL protocol is high compared to other methods that this difference is increased by increasing malicious nodes. RPL did not have a mechanism to deal with the malicious nodes, which was the main reason for the increase of ARC in this method. Note that in the selective forwarding attack, the malicious agents only for part of the sent packets have malicious behavior, so the instability for the RPL in the presence of this attack was relatively less than that of the black hole. Among other methods, Sectrust and CT-RPL performed weaker than EATE-RPL. These methods, in particular Sectrust, were less effective in detecting malicious nodes and countering attacks compared to the EATE-RPLs, and therefore ARC is increased for these protocols. To control topological instability due to malicious behaviors, nodes changed their parents frequently, leading to an increase in the rate of rank change. EATE-RPL has been more successful in reducing the negative effects of attacks and maintaining network stability in terms of measures to improve the accuracy of detecting various attacks and prevent the presence of malicious nodes. CT-RPL offers better results than Sectrust, which leads to greater stability, but this method is also sometimes vulnerable to deceptive behaviors, which has led to a slight increase in ARC.

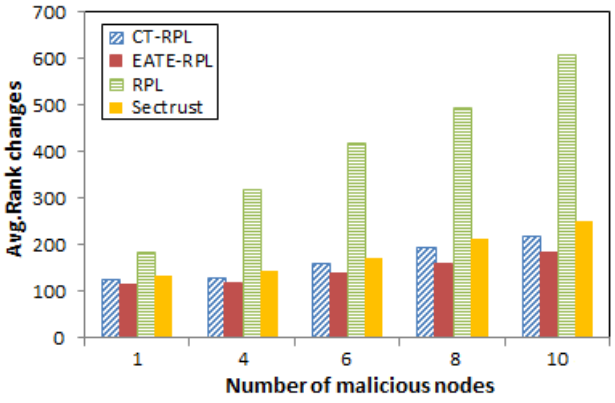

Figure 5. Average rank changes under black hole attack

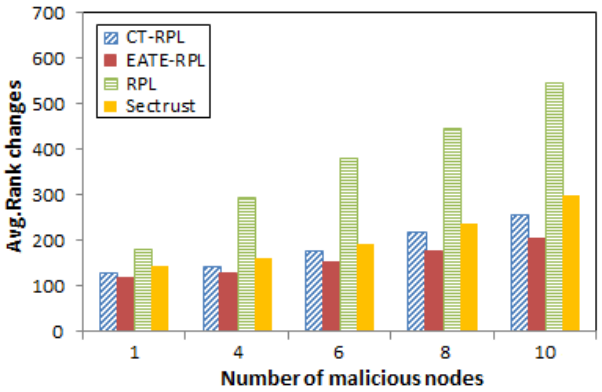

Figure 6. Average rank changes under selective forwarding attack

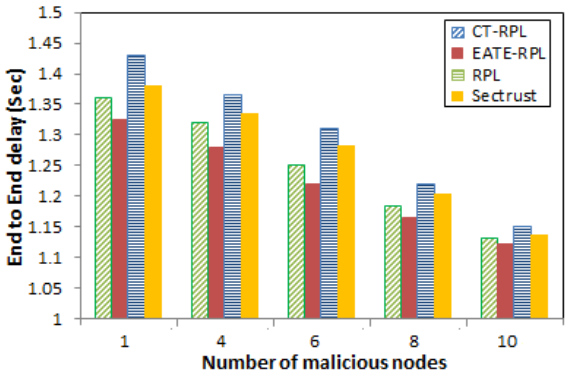

Figure 7. End-to-End delay under black hole attack

5.2.4. End to End Delay (EED)

Figures 7 and 8 show the results of end-to-end delays under black hole attacks and selective forwarding, respectively. Delay in experiments is estimated in terms of the average time to send data packets that have been successfully received by the root. Delay has generally decreased with increasing malicious nodes, because with the increase of malicious nodes, the probability of successful receptions from shorter routes is higher than longer routes, so delay is reduced. In other words, in scenarios with more malicious nodes, the probability of the presence of malicious nodes and data loss in long routes is higher, and in this regard, most data is received from shorter paths (with less delay). Since delay has been calculated for the received data successfully, it has been reduced in scenarios with more malicious nodes. In addition, the results showed that delay for EATE-RPL was lower than that of Sectrust and CT-RPL. In addition to trust, EATE-RPL considers QoS criteria for parental selection. This selection has led to improved exchanges and delay for the proposed method. In addition, EATE-RPL has been more successful in maintaining network topology stability, resulting in reduced disturbances leading to increased delay. However, in scenarios with more malicious nodes, delay for EATE-RPL is closer to other methods. Improved trusted exchanges in exchange for EATE-RPL performance have made it more likely to receive data from longer routes than other protocols in scenarios with more malicious nodes. Therefore, delay in these scenarios for the proposed method is closer to other protocols compared to the scenarios with less malicious nodes.

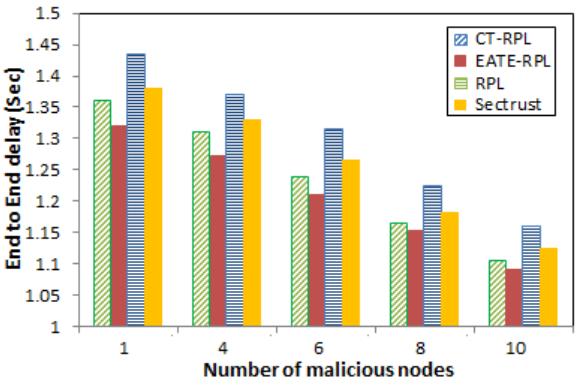

Figure 8. End-to-End delay under selective forwarding attack

5.2.5. Average Energy Consumption (AEC)

The AEC results are shown in figures 9 and 10 for the 60 and 20 minutes of simulations under selective forwarding attack, respectively. Energy consumption had increased with increasing density of malicious nodes due to increased instabilities, topological changes and rank of parental replacement. According to the results of figure 9, with increasing malicious nodes, EATE-RPL had more successful performance and less AEC increase compared to the other methods. EATE-RPL had been more successful in maintaining network topology stability in terms of identifying and preventing malicious nodes, resulting in better AEC, which is more tangible for scenarios with more malicious nodes. In RPL, the AEC has been increased significantly with the increase of malicious nodes. The lack of a mechanism to deal with malicious nodes has led to increased instabilities and topological changes resulting in an increase in AEC. CT-RPL was more successful than SecTrust in dealing with malicious nodes, and AEC was less successful in this respect.

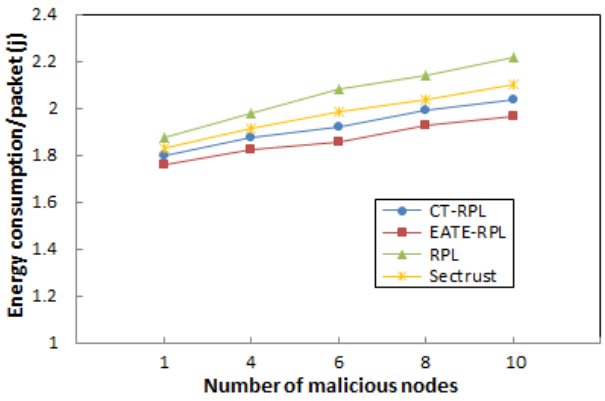

Figure 9. Average energy consumption under selective forwarding attack with simulation time 60 minutes.

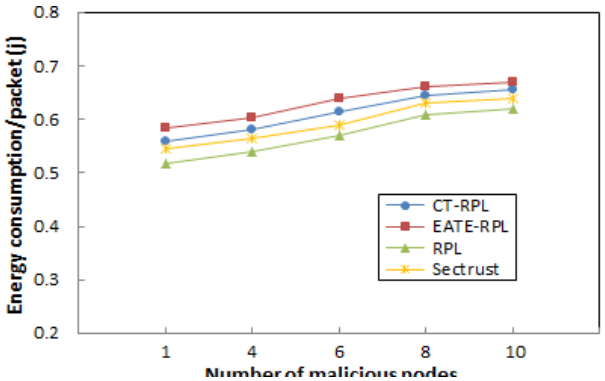

Figure 10. Average energy consumption under selective forwarding attack with simulation time 20 minutes.

According to the results of figure 10, for the simulation time of up to 20 minutes, EATE-RPL consumed more energy than other methods. One reason for this is that the EATE-RPL under attack consumes more energy to calculate and transmit DIO packets, but after identifying and preventing the malicious nodes, the topology stability is maintained and consumption is reduced. Another reason is that over time, the energy consumption of the nodes in EATE-RPL becomes more efficient and balanced. In this method, the residual energy and rank for selecting the preferred parents, respectively, lead to balance energy consumption and optimize the intermediate routes.

6. EATE-RPL Analysis

The proposed method provides the ability to support both of trust and reliability and it is an adaptive method with multi-application capability. The reason for this is the reliability metric review, in addition to the trust in selecting preferred parents, the SE flag prediction to determine security needs, the adaptive evaluation of trust, and the thresholds for trust and intrusion detection.

In EATE-RPL, nodes can meet different needs in different applications due to the SF flag and the proportional value to the weights (w_i). According to security needs, variable rigor of trust can be applied by giving proportional value to trust threshold and intrusion detection. Accordingly, depending on the conditions and environment of the IoT network, an effective trade-off between the effectiveness of trust and reliability can be provided, and on the other hand, a proportional stricter of trust can be applied. For example, in normal applications, the routing and interaction process can only be done by focusing on QoS metrics (rank, residual energy, and ETX). For another example, trust and reliability are supported at the same time as starting the network by giving equal weight to w_i, or by increasing the value of w_i, the value of trust or reliability in decisions is increased. In another scenario, if energy has high importance for an application, routing can be done by focusing on energy by increasing the weight of this parameter. It is also possible to increase the rigor of trust and accuracy in detecting malicious nodes by giving appropriate value to the trust and intrusion detection thresholds accordingly. There is no set value for trust and intrusion detection thresholds, and depending on the application and security needs it is defined. It is worth noting that increasing the trust threshold and decreasing the intrusion detection threshold can be more effective in identifying malicious nodes. But it also increases error, which may mistakenly detect some normal nodes as malicious. Such an issue will cause topological instability, decrease network efficiency and increase network energy consumption. On the other hand, if the trust threshold is low and the intrusion detection threshold is high, the detection accuracy will be reduced and the detection of malicious nodes will take longer.

An important issue and limitation of EATE-RPL include the storage space required for calculations and trust. As mentioned, nodes in the EATE-RPL store and maintain a list of their neighbors in which the nodes' trust and related updates are stored. This storage, however, does not create high overhead for small networks and requires little memory

consumption, but when the network is scalable and implemented over a large geographical area, it requires a lot of memory and heavy overheads is imposed on the network. it is noteworthy that in the proposed method, trust assessment is limited to direct trust and recommendations are not used. In this respect, EATE-RPL performs better than other trust models, which is one of the important features of EATE-RPL. However, storing and updating trust requires high memory consumption. To overcome this limitation, measures should be envisaged that, while establishing effective trust, make the best use of network resources. In future work, focus on improving EATE-RPL by focusing on applying the proposed method to scalable networks.

7. Conclusion and future work

Many studies have been done to ensure trust in routing and IoT data exchanges and this issue is an important tool for identifying malicious nodes and ensuring the accuracy of network performance. However, establishing effective trust is a very complex issue because the nodes' trust is determined based on their behavior. This way of assessing trust is not enough due to the widespread attacks of malicious nodes, especially in the face of deceptive attacks. To improve this issue, this paper presents an improved protocol called dual data-communication trust mechanism for RPL (EATE-RPL). EATE-RPL focuses on creating secure and reliable routing topology and detecting malicious agents with high accuracy. For this purpose, a new objective function has been introduced to select the preferred parents, taking into account the trust and reliability of the nodes and the routing topology is created based on this function. The objective function of the proposed method is created in such a way that the selection of parents from nodes will be done with the most trust and reliability. Data exchanges are then initiated through the network communication graph, and trust models combined with intrusion detection system are used to detect malicious nodes. This design increases the accuracy of assessing and detecting malicious nodes. The results of EATE-RPL simulation using Cooja in different scenarios indicate the high efficiency of the proposed method in detecting malicious nodes, improving trust and other influential metrics of reliable exchanges compared to previous researches. In future work, an attempt has been made to improve the efficiency of EATE-RPL for use in mobile applications by developing a proposed method considering the dynamics of nodes.

References

[1] Hassan, Rondik J., et al. "State of art survey for iot effects on smart city technology: challenges, opportunities, and solutions." *Asian Journal of Research in Computer Science* 22 (2021): 32-48.

[2] Hassan, Rosilah, et al. "Internet of Things and its applications: A comprehensive survey." *Symmetry* 12.10 (2020): 1674.

[3] Khanna, Abhishek, and Sanmeet Kaur. "Internet of things (IoT), applications and challenges: a comprehensive review." *Wireless Personal Communications* 114.2 (2020): 1687-1762.

[4] Al-Emran, Mostafa, Sohail Iqbal Malik, and Mohammed N. Al-Kabi. "A survey of Internet of Things (IoT) in education: Opportunities and challenges." *Toward social internet of things (SIoT): enabling technologies, architectures and applications* (2020): 197-209.

[5] Abiodun, Oludare Isaac, et al. "A review on the security of the internet of things: challenges and solutions." *Wireless Personal Communications* 119.3 (2021): 2603-2637.

[6] Muzammal, Syeda M., Raja Kumar Murugesan, and N. Z. Jhanjhi. "A comprehensive review on secure routing in internet of things: Mitigation methods and trust-based approaches." *IEEE Internet of Things Journal* 8.6 (2020): 4186-4210.

[7] Sharma, Avani, et al. "Towards trustworthy Internet of Things: A survey on Trust Management applications and schemes." *Computer Communications* 160 (2020): 475-493.

[8] Chahal, Rajanpreet Kaur, Neeraj Kumar, and Shalini Batra. "Trust management in social Internet of Things: A taxonomy, open issues, and challenges." *Computer Communications* 150 (2020): 13-46.

[9] Pourghebleh, Behrouz, Karzan Wakil, and Nima Jafari Navimipour. "A comprehensive study on the trust management techniques in the Internet of Things." *IEEE Internet of Things Journal* 6.6 (2019): 9326-9337.

[10] Solapure, Sharwari S., and Harish H. Kenchannavar. "Design and analysis of RPL objective functions using variant routing metrics for IoT applications." *Wireless Networks* 26 (2020): 4637-4656.

[11] Lamaazi, Hanane, and Nabil Benamar. "A comprehensive survey on enhancements and limitations of the RPL protocol: A focus on the objective function." *Ad Hoc Networks* 96 (2020): 102001.

[12] Dey, Amlan Jyoti, and Hiren Kumar Deva Sarma. "Routing techniques in internet of things: A review."

Trends in Communication, Cloud, and Big Data (2020): 41-50.

[13] Marietta, J., and B. Chandra Mohan. "A review on routing in internet of things." *Wireless Personal Communications* 111.1 (2020): 209-233.

[14] Gaddour, Olfa, and Anis Koubâa. "RPL in a nutshell: A survey." *Computer Networks* 56.14 (2012): 3163-3178.

[15] Kushalnagar, Nandakishore, Gabriel Montenegro, and Christian Schumacher. "IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals." (2007): 1-11.

[16] Simoglou, George, et al. "Intrusion detection systems for RPL security: a comparative analysis." *Computers & Security* (2021): 102219.

[17] Almusaylim, Zahrah A., Abdulaziz Alhumam, and N. Z. Jhanjhi. "Proposing a secure RPL based internet of things routing protocol: a review." *Ad Hoc Networks* 101 (2020): 102096.

[18] Raoof, Ahmed, Ashraf Matrawy, and Chung-Horng Lung. "Routing attacks and mitigation methods for RPL-based Internet of Things." *IEEE Communications Surveys & Tutorials* 21.2 (2018): 1582-1606.

[19] Verma, Abhishek, and Virender Ranga. "Security of RPL based 6LoWPAN Networks in the Internet of Things: A Review." *IEEE Sensors Journal* 20.11 (2020): 5666-5690.

[20] Khraisat, Ansam, and Ammar Alazab. "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges." *Cybersecurity* 4.1 (2021): 1-27.

[21] Hajiheidari, Somayye, et al. "Intrusion detection systems in the Internet of things: A comprehensive investigation." *Computer Networks* 160 (2019): 165-191.

[22] Arshad, Junaid, et al. "A review of performance, energy and privacy of intrusion detection systems for IoT." *Electronics* 9.4 (2020): 629.

[23] Seyfollahi, Ali, and Ali Ghaffari. "A review of intrusion detection systems in RPL routing protocol based on machine learning for internet of things applications." *Wireless Communications and Mobile Computing* 2021 (2021).

[24] Napiah, Mohamad Nazrin, et al. "Compression header analyzer intrusion detection system (CHA-IDS) for 6LoWPAN communication protocol." *IEEE Access* 6 (2018): 16623-16638.

[25] Smys, S., Abul Basar, and Haoxiang Wang. "Hybrid intrusion detection system for internet of things (IoT)." *Journal of ISMAC* 2.04 (2020): 190-199.

[26] Agiollo, Andrea, et al. "DETONAR: Detection of routing attacks in RPL-based IoT." *IEEE Transactions on Network and Service Management* 18.2 (2021): 1178-1190.

[27] Pasikhani, Aryan M., et al. "Intrusion Detection Systems in RPL-based 6LoWPAN: A Systematic Literature Review." *IEEE Sensors Journal* (2021).

[28] Boyanapalli, Arathi, and A. Shanthini. "A Comparative Study of Techniques, Datasets and Performances for Intrusion Detection Systems in IoT." *Artificial Intelligence Techniques for Advanced Computing Applications*, 2021. 225-236.

[29] Arış, Ahmet, Sıddıka Berna Örs Yalçın, and Sema F. Oktuğ. "New lightweight mitigation techniques for RPL version number attacks." *Ad Hoc Networks* 85 (2019): 81-91.

[30] Ioulianou, Philokypros, et al. "A signature-based intrusion detection system for the internet of things." *Information and Communication Technology Form* (2018).

[31] Kfoury, Elie, et al. "A self organizing map intrusion detection system for rpl protocol attacks." *International Journal of Interdisciplinary Telecommunications and Networking (IJITN)* 11.1 (2019): 30-43.

[32] Luangoudom, Sonxay, et al. "svBLOCK: mitigating black hole attack in low-power and lossy networks." *International Journal of Sensor Networks* 32.2 (2020): 77-86.

[33] Soni, Gaurav, and R. Sudhakar. "A L-IDS against dropping attack to secure and improve RPL performance in WSN aided IoT." *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 2020.

[34] Mayzaud, Anthéa, et al. "Using the RPL protocol for supporting passive monitoring in the Internet of Things." *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2016.

[35] Mayzaud, Anthéa, Rémi Badonnel, and Isabelle Chrisment. "Detecting version number attacks in RPL-based networks using a distributed monitoring architecture." *2016 12th International Conference on Network and Service Management (CNSM)*. IEEE, 2016.

[36] Yavuz, Furkan Yusuf, Ü. N. A. L. Devrim, and G. Ü. L. Ensar. "Deep learning for detection of routing attacks in the internet of things." *International Journal of Computational Intelligence Systems* 12.1 (2018): 39-58.

[37] Li, Yuxi, et al. "Deep learning in security of internet of things." *IEEE Internet of Things Journal* (2021).

[38] Thamilarasu, Geethapriya, and Shiven Chawla. "Towards deep-learning-driven intrusion detection for the internet of things." *Sensors* 19.9 (2019): 1977.

[39] Junior, Franklin Magalhães Ribeiro, and Carlos Alberto Kamienski. "A Survey on Trustworthiness for the Internet of Things." *IEEE Access* 9 (2021): 42493-42514.

[40] Mohammadi, Venus, et al. "Trust-based recommendation systems in Internet of Things: a systematic literature review." *Human-centric Computing and Information Sciences* 9.1 (2019): 1-61.

[41] Kuseh, Simon Wewoliamo, et al. "A Survey of Trust Management Schemes for Social Internet of Things." *Inform* 7.1 (2022).

[42] Kumar, Rajesh, and Rewa Sharma. "Leveraging blockchain for ensuring trust in IoT: A survey." *Journal of King Saud University-Computer and Information Sciences* (2021).

[43] Karkazis, Panagiotis, et al. "Evaluation of RPL with a transmission count-efficient and trust-aware routing metric." *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014.

[44] Sankar, S., et al. "Trust-Aware Routing Framework for Internet of Things." *International Journal of Knowledge and Systems Science (IJKSS)* 12.1 (2021): 48-59.

[45] Hashemi, Seyyed Yasser, and Fereidoon Shams Aliee. "Fuzzy, Dynamic and Trust Based Routing Protocol for IoT." *Journal of Network & Systems Management* 28.4 (2020).

[46] Djedjig, Nabil, et al. "New trust metric for the RPL routing protocol." *2017 8th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2017.

[47] Djedjig, Nabil, et al. "Trust-aware and cooperative routing protocol for IoT security." *Journal of Information Security and Applications* 52 (2020): 102467.

[48] Muzammal, Syeda Mariam, Raja Kumar Murugesan, Noor Zaman Jhanjhi, and Low Tang Jung. "SMTrust: proposing trust-based secure routing protocol for RPL attacks for IoT applications." In *2020 International Conference on Computational Intelligence (ICCI)*, pp. 305-310. IEEE, 2020.

[49] ul Hassan, Temur, et al. "CTrust-RPL: A control layer-based trust mechanism for supporting secure routing in routing protocol for low power and lossy networks-based Internet of Things applications." *Transactions on Emerging Telecommunications Technologies* 32.3 (2021): e4224.

[50] Patel, Anshuman, and Devesh Jinwala. "A Trust-Integrated RPL Protocol to Detect Blackhole Attack in Internet of Things." *International Journal of Information Security and Privacy (IJISP)* 15.4 (2021): 1-17.

[51] Hashemi, Seyyed Yasser, and Fereidoon Shams Aliee. "Dynamic and comprehensive trust model for IoT and its integration into RPL." *The Journal of Supercomputing* 75.7 (2019): 3555-3584.

[52] Zangeneh, Saeid, and Rassoul Roustaei. "A Novel Approach for Protecting RPL Routing Protocol against Blackhole Attacks in IoT Networks." (2021).

[53] Airehrour, David, Jairo A. Gutierrez, and Sayan Kumar Ray. "SecTrust-RPL: A secure trust-aware RPL routing protocol for Internet of Things." *Future Generation Computer Systems* 93 (2019): 860-876.

[54] Airehrour, David, Jairo Gutierrez, and Sayan Kumar Ray. "Securing RPL routing protocol from blackhole attacks using a trust-based mechanism." *2016 26th International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, 2016.

[55] Prathapchandran, K., and T. Janani. "A Trust-Based Security Model to Detect Misbehaving Nodes in Internet of Things (IoT) Environment using Logistic Regression." *Journal of Physics: Conference Series*. Vol. 1850. No. 1. IOP Publishing, 2021.

[56] Thubert, Pascal. "Objective function zero for the routing protocol for low-power and lossy networks (RPL)." (2012).

[57] Gnawali, Omprakash, and Philip Levis. "The minimum rank with hysteresis objective function." *RFC 6719* (2012).

[58] Amirinasab Nasab, Mehdi, et al. "Energy-efficient method for wireless sensor networks low-power radio operation in internet of things." *Electronics* 9.2 (2020): 320.

[59] Weng, Jianshu, Chunyan Miao, and Angela Goh. "An entropy-based approach to protecting rating systems from unfair testimonies." *IEICE TRANSACTIONS on Information and Systems* 89.9 (2006): 2502-2511.

[60] Leloglu, Engin. "A review of security concerns in Internet of Things." *Journal of Computer and Communications* 5.1 (2016): 121-136.

[61] Abbas, Adeel, et al. "A new ensemble-based intrusion detection system for internet of things." *Arabian Journal for Science and Engineering* 47.2 (2022): 1805-1819.

[62] Dunkels, Adam, Bjorn Gronvall, and Thiemo Voigt. "Contiki-a lightweight and flexible operating system for tiny networked sensors." *29th annual IEEE international conference on local computer networks*. IEEE, 2004.

[63] Tsiftes, Nicolas, Joakim Eriksson, and Adam Dunkels. "Low-power wireless IPv6 routing with ContikiRPL." *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. 2010.

Journal of Optimization of Soft Computing (JOSC)

Vol. 1, Issue 2, pp: (18-25), December-2023

Journal homepage: https://sanad.iau.ir/en/Journal/josc

Research paper

Improved analysis of LUG file-related bulk data using LLG

Azin Pishdad[1], Babak Nikmard[1*], Golnaz Aghaee Ghazvini[1] and Mehrdad Abbasi[1]

1. Department of Computer Engineering- Dolatabad Branch, Islamic Azad University, Dolatabad, Iran

Article Info

Article History:
Received: 2024/1/15
Revised:
Accepted: 2024/2/28

Keywords:
neural network, generative artificial intelligence, large language model, LLM, Log File.

**Corresponding Author's Email Address: a.pishdad@iauda.ac.ir*

Abstract

Nowadays, organizations generate a significant volume of log files that require processing for condition checking, debugging, and anomaly resolution. Outsourcing such processing is not suitable due to the need for real-time processing and security maintenance. Given the multitude of different software and services, organizations face a substantial volume of production logs that should be processed rather than deleted or ignored. In the traditional approach, experts manually check the logs daily. This, on one hand, slows down the process, increases the time and inaccuracy, and, on the other hand, results in a high hiring cost due to the need for an expert force. This article introduces a solution that employs generative neural networks to establish a local structure for log analysis within the organization. The process involves retrieving and parsing text files from various sectors, segmenting them into manageable portions, embedding them, and storing them in a vector database. In this structure, a trained individual without special expertise can quickly access necessary information using appropriate prompts from a local language model available at any time. Therefore, the proposed method can increase the stability of security, increase the speed of analysis, and reduce the costs of human resources.

1. Introduction

As the digital age continues to grant unprecedented access to information, the demand for efficient methods to navigate, search, and extract pertinent data from logs has grown substantially. Organizations generate an enormous volume of logs daily, necessitating thorough analysis. Initially, these files are organized and dispatched to designated individuals responsible for data mining. These individuals are tasked with continuous monitoring of the logs to detect anomalies or suspicious activities. It is at this juncture that the imperative for an automated and streamlined log analysis system becomes evident, rendering the previously employed manual search and interpretation methods obsolete in light of the escalating log volumes.

Typically, an organization's logs are of great value, with a paramount focus on security due to the presence of sensitive information. Consequently, outsourcing such unregulated resources poses the risk of data leakage. The establishment of an in-house system for log analysis and evaluation emerges as the preferred option. This issue finds resolution through the application of artificial intelligence and machine learning tools. In the context of neural networks, intelligent systems, trained with specific parameters, exhibit the capability to comprehend, recover, and furnish meaningful analyses based on a given set of logs.

In a broader sense, the creation of an internal structure using large language models powered by neural networks enables effective data management and user query responses.

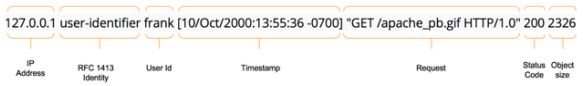

Figure 1- Anatomy of a log file

1.1. Log Management

Logs encompass a collection of data derived from the activities and performance of programs, systems, and users, curated for the purpose of identifying suspicious events. In many business settings, these valuable resources manifest in various formats, including structured, semi-structured, and unstructured files [1]. Log files typically contain sensitive information, and the removal of these records from an organization's internal system can jeopardize data security. The constituents of a log file, akin to its anatomy like figure 1, comprise critical details such as user IDs, system IDs, executed operations with timestamps, information events, and other relevant data [2].

Logs come in various types, including event logs, server logs, access logs, multi-user system logs, resource logs, security logs, and threat logs. These log types are categorized at different levels depending on the ecosystem, such as emergency, warning, informational, and debugging [3]. The management and analysis of logs are instrumental in monitoring the overall performance of programs and tools. By detecting adverse effects and damages, this process effectively distinguishes unexpected and potentially harmful activities, such as errors and intrusions, from routine procedures like the initiation and termination of processes. Additionally, it provides valuable insights to the organization. Handling extensive log data poses a considerable challenge for organizations seeking to monitor their infrastructure for security reasons while maintaining localized control. Various software and systems, such as MongoDB, Kafka, as well as databases, big data, cloud systems, and syslog computing systems, can receive logs. Given the impracticality of outsourcing log management and the unique internal conditions of each organization, the workload of the IT department has increased, necessitating the development and integration of automation tools. Many organizations store their reports within the Security Information and Event Management (SIEM) department. However, as log volumes escalate, this approach becomes problematic, prompting the need for Centralized Log Management (CLM). In a CLM system, data is integrated and directed to a data highway, where it undergoes filtration before being delivered to the intended destination [3]. The implementation of such systems can be facilitated through pipeline creation, ensuring compatibility with large language models and integration into existing infrastructure.

1.2. Analysis of Logs

When dealing with unstructured log files, preprocessing becomes a necessity. To address this, the research conducted by [1] suggests three primary approaches: first: Tokenization: This straightforward method involves segmenting log messages into tokens. It doesn't require a parser and lacks the ability to provide semantic interpretations of symbols. second: Parsing Messages: Parsing involves extracting information from sets of events, encompassing sequences, counts, or statistics. Third: Parameter Extraction: This approach focuses on retrieving parameters, including timestamps, from parsed events.

The analysis process entails the examination and interpretation of logs generated by network systems, operating systems, applications, servers, hardware, and software components. This process provides visibility into the performance and health of the IT infrastructure and application stacks, allowing the detection of potential issues, including security vulnerabilities and failures. It also aids in identifying and preventing problems like overprovisioning and underprovisioning. These techniques are applied to segment documents into manageable chunks, embedding them, and storing them in a vector database to facilitate feeding the document data into a large language model.

Historically, log analysis was performed manually, with experts interpreting log files using text processing software like sedawk. However, the proliferation of logs generated by modern programs has rendered manual analysis infeasible, necessitating automated mechanisms [4]. Hand-coded rules that search for specific keywords have limited utility and are ill-suited for scenarios with unknown parameters [5].

Consequently, the need to employ alternative methods developed by deep learning models has emerged. These methods include pattern recognition using machine learning, keyword tagging, classification, and correlation techniques. Notable examples include the semantic multiphase matching algorithm based on a vector space model and natural language processing (NLP) for matching models [6], as well as the technique for discovering a set of related activities based on k-means clustering in [7] and [8]. Time series analysis is used to detect changes in event frequency [9]. The field of NLP also finds application, such as in linguistic reconstruction [10], where refactoring technologies used in

software engineering are applied for syntactic, semantic, and functional reconstruction. Various tools, such as Logstash, ELK Stack, and Sentry, prove helpful.

One of the most significant advantages of log analysis is anomaly detection [1], which involves reporting deviations from normal system behavior by creating models. These approaches may use unsupervised learning, which does not necessitate labeled logs, or supervised learning. Anomaly detection is categorized into two main types: offline and online. In offline anomaly detection systems, notable examples include PCA [11], which utilizes principal component analysis to detect anomalies, and LogCluster [12], which assigns weights to events and employs cumulative hierarchical clustering to identify patterns. Invariants Mining [13] analyzes single values in reports to learn variables, while LogRobust [14] utilizes a Bi-LSTM neural network for learning. Online anomaly detection systems employ supervised learning, using patterns learned offline from normal logs to detect anomalous events in production. Prominent examples include DeepLog [15] and LogAnomaly [16].

This article delves into issues and their resolutions within the realm of daily log analysis, organized into seven sections: The initial section provides an introduction and discusses the current conditions. The second section outlines the artificial intelligence process. Moving forward, the third section addresses the challenges that require solutions. Subsequently, the fourth and fifth sections delineate the proposed solution and its evaluation. The sixth section establishes limitations and outlines the necessary requirements. Finally, the seventh section conclussion the article.

2. Generative Artificial Intelligence

Generative artificial intelligence (GAI) aims to develop machines capable of reasoning and acting in a manner resembling human cognition [17]. This technology empowers AI systems to generate text, images, or other media items in response to user requests, revolutionizing various industries. GAI models acquire an understanding of patterns and structures within their training data and subsequently generate new data. These models can be trained on extensive datasets, enabling multi-stage learning and domain knowledge development through reinforcement learning from human feedback. GAI can be categorized as either unimodal or multimodal. Unimodal systems process a single type of input, while multimodal systems can handle multiple input modalities.

GAI can further be classified into two main categories: task-specific GAI and general GAI. The focus of this research is on investigating general GAI [18]. Advances in big data representation technologies have led to the emergence of a human-interpretable language for patterns and structures within input data, enabling the accomplishment of diverse objectives across various environments. The objective is to transcend the current language generation paradigm, which often involves fitting sample distributions to specific tasks. However, the development of general GAI is still confronted by several key challenges. These challenges include high training and maintenance costs, dispersion of high-quality data, integration of domain knowledge, interpretability, model validity, resource allocation, and security. A comprehensive review of the development of GAI from 2018 to the present is presented in Figure 2 [18], illustrating the evolution of this field.

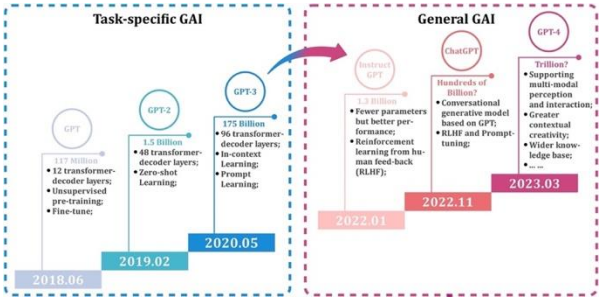

Figure 2. Evolution of GAI [18]

2.1. The Evolution of Generative AI Models

Initially, GPT (Generative Pre-trained Transformer) demonstrated its potential to generate specific natural language through unsupervised pre-training and fine-tuning in downstream tasks. It utilizes 12 transformer decoder layers to perform next-word prediction and generate coherent text. The subsequent model, GPT-2 [19], extended this structure and introduced the concept of zero-shot learning.

Building upon these advances, GPT-3 [20] introduced the use of prompts to reduce the reliance on supervised datasets. It leverages the prior knowledge acquired during the pre-training phase to enhance the quality of generated content. This approach enables the language model to adapt quickly to new scenarios, even with limited or unlabeled data. Additionally, the emergence of reinforcement learning became notable during this period.

InstructGPT, introduced as a general GAI, utilizes Reinforcement Learning from Human Feedback (RLHF) to make decisions aligned with human

preferences, achieving improved performance with fewer parameters. This results in an intelligent system characterized by consistency, interpretability, and reliability. RLHF optimizes the original problem by transforming indistinguishable objectives in language production tasks into sequential decision processes [21].

In November 2022, ChatGPT, a general-purpose OpenAI-based chatbot known for its ability to "produce human-like text," attracted millions of users [22]. ChatGPT, built on the foundations of large language models, serves as a conversational user interface and application programming interface (API). It empowers AI applications to generate text and images [17]. These large language models, equipped with self-supervised learning capabilities, exhibit exceptional proficiency in complex tasks and question answering, even in the absence of intrinsic motivations or goals. Beyond language mastery, transformer-based Large Language Models (LLMs), like those in ChatGPT, can tackle diverse and challenging tasks, including mathematics, coding, medical analysis, law, psychology, and more. These models learn from large text datasets through unsupervised learning methods, allowing them to recognize statistical patterns and regularities in human language, thereby generating coherent and contextually relevant responses to user inputs [23]. ChatGPT relies on a self-generated LLM, a machine learning system capable of independent learning from data to produce sophisticated and seemingly intelligent text after extensive training on large text datasets [24].To further enhance the model's capabilities, reinforcement learning from human feedback is integrated with GAI models, enabling domain knowledge to play a role in the training process. This integration marks a shift from "fitting-generation" to "Pretraining-Prompting-Generation" [25].

Recently, GPT-4, the latest model developed by OpenAI, has been trained on an unprecedented scale of computation and data, delivering remarkably human-like performance across a wide array of tasks. The introduction of GPT-4 signifies the evolution of general GAI based on the GPT series, which now includes the understanding and generation of multimodal data, encompassing text, images, and audio. Moreover, it can identify logical errors in reasoning to generate valuable responses [26].

3. Challenges in Log Analysis

The purpose of this article is to investigate three critical aspects: security, time efficiency, and cost reduction in the analysis of logs received from various departments within an organization.

Security:

Logs contain intricate details of valuable and private information pertaining to the organizational structure. Careful examination of this data is crucial, and outsourcing beyond the organization's internal network is not considered. The proposed method establishes a local structure within the organization to uphold the security of such sensitive information.

Time Efficiency:

The sheer volume of logs generated and stored for recovery and analysis is extensive. Traditional methods of checking logs involve writing various queries to extract useful information and reports, which is time-consuming. This results in a slowed-down log checking process, making it nearly impossible to review all logs. The proposed method streamlines and accelerates this process, providing a solution to the time constraints associated with log analysis.

Cost Reduction:

Log analysis typically requires the expertise of specialized individuals, and the annual human resources costs escalate when hiring such personnel. Furthermore, the nature of the job demands more than one person, leading to increased salary expenses. In contrast, the proposed method operates efficiently by reducing employment costs. It achieves this by utilizing a trained professional, eliminating the need for high-level experts and minimizing financial burdens.

4. Proposed Method: Development of report analysis using local LLMs

Development of Report Analysis using Local Large Language Models (LLMs): There exists a diverse array of AI applications, primarily relying on machine learning, deep learning, and natural language processing (NLP). Machine learning harnesses advanced algorithms to analyze data and identify patterns for predictive purposes. Deep learning employs neural networks to process vast datasets. Large language models (LLMs) [20], [32], [33], considered a groundbreaking technology in the realm of natural language processing, empower developers to create previously unattainable applications, such as the Copilot programming assistant [34]. However, the true potential of these LLMs is realized when they are integrated with other computational or knowledge resources. This amalgamation can result in the creation of customized solutions for

individual organizations. With appropriate training, these integrated systems can ingest all available logs and provide meaningful analyses. For instance, organizations like Google have introduced competitive artificial intelligence chatbots like ChatGPT and other advanced models proposed by OpenAI. Google's AI chatbot, known as "Bard," showcases impressive conversational skills. Meanwhile, several Chinese companies, including HUAWEI, Baidu, Alibaba, and Tencent, have offered their LLMs to develop industrial-grade models, contributing to the industrialization of large-scale artificial intelligence models.

To respond to user inquiries regarding the source of collected logs, it's essential to train LLMs. While LLMs possess remarkable capabilities, they lack knowledge beyond what they've been trained on. "Retrieval Augmented Generation" is a technique that enables the creation of intelligent systems connecting a language model to additional data sources to provide instructions to the LLM, enabling it to manage and interact with its environment.

In this proposed method, as illustrated in Figure 3, the log files undergo initial loading and are subsequently divided into smaller, manageable segments using a text splitter to ensure compatibility with the model. These logs are then embedded for storage. Utilizing these embeddings, semantically similar logs to a given query can be retrieved. These snippets are indexed in a database to facilitate future searches and retrievals. This created index can be employed for generation, essentially functioning as a search format that operates in two stages: capturing the logs in a query format, followed by retrieving the appended production chain. A vector database is integrated to construct high-performance vector search programs, enhancing the speed and accuracy of search and retrieval processes. Various index types are available, with Vectorstore being the most common. Its user-friendly API, scalability, and advanced algorithms allow developers to efficiently manage extensive vector data, achieving real-time retrieval and building effective search engines.

When a user submits a query, the query is initially embedded, and a similarity search is conducted in the vector database. The retrieved documents, in conjunction with the query, serve as input to the query chain. This consolidated input is then forwarded to the LLM, enabling it to generate a response that precisely addresses the user's inquiry. The search encompasses the logs relevant to the question and employs both a PromptValue, which acts as model input, and a PromptTemplate, responsible for creating these inputs. PromptValue is conveyed to the model, and the outcome, representing an accurate and well-informed response based on the information found in the pertinent documents, is delivered in a natural language format comprehensible to all stakeholders. These search results feed into the text understanding for the Large Language Model (LLM), enabling it to provide an accurate response to the user's query from the extensive log collection.

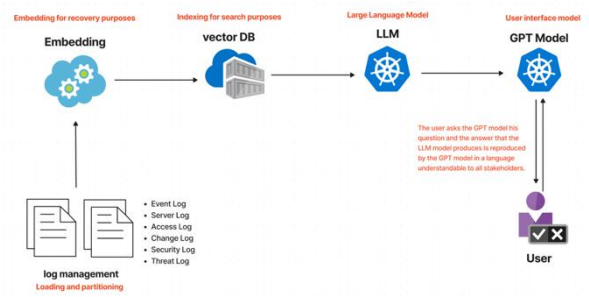

Figure 3. Report analysis using local LLMs

5. Assessment

To provide a comprehensive overview of the utility of this model, along with its notable advantages, which include exceptional text comprehension, proficiency in handling unstructured data and complex patterns, continuous learning capabilities, and the generation of natural language output for improved comprehension among non-technical stakeholders, three primary advantages emerge:

First: Enhanced Log Security

One of the key benefits lies in log security. This model is implemented locally, eliminating the need to transmit logs outside the organization or outsource their analysis. This inherent security feature ensures that sensitive log data remains within the organization's control.

Second: Time Efficiency

The model offers a substantial reduction in the time required to extract necessary information from logs. In traditional log analysis, composing several query lines demands time for contemplation and typing. However, with the assistance of the large language model, formulating a question in natural language requires significantly less time. To quantify this time-saving advantage, we considered 24 distinct non-repetitive questions to be written as queries. The time taken for each query was compared with the time it would take a person to naturally pose the same questions to the model. Initially, users working with the model may spend slightly more time formulating questions. However, over time, as users gain experience,

Chart 1 depicts a noticeable trend toward reduced query time. This assessment underscores the model's capacity to enhance both log security and operational efficiency, making it a valuable asset for organizations seeking to streamline their log management and analysis processes.

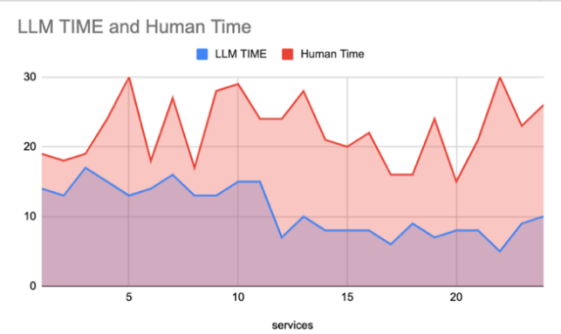

Chart 1. Comparison of time consumption in each method

Third: Cost Reduction

The second advantage discussed relates to the reduction in costs. Traditionally, writing a suitable query for log analysis necessitates the presence of an expert with relevant knowledge within the organization, responsible for continuous analysis and necessary checks. However, with the language model, anyone can pose a question without the need for comprehensive expertise, and receive a suitable answer. This capability contributes to cost savings by obviating the need to hire human resources for log analysis.

In Chart 2, presented as a cumulative frequency chart, a cost comparison is depicted. The chart illustrates the requirements, implementation, and maintenance costs over two years for both the traditional model and the language model in a private company. Initially, the language model incurs higher implementation costs, but as time progresses, only maintenance costs are incurred, demonstrating a declining trend. Conversely, in the traditional model, the cost over two years, primarily related to hiring an expert, is nearly twice that of the language model. This third advantage highlights the substantial cost savings achievable by adopting the language model for log analysis, offering a more cost-effective and accessible approach to organizations.

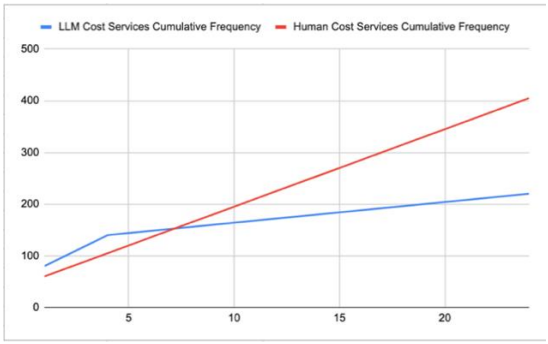

Chart 2. Cost comparison in each method in the form of cumulative frequency

6. Limitations

Despite significant progress and advancements in the field of log analysis with the aid of artificial intelligence, several challenges persist, offering opportunities for future research and the expansion of these methods. The following are notable challenges, along with potential solutions:

Generalization Ability: Achieving a valid output, in addition to log data, necessitates the inclusion of more information for model training. This expanded dataset increases the workload of the ensemble, thereby demanding additional research efforts.

Interpretability: Enhancing methods and models that make the behavior and predictions of machine learning systems comprehensible to humans is an ongoing research endeavor [27]. Improving interpretability is crucial in ensuring that AI-driven log analysis can be effectively understood and utilized.

Resource Consumption: GAI models face significant challenges related to resource consumption, including the computational power required for model training and operational costs, primarily concerning energy consumption during model operation [28]. Research efforts should focus on cost-effective learning methods to mitigate these resource-related challenges.

Security: Security remains a paramount concern in log management and analysis across all organizational departments. Entrusting log data to external entities or utilizing non-local tools increases security risks. GAI security models often exhibit "black box" features, raising security concerns [29]. To address these concerns and safeguard data privacy and intellectual property, organizations must implement security measures related to production models and strengthen defenses against potential attacks. This article proposes the establishment of a local structure to address these security issues.

Limitations in Large Language Models (LLMs): LLMs sometimes produce unrealistic yet

seemingly acceptable predictions, referred to as hallucinations [30]. To tackle this issue, new research directions, such as Augmented Language Models (ALMs), are explored [31]. ALMs are language models equipped with enhanced reasoning skills and the ability to utilize external tools and modules, expanding their processing capabilities and potentially mitigating hallucination-related limitations.

These challenges present opportunities for further research and innovation in the field of log analysis with the assistance of artificial intelligence, ultimately enhancing the capabilities and effectiveness of log analysis systems.

7. Conclusion

By harnessing the latest advancements in artificial intelligence technologies and locally deploying a large language model, without the need for outsourcing, organizations can securely engage with logs originating from various departments, such as network and data center management, marketing, human resources, finance, and accounting. These logs are systematically collected, stored, and efficiently managed, making them readily available for analysis through the implementation of a large language model. This approach culminates in the establishment of an accurate and effective system for addressing inquiries based on logs. It addresses the limitations of traditional log analysis by significantly reducing the time required for analysis. Moreover, it allows users to obtain answers simply by framing questions in natural language. This streamlined approach also leads to substantial cost savings by eliminating the necessity of hiring specialized experts. Taken together, these measures mark a significant stride toward organizational growth and success, enhancing the efficiency of log analysis and information retrieval, ultimately contributing to improved decision-making and operational effectiveness.

References

[1] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, “Deep learning for anomaly detection in log data: A survey,” *Machine Learning with Applications*, vol. 12, p. 100470, Jun. 2023, doi: 10.1016/J.MLWA.2023.100470.

[2] S. Muthurajkumar, S. Ganapathy, M. Vijayalakshmi, and A. Kannan, “Secured Temporal Log Management Techniques for Cloud,” *Procedia Comput Sci*, vol. 46, pp. 589–595, Jan. 2015, doi: 10.1016/J.PROCS.2015.02.098.

[3] R. Meyers, “Data highway and the digital transformation: arguments for secure, centralised log management,” *Network Security*, vol. 2020, no. 10, pp. 17–19, Oct. 2020, doi: 10.1016/S1353-4858(20)30119-7.

[4] Q. Wang, X. Zhang, X. Wang, and Z. Cao, “Log Sequence Anomaly Detection Method Based on Contrastive Adversarial Training and Dual Feature Extraction,” *Entropy*, vol. 24, no. 1, 2022, doi: 10.3390/e24010069.

[5] H. J. Liao, C. H. Richard Lin, Y. C. Lin, and K. Y. Tung, “Intrusion detection system: A comprehensive review,” *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, Jan. 2013, doi: 10.1016/J.JNCA.2012.09.004.

[6] T. Niesen, S. Dadashnia, P. Fettke, and P. Loos, “A vector space approach to process model matching using insights from natural language processing,” *Multikonferenz Wirtschaftsinformatik (MKWI)*, pp. 93–104, 2016.

[7] N. Wang, S. Sun, and D. OuYang, “Business Process Modeling Abstraction Based on Semi-Supervised Clustering Analysis,” *Business & Information Systems Engineering*, vol. 60, Jul. 2018, doi: 10.1007/s12599-016-0457-x.

[8] M. Landauer, F. Skopik, M. Wurzenberger, and A. Rauber, “System log clustering approaches for cyber security applications: A survey,” *Comput Secur*, vol. 92, p. 101739, May 2020, doi: 10.1016/J.COSE.2020.101739.

[9] M. Landauer, M. Wurzenberger, F. Skopik, G. Settanni, and P. Filzmoser, “Dynamic log file analysis: An unsupervised cluster evolution approach for anomaly detection,” *Comput Secur*, vol. 79, pp. 94–116, Nov. 2018, doi: 10.1016/J.COSE.2018.08.009.

[10] M. and M. Q. and H. B. and L. T. Dai Fei and Liu, “Refactor Business Process Models for Efficiency Improvement,” in *Cloud Computing, Smart Grid and Innovative Frontiers in Telecommunications*, G. and Q. M. and X. W. and H. T. Zhang Xuyun and Liu, Ed., Cham: Springer International Publishing, 2020, pp. 454–467.

[11] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, “Detecting Large-Scale System Problems by Mining Console Logs,” in *Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles*, in SOSP ’09. New York, NY, USA: Association for Computing Machinery, 2009, pp. 117–132. doi: 10.1145/1629575.1629587.

[12] Q. Lin, H. Zhang, J.-G. Lou, Y. Zhang, and X. Chen, “Log Clustering Based Problem Identification for Online Service Systems,” in *Proceedings of the 38th International Conference on Software Engineering Companion*, in ICSE ’16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 102–111. doi: 10.1145/2889160.2889232.

[13] J.-G. Lou, Q. Fu, S. Yang, Y. Xu, and J. Li, “Mining invariants from console logs for system problem detection,” in *2010 USENIX Annual Technical Conference (USENIX ATC 10)*, 2010.

[14] X. Zhang *et al.*, “Robust Log-Based Anomaly Detection on Unstable Log Data,” in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, in ESEC/FSE 2019. New York, NY, USA: Association for Computing Machinery, 2019, pp. 807–817. doi: 10.1145/3338906.3338931.

[15] M. Du, F. Li, G. Zheng, and V. Srikumar, “DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, in CCS ’17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 1285–1298. doi: 10.1145/3133956.3134015.

[16] W. Meng *et al.*, “Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs.,” in *IJCAI*, 2019, pp. 4739–4745.

[17] H. Gimpel *et al.*, “Unlocking the power of generative AI models and systems such as GPT-4 and ChatGPT for higher education: A guide for students and lecturers,” Universität Hohenheim, Fakultät Wirtschafts- und Sozialwissenschaften, Stuttgart, 2023.s

[18] Y. Liu *et al.*, “Generative artificial intelligence and its applications in materials science: Current situation and future perspectives,” *Journal of Materiomics*, vol. 9, no. 4, pp. 798–816, Jul. 2023, doi: 10.1016/J.JMAT.2023.05.001.

[19] A. Radford *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[20] T. Brown *et al.*, “Language Models are Few-Shot Learners,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020, pp. 1877–1901.

[21] L. Ouyang *et al.*, “Training language models to follow instructions with human feedback,” in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., Curran Associates, Inc., 2022, pp. 27730–27744.

[22] A. Gilson *et al.*, “How Does ChatGPT Perform on the Medical Licensing Exams? The Implications of Large Language Models for Medical Education and Knowledge Assessment,” *medRxiv*, 2022, doi: 10.1101/2022.12.23.22283901.

[23] Y. K. Dwivedi *et al.*, “Opinion Paper: ‘So what if ChatGPT wrote it?’ Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy,” *Int J Inf Manage*, vol. 71, p. 102642, Aug. 2023, doi: 10.1016/J.IJINFOMGT.2023.102642

[24] E. A. M. Van Dis, J. Bollen, W. Zuidema, R. van Rooij, and C. L. Bockting, “ChatGPT: five priorities for research,” *Nature*, vol. 614, no. 7947, pp. 224–226, 2023.

[25] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, and D. Yang, “Is ChatGPT a general-purpose natural language processing task solver?,” *arXiv preprint arXiv:2302.06476*, 2023.

[26] S. Bubeck *et al.*, “Sparks of artificial general intelligence: Early experiments with gpt-4,” *arXiv preprint arXiv:2303.12712*, 2023.

[27] Y. Liu *et al.*, “Cloud-VAE: Variational autoencoder with concepts embedded,” *Pattern Recognit*, vol. 140, p. 109530, Aug. 2023, doi: 10.1016/J.PATCOG.2023.109530.

[28] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” *arXiv preprint arXiv:1809.11096*, 2018.

[29] D. Ganguli *et al.*, “Predictability and Surprise in Large Generative Models,” in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, in FAccT ’22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 1747–1764. doi: 10.1145/3531146.3533229.

[30] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, and J. Weston, “Neural text generation with unlikelihood training,” *arXiv preprint arXiv:1908.04319*, 2019.

[31] G. Mialon *et al.*, “Augmented language models: a survey,” *arXiv preprint arXiv:2302.07842*, 2023.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[33] A. Chowdhery *et al.*, “Palm: Scaling language modeling with pathways,” *arXiv preprint arXiv:2204.02311*, 2022.

[34] M. Chen *et al.*, “Evaluating large language models trained on code,” *arXiv preprint arXiv:2107.03374*, 2021.

Journal of Optimization of Soft Computing (JOSC)

Vol. 1, Issue 2, pp: (26-38), December-2023

Journal homepage: https://sanad.iau.ir/en/Journal/josc

Research paper

FTRTA : Fault Tolerance and Reliable Transmissions Algorithm based on the Internet of Things

Mohsen Mozafari Vanani[1] and Pouya Khosravian Dehkordi [1*]

1. Department of Computer Engineering, Faculty of Engineering, Islamic Azad University, Shahrekord Branch, Shahrekord, Iran.

Article Info

Article History:
Received:2024/2/9
Revised:
Accepted: 2024/4/4

Keywords:
Internet of Things, fault tolerance, routing, data exchange, protocol, energy.

**Corresponding Author's Email Address: Khosravyan@gmail.com*

Abstract

The limitations of IoT have resulted in increased failures and the need for guaranteed fault tolerance to ensure adequate network performance. While previous studies have effectively improved fault tolerance by addressing various aspects of this area, previous methods are ineffective in ensuring the stability and accuracy of data exchange in the event of failure. The existence of this problem highlights the necessity of proposing a new method that can guarantee the stability and accuracy of data exchange to ensure network performance stability in case of failure. To address this, this research introduces a method called FTRTA, which is based on the enhancement of the RPL protocol and data distribution techniques. These distribution techniques are effective in improving load balancing and fault tolerance of network traffic. FTRTA is developed based on this technique and involves three operational steps. Firstly, it evaluates and analyzes the status of network nodes similar to when sending DIO messages. In the second step, it creates a network communication graph. Finally, in the third step, data transmission is performed using a distribution technique to ensure fault tolerance. Simulation results using Cooja software demonstrate the high performance of FTRTA in ensuring the stability and accuracy of data exchange, improving factors such as successful receptions and network throughput compared to similar studies.

1. Introduction

IoT has enabled all physical elements to communicate and interact with each other. In IoT, each member of the network has a separate digital identifier through which access, management, and communication with other members of the network are possible [1]. The IoT has become very important in today's world due to the wide range of benefits it has provided. One of the most basic applications of this technology in the current era is its use in medical, military, industrial, and general intelligence of physical elements [2]. This high progress makes it likely that many objects will join this network in the not-too-distant future [3]. However, it is noteworthy that this new network has extensive challenges and issues due to the incompleteness of technologies and standards [4]. One of the most important issues in these networks is the reliability and guarantee of tolerance. Due to the great importance of intelligent physical element management, IoT is very important in today's world, and the importance and applications of this new technology are increasing day by day. In IoT, due to severe resource limitations, wireless communications, network variable topology, and other related constraints, fault occurrence is very likely. Therefore, support for fault tolerance is a very important issue. However, the importance of this field is doubling due to the important areas of

application of IoT [5]. Therefore, providing measures to improve reliability and support for fault tolerance is considered an undeniable necessity for IoT to ensure the continuity and accuracy of network performance [33]. IoT is a diverse network with many limitations. The existence of these unique features and limitations raises a number of issues related to this network. One of the most important of them is related to fault tolerance of data transmission. [32] On the other hand, the existence of these special and different features has made the use of traditional techniques (practical techniques in other wired and wireless networks) not suitable for these networks [6 and 7]. Therefore, given the importance and necessity of fault tolerance in maintaining the continuity and accuracy of IoT performance, extensive research has been provided to improve issues related to this area. Most of these studies have focused on improving routing and increasing the reliability of data exchanges. In IoT, nodes are unable to communicate directly with each other and root nodes due to limitations in equipment and their communications. For this reason, data exchanges are performed through other members of the network in multi-hop [8 and 9]. This performance makes network activity dependent on correct routing and reliable data exchange. Accordingly, a large section of the research in the field of fault tolerance has focused on this issue and has tried to increase the reliability and guarantee the accuracy of this vital category [10-17]. To this end, most research is based on the development of the RPL protocol [18] to improve the reliability of this routing protocol. It is worth noting that RPL is the most important IoT routing application protocol and is widely used in these networks. However, studies of past research have shown that there were some important issues associated with the protection of fault tolerance that make it necessary to provide more effective research in this area. In fact, most past research has focused on improving parent node choices and increasing the reliability of intermediate routes. However, improving parent choices and increasing the reliability of intermediate routes are very important issues, but neglecting other aspects of fault tolerance, especially fault coverage and ensuring the accuracy of exchanges, will lead to instability and loss of network performance. In fact, the safe choice of parents is a necessary condition, but it is not enough on its own.

In order to improve this issue, this article introduces a method called Fault Tolerance and Reliable Transaction Algorithm for IoT (FTRTA). FTRTA is based on the optimization of the RPL protocol and is based on the efficiency of data distribution techniques, and based on this, it tries to improve the reliability combined with the support of fault tolerance. FTRTA performance is generally divided into three main steps. In the first step, the reliability of the network nodes is assessed. In the second step, the DODAG graph is formed based on the proposed FTRTA measures. In the third step, data exchange is based on the data distribution technique [34].

The primary highlights of the Fault Tolerance and Reliable Transaction Algorithm for IoT (FTRTA) are the following:

- The main goal of FTRTA is to improve the tolerance fault of IoT network data exchanges.
- RPL optimization is used to improve routing and increase the reliability of intermediate routes.
- The efficiency of FTRTA is evaluated using COOJA simulator.

In the second section of this article the past works will be examined. Details of the proposed FTRTA will be presented in the third section. In the fourth section, the proposed method based on the Cooja software will be simulated and its performance will be evaluated. At the end, the article will be concluded.

2. Related Works

As noted, most reliability-based research has been developed based on the RPL protocol to focus on improving routing and data exchange. Some of these articles have focused on enhancing the RPL objective function and have attempted to increase the reliability of this protocol [10, 14-16, and 19-21]. Many studies have concentrated on energy-efficient routing and data exchange [23-26]. Others have been designed to evaluate link stability [11, 17, and 22] and have been developed based on assessing the condition from end to end of the intermediate routes [12 and 15]. However, methods that focus on improving the RPL objective function have shown relatively better performance in enhancing the reliability of data exchanges. Some of the most significant studies in this area are discussed below. Sennan et al. proposed a protocol called EDADA (Energy and Delay Aware Data Aggregation) [19]. This two-step method involves data compression techniques in the first step and evaluation of energy status and delay in the second step to select intermediate routes based on node assessment. While examining the energy situation has been effective in improving parent selections, focusing solely on this criterion may not meet all needs.

Sanmartin and colleagues introduced a method called SIGMA-ETX (SIGMA Expected Transmission Count) to enhance reliability [15]. This approach, based on RPL development and SIGMA evaluation, performs routing elections by analyzing the expected transmission count (ETX) and its variance for intermediate route nodes. This method has been effective in assessing the end-to-end state of intermediate routes, leading to improved reliability in data exchanges.
A method named MRHOF (Minimum Rank with Hysteresis Objective Function) was introduced by Lazarevska et al., based on RPL protocol development to enhance reliability [16]. In MRHOF, the network communication graph formation is based on evaluating the ETX index, node energy efficiency, and signal quality, followed by parent selections. Simulation results of this method demonstrate improved reliability in data exchanges. To enhance reliability, Sousa et al. introduced ERAOF (A New RPL Protocol Objective Function) [20]. This method optimizes the RPL protocol's objective function by evaluating ETX index, remaining energy metrics, and connection quality to make decisions. This approach has been effective in improving quality and increasing reliability in data exchanges. The MIQRP protocol (Multiple Instances QoS Routing in RPL) introduced by Nassar et al. [10] aims to improve exchange reliability by introducing a new objective function called mOFQS. This function evaluates nodes based on ETX, energy, and delay, leading to appropriate parent selections. Simulation results indicate improved data interactions and reduced energy consumption.
In 2019, a method called EEMA (Energy Efficient and Mobility Aware) was introduced based on RPL protocol development to improve stability and reliability in communication, particularly for mobile networks. EEMA evaluates node mobility based on received signal quality and makes decisions by combining energy evaluations. This design has proven effective in enhancing reliability and stability of links, especially in mobile networks, though it lacks fault tolerance support. Vaziri et al. introduced a method called Brad-OF (Enhanced Energy-Aware Method for Parent Selection and Congestion Avoidance) to control congestion and enhance communication ability [21]. In Brad-OF, high-density node presence in the network communication graph is prevented, and nodes are evaluated based on ETX, remaining energy, and delay for parent selection. Simulation results show congestion control and improved reliability in data exchanges.
The MAPS protocol (Mobility-Aware Parent Selection for Routing) proposed in 2019 aims to improve network communication graph stability, especially in mobile networks. MAPS evaluates node mobility to select and form communication graphs based on signaling power for link quality and stability assessment. Parent elections are made based on stability and signaling quality predictions. Previous studies have attempted to improve routing reliability and data exchanges based on various measurements. However, they did not propose measurements to cover faults and guarantee the accuracy and continuity of data exchanges. This paper focuses on optimizing the RPL protocol and distribution techniques to address this fundamental issue. The studies are analyzed and discussed in Tables (1) in terms of evaluation metrics, purpose, simulation tools, and applied strategies.

Table 1. The review of introduced articles focusing on the RPL OF

Reference	Evaluation metric	Simul Ator	Goal
EDADA [19]	Energy, delay	Cooja	Energy optimazation
SIGMA [15]	ETX	Cooja	PDR Improvment
MRHOF [16]	Energy, ETX, RSSI	Cooja	QoS Improvment
ERAOF [20]	Energy, link quality, ETX	Cooja	QoS Improvment
MIQRP [10]	ETX, Energy, delay	Cooja	QoS Improvment
EEMA [17]	mobility	Cooja	PDR Improvment
Brad‑OF [21]	Energy, ETX, Delay	Matlab	Congestion control
MAPS [30]	mobility	Cooja	PDR Improvment

3. Definitions

3.1. Communication Model

The desired network includes m nodes and a DODAG root. The nodes are randomly located in the networked environment. FTRTA follows the tree-based approach to exchanges. In this approach, the sensors, after collecting data, send it to the root node of the tree structure in response to the proposed measures. Data is distributed and sent to parents for the purpose of maintaining fault tolerance. How to distribute and send is determined

by the fault tolerance and FTRTA decision-making. Finally, the root node receives the sent data and recovers it based on the applied technique, even if an error occurs. FTRTA is based on the RPL protocol, so the FTRTA communication model is consistent with this routing protocol [18]. RPL supports three different types of communication: point-to-point communication (P2P) for the connection between two nodes in the DODAG graph, point-to-multipoint communication (P2MP) to send traffic from the root node to the pages, and multipoint-to-point communication (MP2P) to collect and send data from network nodes to the DODAG root.

3.2. Energy Consumption Model

In networks with energy limitations, in order to design a routing technique, it is necessary to determine the network energy consumption model. Sending and receiving data on the IoT network are associated with the energy consumption of the nodes. The energy consumption for this purpose is determined by a function of the distance between the sender and receiver. In addition, the process of listening to the media to receive data and the sleep state of the nodes is accompanied by energy consumption. In the standard of IEEE 802.15.4 WSNs, energy consumption of sensor (a) on the link of $e\,(a,b) \in E$ for processing a message is evaluated by equation (1) [27].

Where E_c^a is Energy Consume of node a, E_l^a, E_t^a, E_r^a and E_s^i are the energy consumed during the periods of listening, transmitting, receiving and sleeping, respectively. I_t, I_r, I_l and I_s are the current drawn in the transmitting receiving, listening and sleeping modes, respectively. t_s^a, t_l^a are the current drawn in the transmitting, receiving, listening and sleeping modes, respectively. V is the battery voltage of the nodes, L (bits) is the packet length and BR (Kbps) is the data rate in the WSN.

In networks with energy limitations, in order to design a routing technique, it is necessary to determine the network energy consumption model. Sending and receiving data on the IoT network are associated with the energy consumption of the nodes. The energy consumption for this purpose is determined by a function of the distance between the sender and receiver. In addition, the process of listening to the media to receive data and the sleep state of the nodes is accompanied by energy consumption. In the standard of IEEE 802.15.4 WSNs, energy consumption of sensor (a) on the link of $e\,(a,b) \in E$ for processing a message is evaluated by equation (1) [27].

Where E_c^a is Energy Consume of node a, E_l^a, E_t^a, E_r^a and E_s^i are the energy consumed during the periods of listening, transmitting, receiving and sleeping, respectively. I_t, I_r, I_l and I_s are the current drawn in the transmitting receiving, listening and sleeping modes, respectively. t_s^a, t_l^a are the current drawn in the transmitting, receiving, listening and sleeping modes, respectively. V is the battery voltage of the nodes, L (bits) is the packet length and BR (Kbps) is the data rate in the WSN.

$$E_c^a = E_l^a + E_t^a + E_r^a + E_s^a = \left(t_l^a I_l + (I_t + I_r)\frac{L}{BR} + t_s^a I_s\right) V \quad (1)$$

Based on reference [28], values of t_s^a and t_l^a are based on equation (2) and (3):

$$t_s^a = BI - SD = aBaseSD \times (2^{BO} - 2^{SO}) \text{symbols} \quad (2)$$

$$t_l^a = BI - (t_t^a + t_r^a + t_s^a) \quad (3)$$

We assume that $E_r^a = 0$ (or $t_r^a = 0$) if a is a source node (Tx) and $E_t^a = 0$ (or $t_t^a = 0$) if a is a destination node (Rx).

Therefore, the consumed energy to send and receive of data is evaluated based on the equations (4) and (5).

$$E_c^a = t_l^a I_l + I_t \frac{L}{BR} + t_s^a I_s \quad (4)$$

$$E_c^a = t_l^a I_l + I_r \frac{L}{BR} + t_s^a I_s \quad (5)$$

3.3. The Network Assumptions

- Network nodes are homogeneous and fixed.
- Nodes are randomly located on the network.
- The network topology is various.
- Nodes don't have GPS equipment.
- Nodes have a unique identifier the energy of them is limited.
- It is assumed that fault occurs for 10 percent of sent data.

4. Fault Tolerance and Reliable Transaction Algorithm for IoT (FTRTA)

FTRTA is designed based on optimizing RPL protocol and distribution technique. Its goal is improvement of reliability and data fault tolerance. FTRTA to implement is compatible with IoT networks and RPL protocol. FTRTA is segmented to three steps:

1. Analysis of node status along with the process of sending DIOs.
2. Create a DODAG graph based on the proposed FTRTA measures.
3. Data transmission according to data distribution technique.

In the following is described detailed description of each step with the relevant flowchart.

4.1. Network Nodes Status Analysis

This step is based on the developed ROL protocol DIO messages, and its most important purpose is to assess the reliability and position of nodes in the network. Figure (1) presents this step of the FTRTA and describes its details below.

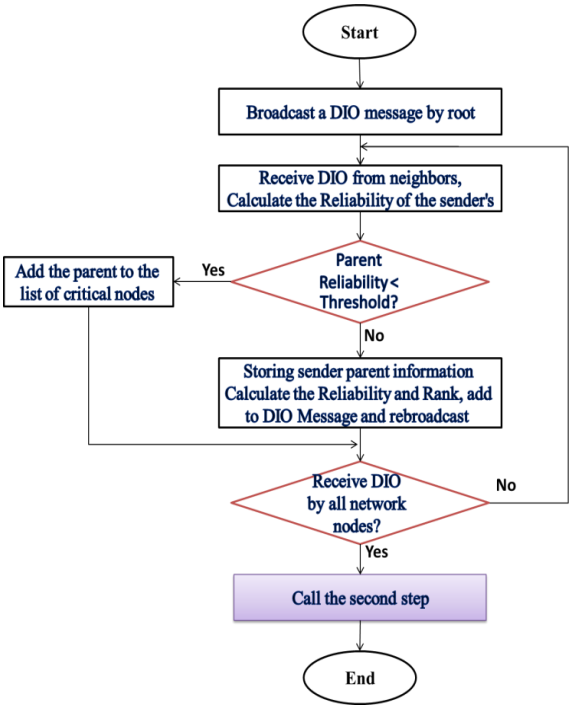

Figure 1. The first step of the proposed FTRTA

According to the flowchart in Figure (1), when the network starts activity, the root node (based on sending DOI message of RPL protocol [18]) creates a DIO message and sends it to the network in the form of a broadcast. The purpose of distributing this message in the RPL protocol is to consider the node status from the root. In FTRTA, this goal includes considering reliability in addition to the node status. The sending of DIO process in FTRTA is based on RPL. Therefore, within a specified period of time, it is repeated to update the network topology. In the RPL protocol, after sending the DOI message from the root, it is shared among nodes so that all members of the network receive it. Sending DIOs in FTRTA has three differences compared to in the RPL protocol:
1. The first difference: In addition to performing operations related to the process of sending DIOs in accordance with the RPL, nodes also assess their reliability and add the result of this evaluation to the DIO message.
2. The second difference: If the DIO message is received from a parent whose reliability is less than the threshold, the parent will be added to the list of critical nodes and will be removed from the list of parent collections. This is intended to prevent the network graph from being constructed by low-reliability nodes. In FTRTA, the value of the confidence threshold is 0.1.
3. The third difference: The rank in the RPL protocol is evaluated according to the position of the nodes relative to the root. In FTRTA, however, in addition to location, node reliability is also influential in ranking. Accordingly, the ranking of nodes, in addition to location, also depends on the reliability of nodes. Figure (2) provides an overview of the DIO message sent to FTRTA. The details of the sent DIO message fields in FTRTA are as follows:

- IoT Graph Information-Base on DIO: This field corresponds exactly to the content of the DIO message of the RPL protocol and includes DADOG graph information, including root ID, graph ID, version, and other related items. Sufficient details are provided in [18].
- Rank: This field contains the sender's node rank. Rank refers to the position of the node relative to the root. If the rank of a node is lower, the node is closer to the root. Rank in RPL is rated for node distance from the root, but in FTRTA, in addition to node position, its level of reliability is also involved in ranking rating.
- RL: This field contains the reliability of nodes and is calculated according to the three concepts of energy, fault rate, and probability of data loss. Then it is added to the DIO.

Figure 2. DAO message in FTRTA

According to the flowchart presented in Figure (1), after receiving DIO messages by network nodes, first the reliability of the sender of the message is checked and if it is less than the threshold, the parent is removed from the list of total parents. . Otherwise, the receiver node stores the parent's information and puts it in the parent collection list. In equation (6), the details of detection and eliminating critical parents are presented. RLi is equivalent to i node reliability (reliability value is between zero and one, so the smaller the value, the lower the node reliability) and DF is the critical node detection flag.

$$DF = \begin{bmatrix} 0 & IF & (RL_i > 0.1) \\ 1 & Else & \end{bmatrix} \quad (6)$$

After performing the process of identifying critical parents, the receiver node evaluates the reliability based on the equation (7) and the rank according to the equation (8) and adds the result of these calculations to the DIO message. The message is then resent on the network in the form of broadcast. In equation (7) RLi is equivalent to node i reliability, Eri is remain energy of node i, EInit is equivalent to i node energy at the first instant, ERi is the number of fault occurring during i node previous interactions, $\sum$ No. of success Sending is the total number of messages that k node has been successful in sending them. No.of all Packet Received is equivalent to the total number of messages received by k node, and α is the various valuation coefficient according to the reliability evaluation criteria and has a value between zero and one.

$$RL_i = \alpha.\left(\frac{Er_i}{E_{Init}} \times \frac{1}{Log(ER_i)}\right) + (1 - \alpha).\left(\left[\frac{\sum \text{No.of Successl Sending}}{\text{No.of all Packet Received}}\right]\right) \quad (7)$$

In equation (8) Rki is equivalent to the rank of a node, RkF is the rank of parent node i, MinRkInc is the constant rate of increase in children's rank, and ω is an influential indicator of reliability in node rank evaluation and has a value between zero and one. The larger the value of it, the greater the effect of reliability on node rank evaluation.

$$Rk_i = Rk_F + MinRk_{Inc} + \omega \times \left(\frac{1}{RL_i}\right) \quad (8)$$

The process provided in connection with sending DIOs is repeated until all the nodes of the network finally receive the DIO message. Based on the performance of this step, unreliable parents are identified, nodes are informed of their location and network topology, and nodes are informed of their parents' reliability. After sending the DIOs, the second step of FTRTA is executed to form the DODAG graph.

4.2. Creating a DADOG Graph

The purpose of this step is to select parents and create an improved DADOG graph based on the proposed FTRTA measures. Figure 3 of presents this step of the proposed method.

This step of the FTRTA is in accordance with the sending process of DAO in the RPL protocol. The difference is that the objective function and the parents are evaluated and selected according to the FTRTA measures.

According to the flowchart presented in Figure (3) after completing the sending of DIOs, the nodes first check the number of parents after creating the DAO message (in accordance with the basic RPL protocol). Based on the results of this study, two different cases are presented as follows.

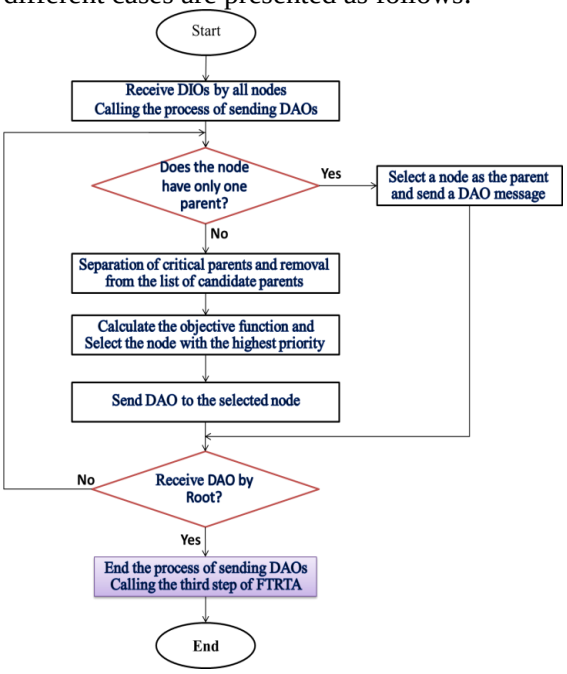

Figure 3. Flowchart of the second step of the proposed FTRTA

1) The node has only one parent. In this case, the parent is selected as the original parent and the DAO message is sent to them.

2) The node has more than one parent. In this case, first the critical parents are separated from the other parents and then the best parent is selected as the main parent from the remaining parents. After selecting the parent, the DAO is sent to the selected parent.

The main parent in the RPL is selected based on the objective function. The RPL objective function is evaluated based on the indicator called ETX, and the main parent is selected based on the results of this evaluation. In the proposed FTRTA, the objective function has been improved, and in addition to the ETX index, node rank and reliability also play important roles in the evaluation and selection of the main parent. This is intended to improve the reliability of the DODAG graph in FTRTA. Equation (9) provides details of the objective function assessment in FTRTA. In this regard, OF_i is equivalent to the objective function evaluated for parent I, RL_i is equivalent to parental value k in terms of capability, ETX_i is equivalent to the expected transfer rate of node i and Rk_i is equal to parent value of i. w_1, w_2 and w_3 are equivalent to variable valuation coefficients to the evaluation criteria of the proposed objective function. These coefficients have value between 0 and 1 that sum of them is 1.

$$OF_i = (w_1 \times RL_i) + (w_2 \times ETX_i) + \left(w_3, \frac{1}{Rk_i}\right) \quad (9)$$

ETX is a factor for elections of parents in RPL protocol that is evaluated based on successful possibility for sending packet and receiving ACK. Details of considering this factor is in equation (10). So, df_i is the measured probability that a packet is received by the neighbor and dr_i is the calculated probability that the acknowledgment packet is received successfully.

$$ETX_i = \frac{1}{df_i \times dr_i} \quad (10)$$

Based on the result of the equation (9), each child chooses the best parent as the main parent and sends the DAO message to him. By receiving the DIO message, the parent stores the child's information and this process is repeated until the DAO message is finally received by the root. By receiving DAO messages from the root, the main parents are determined and network graph are formed. After this step, the final step of the FTRTA is for data transmission.

4.3. Data Transmission Based On Data Distribution Technique

After forming the DODAG graph, whenever the node intends to send data, it performs the sending process based on the performance of this process. The main purpose of this step is to ensure data interaction fault tolerable. Figure 4 shows the flowchart of this step and then analyzes of its performance details have been prepared.

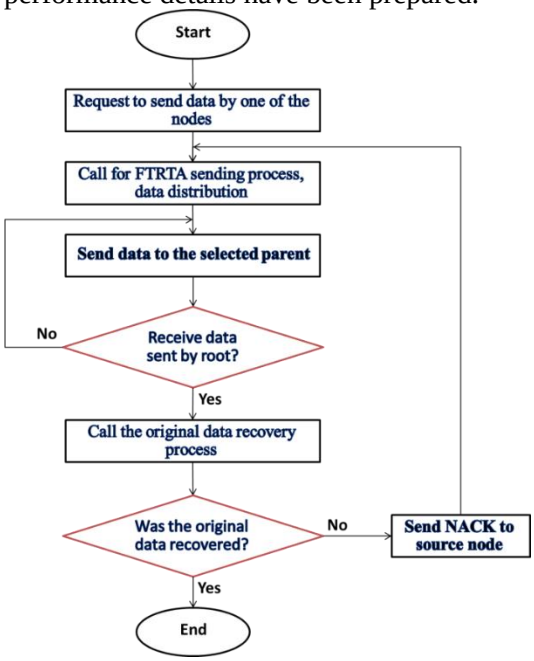

Figure 4. The flowchart of the third step of the proposed FTRTA

The data transmission in FTRTA is based on data distribution techniques, aiming to ensure fault tolerance even in the event of errors and data loss. In this technique, the sender node in a distribution process divides the data into several sections (n sections) and sends them to the root through the parent nodes and the network communication graph. The root node can recover the main data by receiving a certain number of these sections (k < n). During the sending process, if one or more sections are corrupted or lost, the root node can retrieve and recreate basic information by receiving other sections, ensuring fault tolerance.

Additionally, the distribution technique can be adaptably adjusted to meet fault tolerance requirements, making it a prominent benefit of this technique. Data distribution and retrieval processes can be defined so that receiving the minimum sent sections still allows for the retrieval of basic information. It is important to note that the greater the need for fault tolerance, the higher the resource consumption, as the amount of network consumption resources is related to the required level of fault tolerance.

To model the proposed distribution technique, a multi-sentence linear equation is required. This equation is adjusted as a variable according to the fault tolerance requirements. For instance, if we want the root node to recover the original information by receiving 3 sections during the send processes, the number of linear equation components must be considered as 4 components. Similarly, if we want the root node to recover the original information by receiving two sections, then the number of components of the linear equation must be considered as 3 components. The source node with different values of these components can create any number of required sections for sending. Note that the value of the components is random, but prime numbers must be used.

The number of sections created for sending is not related to the ability to retrieve main information. The recovery process is independent of the number of sections created and is done with a certain number of sections. The basis of the data recovery process at the destination is determined by the number of linear equation components. For example, if the linear equation has 3 components, the root node can recover the main information by receiving three sections. However, if it receives less than 3 sections in this scenario, the main information will not be recoverable.

Algorithm (1) provides details of the FTRTA data distribution process, intended for a scenario in which the information is divided into three sections, and the root node will be able to recover

the main information by receiving the second section.

Algorithm 1. Send data based on FTRTA

data distribution process

Every node Request to send data {
Scenario for (3,2);
// Divide the data into 3 sections and retrieve the original data based on the 2 sections
For This Scenario → $\mathbf{F(x) = c_1, x + c_0\ Mod\ u}$
// $\mathbf{c_1}$ and $\mathbf{c_0}$ are randomly selected (for example Respectively 2 and 4)
//$\mathbf{u}$ A Prime number is selected that is greater than all values (for example 7)
Assign the Prime numbers to the variable c to create sections;
$\mathbf{F(2) = (4 + 4\ Mod\ 7) \rightarrow F(2) = 1}$; $\mathbf{F(3) = (6 + 4\ Mod\ 7) \rightarrow F(3) = 3}$;
$\mathbf{F(5) = (10 + 4\ Mod\ 7) \rightarrow F(5) = 0}$;
After create sections, send sections for Root by Selected father;
After recevied sections by Root;
original data recovery Process Summon based on Lagrange Equ;
For example two sections $\mathbf{F(3)}$ and $\mathbf{F(2)}$ Recevied, and $\mathbf{F(5)}$ Lost;
Recovery by two sections;
$\mathbf{F(3) = 3}$, $\mathbf{F(2) = 1} \rightarrow \mathbf{F(x) = c_1, x + c_0\ Mod\ u} \rightarrow \begin{bmatrix} \mathbf{3c_1 + c_0 = 3} \\ \mathbf{2c_1 + c_0 = 1} \end{bmatrix} \rightarrow$
$\mathbf{c_1 = 2} \rightarrow \begin{matrix} \mathbf{(3 * 2 + c_0)\ Mod\ 7 = 3} \\ \mathbf{(2 * 2 + c_0)\ Mod\ 7 = 1} \end{matrix} \rightarrow \mathbf{c_0 = 4}$

Based on what has been provided, data exchange is performed and fault tolerance is ensured during the send process.

5. Performance Evaluation

5.1. Simulation Setup

To implement and evaluate FTRTA performance, this method has been simulated with the Cooja software [29] and compared with the MIQRP [10] and RPL [18] method]. For the experiments, we used the Contiki IPv6 / 6loWPAN model and the RPL text protocol called ContikiRPL [30]. The configuration parameters of the simulation scenarios are presented in Table (2).

Table 2. Simulation parameters.

Parameter	Value
Operating system	Contiki master version (2.7)
Loss Model	Distance loss
Sensors	Skymote
Adaptation	6LoWPAN
Communication protocol	CSMA, RDC contikimac, IEEE 802.15.4, ContikiRPL, IPv6
OF	OFQS, FTRTA (Proposed OF), ETX
The number of sensors	15, 30, 45, 60, 75, 90
Network area	500M*500M
Microcontroller unit	ARM Cortex M3, 32-bits, 72 MHz, 64 kB RAM
Data packet size	30 bytes
DIO, DAO and DIS size	16, 16, 4 byte
Band width	250 Kbps
Transmission layer	UDP
Initial energy of sensors	1500 mA
Radio model	Unit Disk Graph Medium (UDGM)
Data errors during sending	10% of all data
Simulation time	600 s
Result	Avg 20 round

The simulations used Contiki Power trace to investigate the energy consumption of nodes. Power trace output is the total energy consumption of the nodes while they are active [31]. Table (3) shows the amount of energy consumption when the nodes are active.

Table 3. The power consumption when mote sky is active

Mode	Energy Consume
The MCU is active and the radio unit is in reception mode	2.18 mA
MCU is active and the radio unit is in sending mode	19.5 mA
MCU is active and radio unit is inactive	µA 1800
The MCU is idle and the radio unit is inactive	µA 54.5
MCU ready to operation	µA 5.1

The energy in the 6LowPAN network is checked by the duty cycle in which the radio units are only active at the time of sending and receiving. Contiki MAC is used for this purpose. Contiki MAC is an MAC standard in which the radio unit is 6% active when there is no traffic. Based on the presented topics, the equation (11) presents energy consumption based on time.

$$\text{Energy Consume}_i(\text{mj}) = \text{Send} \times 19.5\text{mA} + \text{Received} \times 21.8\text{mA} + \text{Cpu} \times 1.8\text{mA} + \text{Lpm} \times 54.5\text{mA} \quad (11)$$

5.2. Result

This section details the simulation results. In the experiments, the number of nodes for different scenarios varies between 15 and 90, and the result of each scenario is examined and displayed with an average of 20 cycles. In order to evaluate the performance of the methods, the criteria of percentage of data loss, energy consumption, delay and transit have been used. Details of these criteria are provided below.

Data loss rate: This criterion is defined in relation to the lost data rate in relation to all sent data and is evaluated according to Equation (12).

$$\text{PDR} = \frac{\sum \text{No.of Packet Drop}}{\sum \text{No.of Packet Send}} \quad (12)$$

Network delay: This criterion depends on the average time required to receive the data by the root node and is evaluated based on the equation (13).

$$\text{Delay} = \frac{\sum_{j=1}^{\text{No,of send data}} \text{Arrival Time}_j\text{-Send Time}_j}{\text{No,of send data}} \quad (13)$$

Energy consumption: This factor is related to the total energy consumption of network sensors and is evaluated based on the equation (14).

$$\text{Energy Consume} = \sum_{i=1}^{n} \text{Energy Consume}_i(\text{mj}) \quad (14)$$

Network throughput: This factor is related to the network throughput (network actual interaction rate) and is evaluated based on equation (15).

$$\text{Network Throughput} = \frac{\sum \text{No of byte Receive} \times 8}{\text{Time (s)}} \text{bps} \quad (15)$$

5.2.1. The effect of network density

In this section, the effect of node density on the compared methods is investigated. Therefore, the number of nodes in different scenarios is considered to be between 15 and 90 nodes, randomly placed in the network. The traffic rate in the network is set at 40 PPS (packets per second) for different scenarios.

Data loss rate: Figure (5) illustrates the effects of node density variation on the percentage of network data loss. It is observed that in all three methods, as the number of nodes increases, the percentage of data loss also increases. This increase can primarily be attributed to the rise in network traffic rate and its interruptions. Furthermore, with an increasing number of nodes, the length of intermediate routes (the number of hops in the intermediate routes) also increases, leading to a higher likelihood of data loss. FTRTA outperforms MIQRP and RPL in terms of establishing reliable routing based on parental choices and ensuring fault tolerance. Additionally, FTRTA offers the capability to identify unreliable nodes, further reducing the risk of data loss. Although MIQRP excels in selecting parents and creating secure routes, it lacks the ability to tolerate faults and mitigate the negative impacts of uncertain nodes. On the other hand, RPL, as a fundamental method, does not incorporate measures to enhance reliability and fault tolerance, resulting in inferior performance compared to the other two methods.

When the network consists of 15 nodes, the data loss percentage for FTRTA is approximately 10%, surpassing MIQRP and RPL by 3.5% and 6%, respectively. However, with 90 nodes present, the data loss rate for FTRTA reaches 40%, marking a success rate 9.5% and 15% higher than MIQRP and RPL, respectively. The study indicates that FTRTA demonstrates greater success in scenarios with an increasing number of nodes.

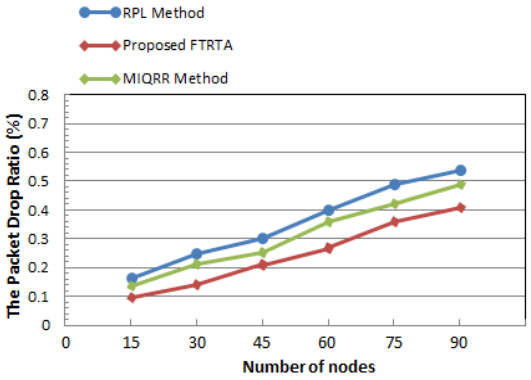

Figure 5. Packet drop ratio by changing the number of nodes

Network energy consumption: Figure (6) illustrates the effects of varying node presence on network energy consumption. In all three methods, the level of network energy consumption has increased with the rising number of nodes. However, the consumption rate in FTRTA is higher than that of the MIQRP and RPL methods, and it escalates as the number of nodes increases. This increase is attributed to the rise in network traffic rates facilitated by FTRTA, consequently leading to an increased network energy consumption, aligned with its measures aimed at ensuring fault tolerance. This may present a limitation in terms of guaranteeing fault tolerance. Comparatively, MIQRP outperforms RPL in this aspect. MIQRP has been more successful in reducing energy consumption compared to RPL, achieved by considering the energy status of nodes during parental selections and optimizing delay-based exchanges.

By analyzing the energy status and making appropriate decisions based on requirements, both the concept of optimization and the amount of consumption are well supported. EDADA focuses on examining the energy levels of parent nodes but does not provide the capability to manage energy. RPL has not implemented any measures in this area and has resulted in increased energy consumption compared to the other two methods.

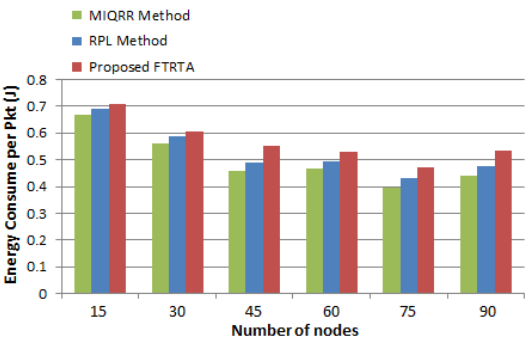

Figure 6. Network energy consumption by changing the number of nodes

Network Delay: Figure (7) shows the effect of node density on delay. FTRTA optimizes intermediate routs during the formation of graph according to rank in elections of parents. It also guarantees communication continuity and fault tolerance during transmission of data based on the distribution technique. On this basis, in addition to optimizing intermediate routs, service connectivity is guaranteed in different situations, which has resulted in improved end-to-end delay. MIQRR has been successful in improving parental choices and optimizing delay based on improving the objective function-based delay index. However, this method does not take proper measures to maintain the continuity of service. The ETX-based RPL objective function is evaluated and the parents are selected based on this. Therefore, this protocol has a higher delay than the other two methods.

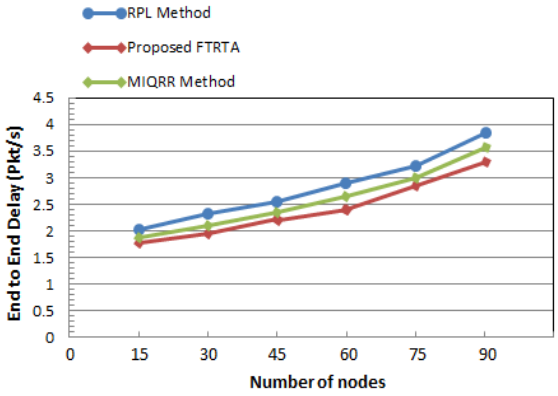

Figure 7. End to end delay by changing the number of nodes

Network throughput: Figure (8) shows the effects of the presence of variable nodes on the network throughput. FTRTA, based on its proposed steps, provides the ability to detect critical nodes, increases the reliability of the network communication graph as much as possible, and ensures fault tolerance of exchanges. The result of this three-step design is a total improvement in exchanges and an increase in network throughput. MIQRP is effective in graph reliability, but this method does not cover other aspects, especially fault tolerance during exchanges. RPL has not any measures to support reliability and fault tolerance, and due to issues caused by this inefficiency, it has been associated with a further decline in network throughput compared to the other two methods.

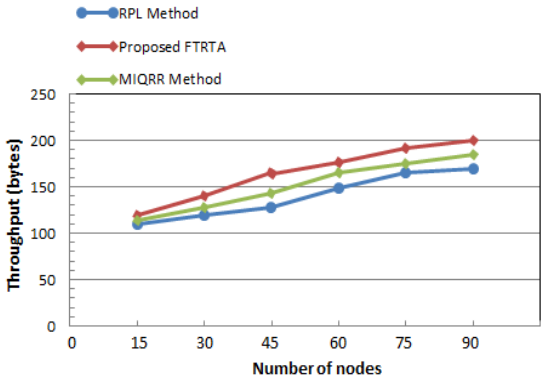

Figure 8. Network throughput by changing the number of nodes

5.2.2. The effect of traffic loads

In this section, the effect of traffic load on the methods being compared is investigated. So, the number of nodes in different scenarios is considered to be 50 nodes that are randomly placed in the network. The rate of traffic sent in the network for different scenarios is among 20-100 PPS for different scenarios.

• Packet Drop Ratio (PDR)

Figure (9) shows the effects of traffic load on PDR. With increasing traffic load, PDR has increased in all three methods. The reason for this is disruptions and problems caused by increased traffic and congestion. But this increase for FTRTA was lower compared to the other two methods. This is because of the advantages that FTRTA provides in terms of data distribution. Due to the measures of the distributed data exchanges for providing fault tolerance, FTRTA has caused the traffic in the intermediate routes is distributed and balanced. This distributed transmission is very effective in improving congestion issues, especially in high-traffic scenarios. This performance, along with other FTRTA measures to support transaction reliability, has led to improved successful delivery and reduced PDR for the proposed method. This is while neither of the two methods provides the ability to balance traffic. This inefficiency has increased PDR, especially in high-traffic scenarios. However, the increase in PDR for RPL is more severe than for MIQRP. RPL is primarily designed for low traffic networks and is inefficient in high traffic scenarios. When the traffic sent through the network was 20 PPS, PDR for FTRTA was about 10% that it was 3.5% and 6% more successful than that of MIQRP and RPL, respectively. At a traffic rate of 100 PPS, the PDR for FTRTA was 40%, which was 9.5% and 15% more successful than for MIQRP and RPL, respectively. This study concludes that the effect of traffic distribution is greater in scenarios with higher traffic rates.

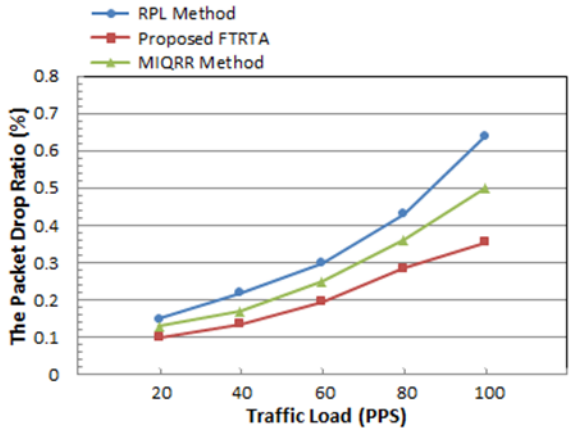

Figure. 9 Packet drop ratio by changing the traffic load

• Network energy consumption

Figure (10) shows the effects of traffic rates on the network energy consumption. With increasing traffic rates, energy consumption has also increased due to the direct effect of increasing data transmission on network energy consumption. The important point is that with increasing traffic load, energy consumption for FTRTA has been less than for the other two methods because of the effects of distributed data transmission on maintaining the balance of route traffic and reducing the negative effects of increased congestion, especially energy consumption. Data distributed transmission in FTRTA balances traffic on the network communication routes. This has been effective in reducing buffer overflow, data loss, resend and other congestion issues, which are sever in heavier traffic scenarios. These issues increase energy consumption, which FTRTA effectively prevents from occurring these. MIQRP has been successful in reducing energy consumption by considering energy in parent selection and graph formation, but the lack of capacity to balance traffic has exacerbated congestion issues that its effect on increasing the energy has been more severe, especially in scenarios with more traffic.

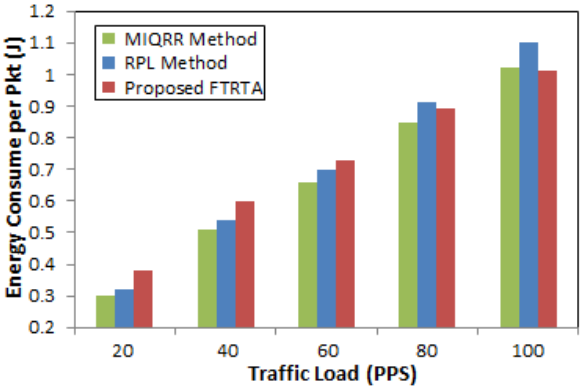

Figure. 10 Netwotk energy consumption by changing the traffic load

• Delay

Figure (11) shows the effects of traffic rates on the network delay. Delay is directly related to increased traffic load. Accordingly, with increasing traffic, delay has also increased, which has intensified for scenarios with heavier traffic, because congestion has increased in these scenarios, which has a reciprocal effect on increasing delay. The RPL protocol is inefficient in terms of traffic management and congestion control, and in this regard, with increasing traffic load, delay in this protocol has increased much more. Although MIQRR has considered delay in its routing, this alone is not enough, especially for high-traffic scenarios. In addition to creating a secure graph and improving the reliability of data exchanges, FTRTA operates during exchanges in such a way that it maintains the traffic balance of the routes by sending distributed data. This has resulted in improved delay for MIQRR.

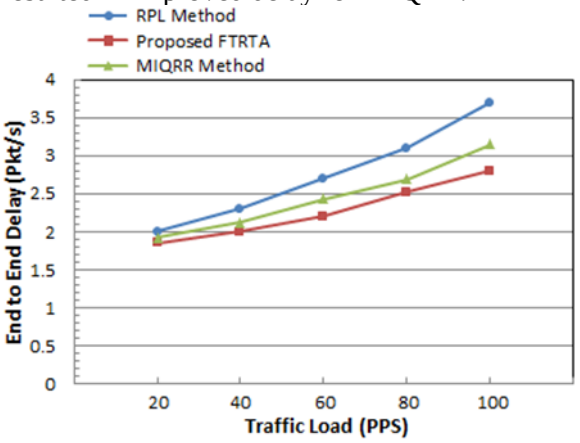

Figure. 11 End to end delay by changing the traffic load

• Network throughput

Figure (12) shows the effects of traffic load on network throughput. With the increase of traffic load up to 80 PPS, the throughput for FTRTA and MIQRP has an increasing trend, but after that it has had a decreasing trend. According to what was presented, FTRTA, in addition to forming a reliable graph and supporting data reliability, has also been successful in maintaining traffic balance and controlling congestion of intermediate routes. The result of this successful performance has been improved data exchange and increased network throughput for FTRTA. MIQRP has been successful in supporting the reliability of data exchanges, but this method has not managed traffic and congestion, which has led to decrease throughput, especially in scenarios with heavy traffic. RPL is a simple routing protocol and is primarily designed for use in low density networks.

This has led to a drop in throughput, especially in scenarios with heavy traffic.

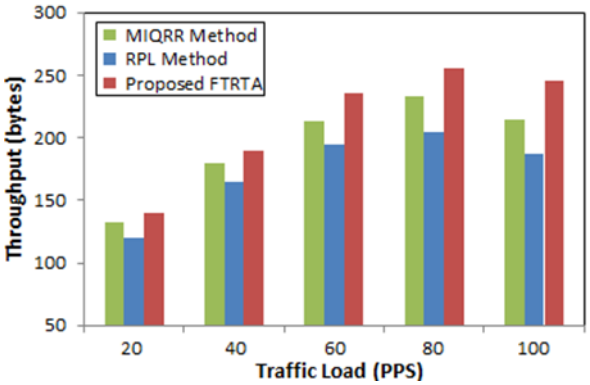

Figure. 12 Network throughput by changing the traffic load

6. Conclusion and future works

In this paper, a different method called FTRTA is introduced to enhance the reliability and fault tolerance of data exchange in IoT. FTRTA is based on optimizing the RPL protocol and utilizing data distribution techniques, aiming to improve the reliability of routing support with fault tolerance. The FTRTA is implemented to evaluate software based on the Cooja simulator software, and the results of its performance indicate an increase in successful network receipts, improved stability of intermediate routes, and increased network throughput compared to similar methods. However, while FTRTA has performed remarkably well in improving fault tolerance, it is ineffective in supporting the quality of routing and data exchange. Therefore, in future research, we will attempt to address this issue by evaluating qualitative criteria in addition to fault tolerance.

References

[1] H. HaddadPajouh, A. Dehghantanha, R. M. Parizi, M. Aledhari, and H. Karimipour, A survey on internet of things security: Requirements, challenges, and solutions, *Internet of Things*, Vol. 14, pp. 100129, Jun, 2021.

[2] R. Hassan, F. Qamar, M. K. Hasan, A. H. M. Aman, and A. S. Ahmed, Internet of Things and its applications: A comprehensive survey, *Symmetry*, Vol. 12, no. 10, pp. 1674, Oct, 2020.

[3] Huang, Haiping, et al. An efficient signature scheme based on mobile edge computing in the NDN-IoT environment, *IEEE Transactions on Computational Social Systems*, Vol. 8, no. 5, pp. 1108-1120, May, 2021.

[4] M. Al-Emran, S. I. Malik, and M. N. Al-Kabi, A survey of internet of things (IoT) in education: opportunities and challenges, *Toward social internet of things (SIoT): Enabling technologies, architectures and applications*, pp. 197-209, 2020.

[5] L. Xing, Reliability in Internet of Things: Current status and future perspectives, *IEEE Internet of Things Journal*, Vol. 7, no. 8, pp. 6704-6721, May, 2020.

[6] K. Gulati, R. S. K. Boddu, D. Kapila, S. L. Bangare, N. Chandnani, and G. Saravanan, A review paper on wireless sensor network techniques in Internet of Things (IoT), *Materials Today: Proceedings*, 2021.

[7] B. Pourghebleh, V. Hayyolalam, and A. A. Anvigh, Service discovery in the Internet of Things: review of current trends and research challenges, *Wireless Networks*, Vol. 26, no. 7, pp. 5371-5391, May, 2020.

[8] Y. B. Zikria, M. K. Afzal, F. Ishmanov, S. W. Kim, and H. Yu, A survey on routing protocols supported by the Contiki Internet of things operating system, *Future Generation Computer Systems*, Vol. 82, pp. 200-219, May, 2018.

[9] A. J. Dey, and H. K. D. Sarma, Routing Techniques in Internet of Things: A Review, *Trends in Communication, Cloud, and Big Data*, PP. 41-50, 2020.

[10] J. Nassar, M. Berthomé, J. Dubrulle, N. Gouvy, N. Mitton, and B. Quoitin, Multiple instances QoS routing in RPL: Application to smart grids, *Sensors*, Vol. 18, no. 8, pp. 2472, Jul, 2018.

[11] A. A. Kadhim, and S. A. Rafea, Routing with Energy Threshold for WSN-IoT Based on RPL Protocol, *Iraqi J. Comput. Commun. Control Syst. Eng*, Vol. 19, no. 1, pp. 71-81, Mar, 2019.

[12] T. L. Jenschke, R. A. Koutsiamanis, G. Z. Papadopoulos, and N. Montavont, Multi-path selection in RPL based on replication and elimination, *International Conference on Ad-Hoc Networks and Wireless*, pp. 15-26, Sep, 2018.

[13] M. Conti, P. Kaliyar, and C. Lal. A robust multicast communication protocol for Low power and Lossy networks, *Journal of Network and Computer Applications*, pp. 102675, Aug, 2020.

[14] T. Muhammed, R. Mehmood, A. Albeshri, and A. Alzahrani, HCDSR: A hierarchical clustered fault tolerant routing technique for IoT-based smart societies, *Smart Infrastructure and Applications*, pp. 609-628, 2020.

[15] P. Sanmartin, A. Rojas, L. Fernandez, K. Avila, D. Jabba, and S. Valle, Sigma routing metric for RPL protocol, *Sensors*, Vol. 18, no. 4, pp. 1277, Apr, 2018.

[16] M. Lazarevska, R. Farahbakhsh, N. M. Shakya, and N. Crespi, Mobility Supported Energy Efficient Routing Protocol for IoT Based Healthcare Applications, *IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 1-5, Oct, 2018.

[17] M. Bouaziz, A. Rachedi, A. Belghith, M. Berbineau and S. Al-Ahmadi, EMA-RPL: Energy and mobility aware routing for the Internet of Mobile Things, *Future Generation Computer Systems*, Vol. 97, pp 247-258, Aug, 2019.

[18] T. Winter, et al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, *rfc*, Vol. 6550, pp. 1-157, Mar, 2012.

[19] S. Sennan, S. Balasubramaniyam, A. K. Luhach, S. Ramasubbareddy, N. Chilamkurti, and Y. Nam, Energy and Delay Aware Data Aggregation in Routing Protocol for Internet of Things, *Sensors*, Vol. 19, no. 24, pp. 5486, Dec, 2019.

[20] N. Sousa, J. V. Sobral, J. J. Rodrigues, R. A. Rabêlo and P. Solic, ERAOF: A new RPL protocol objective function for Internet of Things applications, *2nd*

International Multidisciplinary Conference on Computer and Energy Science (SpliTech), pp. 1-5, Jul, 2017.
[21] B. Vaziri, and A. T. Haghighat, Brad-OF: An Enhanced Energy-Aware Method for Parent Selection and Congestion Avoidance in RPL Protocol, *Wireless Personal Communications*, pp. 1-30, 2020.
[22] S. Hoghooghi, and R. N. Esfahani, Mobility-Aware Parent Selection for Routing Protocol in Wireless Sensor Networks using RPL, *5th International Conference on Web Research (ICWR)*, pp. 79-84, Apr, 2019.
[23] K. Jaiswal, and V. Anand, EOMR: An Energy-Efficient Optimal Multi-path Routing Protocol to Improve QoS in Wireless Sensor Network for IoT Applications, *Wireless Personal Communications*, Vol. 111, no. 4, pp. 2493-2515, Apr, 2020.
[24] S. K. Preeth, R. Dhanalakshmi, R. Kumar, and S. Si, Efficient parent selection for RPL using ACO and coverage based dynamic trickle techniques, *Journal of Ambient Intelligence and Humanized Computing*, Vol. 11, no. 11, pp. 4377-4391, Nov, 2020.
[25] S. Sankar, and P. Srinivasan. Fuzzy logic based energy aware routing protocol for Internet of Things, *International Journal of Intelligent Systems and Applications*, Vol. 10, no. 10, pp. 11, Oct, 2018.
[26] P. Singh, and Y. C. Chen, RPL Enhancement for a Parent Selection Mechanism and an Efficient Objective Function, *IEEE Sensors Journal*, Vol. 19, no. 21, pp. 10054-10066, Jul, 2019.
[27] T. D. Nguyen, J. Y. Khan, and D. T. Ngo, A distributed energy-harvesting-aware routing algorithm for heterogeneous IoT networks, *IEEE Transactions on Green Communications and Networking*, Vol. 2, no. 4, pp. 1115-1127, May, 2018.
[28] "IEEE Draft Standard for Local and Metropolitan Area Networks Part 15.4: Low Rate Wireless Personal Area Networks (LR-WPANs) Amendment to the MAC sub-layer," IEEE P802.15.4e/D 6.0 (Revision of IEEE Std 802.15.4-2006), pp. 1–200, Aug, 2011.
[29] L. Wallgren, R. Shahid, and V. Thiemo, Routing attacks and countermeasures in the RPL-based internet of things, *International Journal of Distributed Sensor Networks*, Vol. 9, no. 8, pp. 794326, Aug, 2013.
[30] J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, A. Terzis, A. Dunkels and D. Culler, ContikiRPL and TinyRPL: Happy Together, *Proceedings of the workshop on Extending the Internet to Low power and Lossy Networks (IPSN)*, April 12-14, 2011.
[31] A. Dunkels, J. Eriksson, N. Finne and N. Tsiftes, Powertrace: Network-level power profiling for low-power wireless networks, 2011.
[32] R. M. Srinivasa, and D. N. Rao. "Faulty Nodes Detection for Reliable Data Transmission in Intelligent Wireless Sensor Networks." *International Journal of Intelligent Engineering & Systems* 17, no. 2024.
[33] V. Mohammadi, A. M. Rahmani, A. Darwesh, and A. Sahafi. "Fault tolerance in fog-based Social Internet of Things." Knowledge-Based Systems vol. 265 pp. 110376, 2023.
[34] Z. Qinbin, T. Zhao, X. Chen, Y. Zhong, and H. Luo. "A fault-tolerant transmission scheme in SDN-based industrial IoT (IIoT) over fiber-wireless networks." *Entropy* 24, no. 2, pp. 157, 2022.

Mohsen Mozafari Vanani received the B.Eng. degree from Islamic Azad University, Shahrekord branch, Iran, in 2020 and the M.S. degree from Islamic Azad University, Shahrekord branch, Iran, in 2021. Also since 2022, he is a Ph.D. student at Islamic Azad University, Central Tehran branch, Iran. His Master's thesis with subject Improving Loading communications and exchanges with an optimization approach and the use of ant colony algorithm and with an excellent score It has been judged.

Pouya Khosravian Dehkordi received the B.Eng. degree from Islamic Azad University, Najafabad branch, Iran, in 2005 and the M.S. degree from Islamic Azad University, Arak branch, Iran, in 2008. Since 2009, he is a faculty member of Islamic Azad University, Shahrekord branch, Iran. Also since 2014, he is a Ph.D. student at Islamic Azad University, Yazd branch, Iran. His Ph.D. thesis deals with Service Function Chaining. His current research interests include Software Defined Networks, Service Function Chaining, Natural Language Processing, and Automata Theory.

Journal of Optimization of Soft Computing (JOSC)

Vol. 1, Issue 2, pp: (39-45), December-2023

Journal homepage: https://sanad.iau.ir/en/Journal/josc

Research paper

An Intrusion Detection System for Network Cyber Security Using Hybrid Feature Selection Algorithms

Zahraa Oday Kamil[1], Golnaz Aghaee Ghazvini[2,*]

1Department of Computer engineering, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran
2Department of Computer engineering, Dolatabad Branch, Islamic Azad University, Isfahan, Iran.

Article Info

Article History:
Received: 2024/2/29
Revised:
Accepted: 2024/4/1

Keywords:
intrusion detection; Minimum Redundancy Maximum Relevance; Gray Wolf Optimization; Support Vector Machines.

**Corresponding Author's Email Address: golenaz.aghaee@gmail.com*

Abstract

One of the most important challenges of the expansion of the Internet and virtual space is cyber-attacks. These attacks are becoming new every day and it is becoming more difficult to deal with them. As a result, methods should be used to detect them, which can detect all types of cyber-attacks in the shortest possible time and with proper accuracy. Nowadays, machine learning methods are usually used to detect cyber-attacks. But since the data related to cyber-attacks have many characteristics and are kind of bulky data, as a result, the accuracy of conventional machine learning methods to detect them is usually low. In this research, we have used a hybrid feature selection method to select optimal features from the database related to cyber-attacks, which increases the accuracy of attack detection by classification models. In the proposed feature selection method, first the features that have the least redundancy with each other and at the same time are most related to the category variables (labels) are selected by the MRMR algorithm. Then, using a wrapper feature selection method based on the gray wolf optimization (GWO) algorithm to select a subset of the features selected from the previous step, which maximizes the accuracy of the SVM classifier model, is used this subset has optimal features by which the SVM model is trained. As a result, the accuracy of detecting cyber-attacks by the SVM model increases. According to the simulation results, the average accuracy of the proposed method for detecting cyber-attacks is 99.84%, which has improved compared to the intrusion detection methods of the reference article.

1. Introduction

The military, economy, social sector, and so on now all rely heavily on computer networks as essential tools. It guarantees the communication, coordination, and cooperation between these many industries. Naturally, the number of users rises in tandem with the remarkable improvements of networks. Numerous benefits resulted from the internet's wide accessibility and massive rise in computer network usage. However, the rise in computer hacking has become a serious problem. Various security solutions, including firewalls, antivirus software, internet security tools, and network intrusion detection systems (IDS) are designed to protect computer servers and clients globally against anomalous invasions. On social networks, users who are recognized and/or anonymous may not always have the best of intentions. They are able to take advantage of system and network weaknesses. Additionally, they have access to private or sensitive data that they can read, edit, or destroy. Therefore, in order to prevent prospective assaults on networks, the act of network security has become increasingly important [1].

IDS has been divided into two groups: anomaly behavior detection and signature-based anomaly detection. These two kinds differ from one another in their patterns. The signature-based intrusions periodically scan the network and attempt to match certain specified patterns there. On the other hand, anomalous network intrusion-based systems present typical traffic patterns and search for packets that are similar enough to be identified as intrusions [2].

As stated in [3], anomaly detection, sometimes referred to as undesired persuaded defects, assaults, or faults, is the essential component of intrusion detection.

By dividing packets into two categories—attacks and normal—IDS aims to stop intrusions from occurring in subsequent network transactions. Computational speed and comparison accuracy are two of the most crucial factors in the IDS. Given the abundance of characteristics in every transaction, an appropriate technique is needed to identify both the incursions and the complete feature set by deriving an effective subset of features.

Finding a subset of features (M) from an original set (N) where M<N is the process of feature extraction. Cropping features with more relevant information and those that include data about other features is the aim of feature selection. Moreover, the process time is being decreased with minimal accuracy loss by removing some of the unnecessary or redundant elements. In order to accomplish this goal, a few data mining techniques have been presented and used thus far for feature extraction [4].

Two techniques for feature selection in IDS are wrapper-based and filter-based methods. The wrapper-based strategy leverages machine learning techniques to obtain dependable intrusion detection features. Whereas the filter approach, which minimizes and effectively uses the features that are redundant or irrelevant, never makes use of machine learning [4].

The following describes the current study's structure: An overview of the literature on the application of wrapper and filter-based methods for creating effective NIDS is given in Section 2. A synopsis of the suggested model is provided in Section 3. The findings and discussion are presented in Section 4. Regarding Section 5, it offers the findings.

2. Related works

The Intrusion Detection System has been the subject of numerous prior studies that have been published in the literature. The intrusion detection system's abstract model was first put forth by Denning D.E. [5] in 1987. Intrusion detection is the first security defensive method for the computer system used in this paper. The concept is not dependent on any particular application environment, operating system, or type of intrusion or vulnerability in the system. It is a framework that can serve as a great illustration of how to create application systems for intrusion detection. Though other unidentified reasons that are not anomalous behaviors can also trigger the audit criteria in the proposed model. Furthermore, it remains to be demonstrated whether the approach can identify the greatest intrusion before significant harm is caused. In addition to focusing primarily on anomaly detection using data mining, Wu et al. [6] apply association rules in a forward implementation based on Trie trees. Aumreesh et al.'s review [7] highlights the several kinds of intrusion detection systems, including host-based, network-based, misuse-based, anomaly-based, and hybrid-based systems. It primarily concentrates on agent-based, behavior-based, and anomaly-based technologies in real network traffic. The advantages and disadvantages of the abuse detection strategy and the anomaly detection approach are compared by S. Northcutt et al. [8]. The disadvantage of the anomaly detection approach, as the author notes, is that the Intrusion Detection System may trigger a false positive alarm when it detects a new behavior for the first time. A review of using Machine Learning (ML) technologies in the Intrusion Detection System is provided by L. Haripriya and M.A. Jabbar [9].
Additionally, the rates of false negatives, false positives, and anomaly detection are comparatively considerably higher than the rates of misuse detection. Additionally, they go over how to apply machine learning to a system and provide a thorough comparison of different methods for an intrusion detection system that use machine learning. According to this paper, it can be challenging to train machine learning models when there is a shortage or unavailability of traffic data. Basant Subba et al. [10] provide an effective Artificial Neural Network (ANN) model for an intrusion detection system. A constraint of their methodology is that the suggested model necessitates an extensive training duration. Nevertheless, the inability to add more agents to the prior one won't affect the neural network's overall detection performance. The most popular

feature selection algorithm, Filter and wrapper, is described by Pan-Shi Tang et al. [11] in their work. The Genetic Algorithm-based selection method is also applied to a combination of two algorithms, and the results show that GA is significantly more efficient in picking features than the Filter and Wrapper algorithms. S. Aksoy et al. [12] and B. Kavitha et al. [13] provide a crucial technique for employing the Genetic Algorithm to choose the necessary subset of features. They think feature selection can eliminate unnecessary elements and significantly impact the development of an effective classification system in subsequent stages. Ketan Sanjay Desale and Roshani Ade [14] present a novel approach to feature selection utilizing a genetic algorithm and the mathematical intersection principle. Additionally, a variety of feature selection methods are examined, including IG, CAE, and CFS. Their results with J48 and Naive Bayes (NB), the other two commonly used classifiers, are compared. These articles provide a nice illustration of how to use a genetic algorithm to choose features.

3. Proposed method

Our goal in this work is to provide a high-precision system in order to increase cyber security in the network by using feature selection hybrid algorithms. As a result, in this work, we will use two algorithms - MRMR and Gray Wolf Optimization (GWO) in order to select an optimal subset of features. In this research, first, by using the MRMR algorithm, which uses the concepts of maximum similarity and minimum redundancy, the features with the least redundancy are selected. Then, with the help of the gray wolf optimization algorithm, an optimal subset of the features selected by the MRMR algorithm will be selected. The gray wolf optimization algorithm, which is based on the instinctive behavior of gray wolves, is a form of meta-heuristic algorithm with a hierarchical structure that is inspired by the hunting activity of gray wolves. This population-based approach is simple in its operation and can be easily extended to situations with different dimensions. Also, after choosing the optimal features, these features are classified by the support vector machine algorithm and all types of cyber threats are detected. With the help of mapping the feature space to the dimension with higher resolution, the support vector machine has a large capacity to recognize patterns and thus recognize attack patterns. This enables the SVM algorithm to accurately detect network intrusions. The diagram of the proposed method is shown in Figure (1).

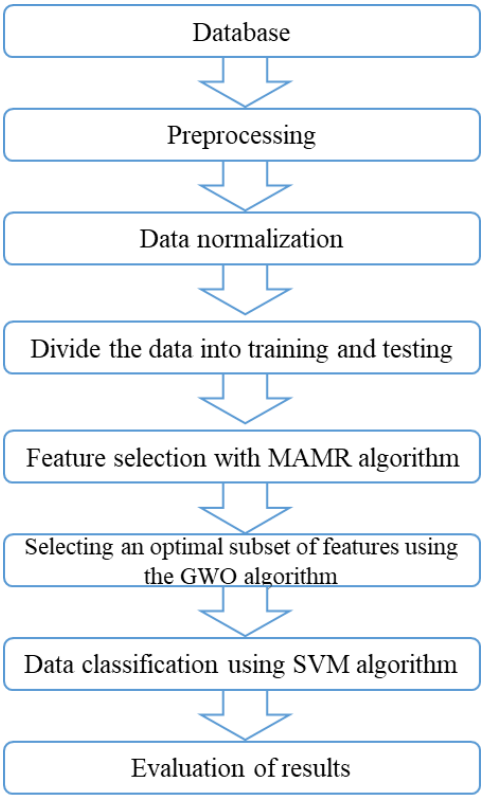

Figure 1: Diagram of the proposed method

4. Steps of the proposed method

According to the diagram of the proposed method, the basic steps in the proposed method for detecting different types of cyber-attacks include pre-processing and dividing the data, selecting the combined feature and classifying the selected features. These steps are explained below.

4.1. Data preprocessing

First, the KDD database used in this research is uploaded to the software, and then the first step is to apply the necessary pre-processing on the data. Some of the necessary pre-processing are outlier data removal, missing data removal and data normalization. In this research, the Max-Min technique was used to normalize the data. Data normalization ensures that all features are in a certain range (for example -1 and 1) and as a result, the effect of all features in the classification is the same. The normalization of the Max-Min method is defined by (equation 1).

$$X_n = \frac{X_i - X_{min}}{X_{max} - X_{min}}(H - L) + L. \quad i = 1.2 \ldots . N \qquad (1)$$

In equation (1), X_i is the actual value of a feature for the network input and $X_n$is the normalized expression of X_i. X_{min} and

X_{max}represent the minimum and maximum X values. The lower and upper limits of normalization are also indicated by L and H and have values of -1 and +1, respectively.

4.2. Dividing the data into two groups of training and testing

Title must be in 15 pt Regular font. Author name Since in the proposed method, a supervised machine learning model is used for data classification, as a result, training and test data are needed to train and evaluate the performance of the proposed model. In this research, we have divided the database samples into two groups of training and testing with a ratio of 66 to 34. The stages of feature selection and SVM model training are first completed by the training samples and then the proposed model is evaluated on the test samples.

4.3. feature selection by MRMR-GWO method

As stated before, the basis of the proposed method in this research is a combined feature selection technique, which improves the performance of the machine learning model in detecting cyber-attacks. Feature selection methods are classified into two general categories, the first category is filter methods and the second category is wrapper methods. The methods that perform feature selection without considering the classification model and only by considering the features and labels are filter methods and in contrast to the methods that select the features by considering They carefully choose the classification model; they are Wrapper methods. In this research, we have used the MRMR algorithm as a filter method and the GWO algorithm as a wrapper method. The diagram of feature selection method in this research is shown in figure (2).

In the proposed feature selection method, the features that are most related to the category variable (label) and at the same time have the least redundancy between the features are selected. In fact, in the MRMR algorithm, the relationship between features with data labels and the redundancy between features are calculated based on mutual information and according to relations (2) and (3).

$$W_I = \frac{1}{|s|^2}\sum_{i,j\in s} I(i,j) \quad (2)$$

$$V_I = \frac{1}{|s|}\sum_{i\in s} I(h,i) \quad (3)$$

In the above relations, S is a specific set of attributes and h is a class variable. Also, W_I is the redundancy between features and V_I is the relation of S with class variable h. Mutual information between two variables $I(x,y)$ is also calculated by the following equation:

$$I(x,y) = \sum_{i,j} \log \frac{p(x_i,y_j)}{p(x_i)p(y_j)} \quad (4)$$

As a result, first all the features are ranked by the MRMR algorithm and a number of those that are most related to the category variable and have the least redundancy among themselves are selected.

In the following, the relationship between the features selected by the MRMR algorithm and the classifier model, which is the SVM algorithm in this research, is examined by a Wrapper method, and the best features that lead to the highest accuracy in the SVM algorithm are selected. In this research, GWO meta-heuristic algorithm has been used to investigate the relationship between features and the classifier model.

The gray wolf algorithm works to select the final feature by first defining a feature space with the dimensions of the features selected from the MRMR algorithm and generating feature vectors with specific dimensions where each wolf is a representative of a feature vector. The wolves move in the feature space and their information is updated according to the objective function, and the quality of each feature vector is evaluated. In this technique, the objective function in the gray wolf algorithm is the accuracy of the classification model i.e. SVM. As a result, the extraction of the features that lead to the maximum of the objective function or the accuracy of the SVM algorithm is selected as the optimal feature vector.

Briefly, in the gray wolf algorithm, it starts from a random population of wolves with random feature vectors, and then these wolves move randomly in the feature space and update the improved feature vectors. In this way, the alpha wolf keeps the best feature vector and the other wolves replace the alpha wolf if the feature vector improves. This algorithm is used for feature selection in various problems due to its fast convergence capability and no need for many parameters.

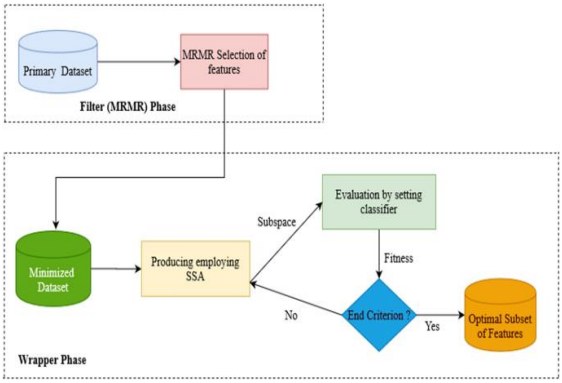

Figure 2: Diagram of feature extraction method.

4.4. Classification using SVM algorithm

In this research, the classifier model, which the wrapper feature selection technique is also performed according to, is the SVM algorithm. SVM is an effective method to build a classifier. Its purpose is to create a decision boundary between two classes to allow prediction of labels from one or more vectors. This decision boundary is called a hyperplane and is oriented so that it is farthest from the nearest data points of each class. These closest points are called support vectors. Assuming we have a labeled estimator dataset:

$$(x_1, y_1 \dots \dots (x_n \cdot y_n) \cdot x_i \in R^d \text{ and } y_i \in (-1.+1) \quad (5)$$

where x_1 represents the feature vector and y_i is the class label (negative or positive) of estimator combination i. Therefore, the desired hyperplane is defined as follows.

$$wx^T + b = 0 \quad (6)$$

where w is the weight vector, x is the input feature vector, and b is the orientation. w and b satisfy all the following inequalities for all components of the estimator set:

$$wx_i^T + b \geq +1 \text{ if } y_i = 1 wx_i^T + b \leq -1 \text{ if } y_i = -1 \quad (7)$$

The purpose of SVM model estimation is to find w and b so that it separates the data super plane and maximizes the boundary $1/||w||2$. Hence, vectors x_i with $(wx^T + b) = 1$ y_i are called support vectors.

4.5. Evaluation of the proposed model

After selecting the best features and training the network using training data, the last step is dedicated to testing the proposed network. This step is done with the help of test data that do not have output labels. In this step, the test data is entered into the classifier model and the accuracy of the trained network is evaluated to detect different types of cyber-attacks on this data.

5. Simulation and Results

In this research, the two-class NSL-KDD dataset has been used. This database is the updated version of the KDDCup99 database that was presented in 2009 in order to categorize different cyber-attacks. NSL-KDD bank consists of data related to normal mode and attack mode. This data bank is registered in two different groups, consisting of KDDTrain+, KDDTest+, and KDDTrain set is used for network training, and KDDTest set is used to evaluate the proposed technique. In order to evaluate the results of the proposed method, the following criteria are used.

$$Accuracy(acc) = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$Precision = TP(TP + FP) \quad (9)$$

$$Recall = TP/(TP + FN) \quad (10)$$

$$F1\ score = \frac{2 * (Recall * Precision)}{(Recall + Precision)} \quad (11)$$

In the above relations, TP represents the number of correct identification of attacks, TN is the number of correct identifications of normal traffic, FP is the number of false identification of attacks, and FN is the number of false identifications of normal traffic.

5.1. Simulation settings

As stated in the proposed method chapter, we have used a two-stage hybrid feature selection technique. The NSL-KDD database has 41 features. In the simulation, we first ranked the features by the MRMR algorithm and selected the top 20 features. Then, in the next step, 8 optimal features from the 20 initially selected features have been selected by the gray wolf algorithm and entered into the SVM algorithm for classification. Also, in the simulation, we have considered 10,000 samples for training the SVM algorithm and 5,000 samples for testing the proposed model. Finally, Table (1) presents the parameters of the gray wolf algorithm for feature selection using the wrapper method.

Table 1: Basic parameters of the gray wolf algorithm

the amount of	parameter
50	The number of repetitions
20	population size
[20-1]	The allowed interval for changing variables
accuracy	The objective function

5.2. Simulation results

In this section, the results for the test data are presented. As stated earlier, the labels of this part of the data have not been seen before by the proposed algorithm, and the proposed method should predict the label of each sample from the optimal features selected by the MRMR-GWO method. Cyber-attack or normal traffic. Figure (3) shows the confusion matrix on the test data. As can be seen, there are 2145 samples in the first row of the confusion matrix, which are samples related to cyber-attacks, and in the second row, there are 2855 samples, which are samples related to normal traffic. In both groups, only 2 samples were misdiagnosed. The overall accuracy of the proposed method for detecting examples of cyber-attacks and normal traffic is 99.9%.

Figure 3: Confusion matrix for test data

Figure (4) also shows the diagram related to the numerical values of the evaluation criteria. The exact values of the criteria of precision, accuracy, recall and F-score are equal to 99.88% according to figure (4). These results are for one simulation run.

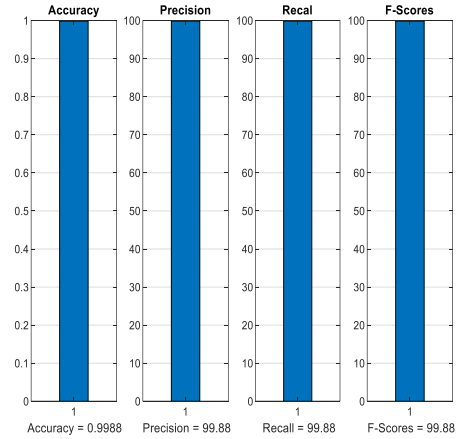

Figure 4: Numerical values of evaluation criteria on test data

Also, the ROC curve related to the test data can be seen in Figure (5). In this graph, the true positive rate (TPR) is plotted against the false positive rate (FPR) at different classification thresholds. In the ROC diagram, the closer the line of a class is to the intentional axis and the farther it is from the minor diameter, the more accurate it is in recognizing the samples of that class. In figure (5), it is clear that the accuracy of detecting samples of both classes is equal.

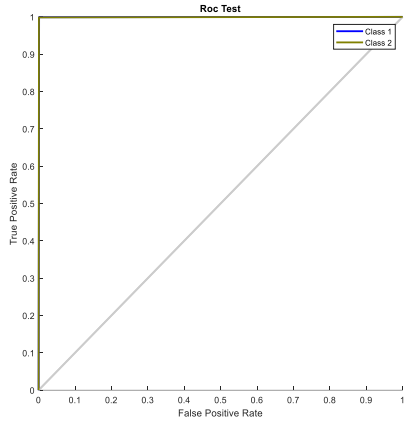

Figure 5: ROC curve on test data

5.2.1. Comparison of results

Finally, in this part, we have compared the results of the proposed method for detecting cyber-attacks with other methods. The simulation results are presented in terms of intrusion detection accuracy, and the important point is that the results of the proposed method are the result of the average of 40 simulation program executions. As it is known, the proposed method has an average accuracy of 99.84%, which has improved compared to other methods in the reference article.

Table 2: Comparison of the proposed method with other methods presented in the reference article [100]

Method	accuracy
Bayesian network	85.76
J48	96.43
SMO	95.99
Reference Article Method (NB-GA) [١٠٠]	99.73
suggested method (MRMR-Bagging)	99.84

Based on the simulation results, the average accuracy of the proposed model for detecting cyber-attacks is 99.84%, which is higher than the

compared methods. This high accuracy in the proposed method is due to the two-stage feature selection method, which makes the classifier model to be trained with the best features and the relationship between the features with the classification variable in the best way for The SVM model is specified.

6. Conclusion

In this research, a method based on machine learning to detect cyber-attacks is presented. A combined feature selection technique is used in the proposed method. This technique has the ability to select the best features, which leads to increase the accuracy of the classifier model, among all the features of the database. In this proposed method, we first rank the features of the NSL-KDD database by the MRMR algorithm and select 20 of the features that are most related to tags and at the same time have the least redundancy with other features. have, we choose. Then, among the 20 selected features, we select a subset of 8 features that maximize the accuracy of the SVM algorithm as optimal features. Then the training samples with these features are entered into the SVM algorithm and the learning process is completed.

Reference

[1] Folorunso, O., O.O. Akande, A.O. Ogunde and O.R. Vincent 2010. ID-SOMGA: A self organising migrating genetic algorithm-based solution for intrusion detection. Comput. Inform. Sci., 3: 80-92.

[2] Trair, D., W. Ma, D. Sharma and T. Nguyen, 2007. Fuzzy vector quantization for network intrusion detection. Proceedings of the IEEE International Conference on Granular Computing, Nov. 2-4, IEEE Xplore Press, Fremont, CA., pp: 566-566. DOI: 10.1109/GrC.2007.124

[3] Lazarevic, A., L. Ertoz, V. Kumar, A. Ozgur and J. Srivastava, 2003. A comparative study of anomaly detection schemes in network intrusion detection. Proceedings of the 3rd SIAM International Conference on Data Mining, (CDM' 03), SIAM.

[4] Chen, Y., Y. Li, X.Q. Cheng and L. Guo, 2006. Survey and Taxonomy of Feature Selection Algorithms in Intrusion Detection System. Proceedings of the 2nd SKLOIS conference on Information Security and Cryptology, Nov. 29-Dec. 1, Springer Berlin Heidelberg, Beijing, China, pp: 153-167.

[5] D. E. Denning, "An intrusion detection model," IEEE Transactions on Software Engineering, vol. 13, no. 2, pp. 222-232, 1987.

[6] W. Gongxing and H. Yimin, "Design of a new intrusion detection system based on database," in Proc. 2009 International Conference on Signal Processing Systems, 2009, pp. 814-817.

[7] A. K. Saxena, S. Sinha, and P. Shukla, "General study of intrusion detection system and survey of agent-based intrusion detection system," in Proc. 2017 International Conference on Computing, Communication and Automation (ICCCA), 2017, pp. 421-471.

[8] S. Northcutt and J. Novak, "Network intrusion detection," IEEE Network, vol. 8, no. 3, pp. 26-41, 2003.

[9] L. Haripriya and M. A. Jabbar, "Role of machine learning in intrusion detection system: Review," in Proc. 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2018, pp. 925-929.

[10] M. B. Subba, S. Biswas, and S. Karmakar, "A neural network-based system for intrusion detection and attack classification," in Proc. 2016 Twenty Second National Conference on Communication (NCC), 2016, pp. 1-6.

[11] P. S. Tang, X. L. Tang, and Z. Y. Tao, "Research on feature selection algorithm based on mutual information and genetic algorithm," in Proc. 2014 11th International Computer Conference on Wavelet Active Media Technology and Information Processing, 2014.

[12] S. Aksoy, "Feature reduction and selection," Department of Computer Engineering, Bilkent University, 2008.

[13] B. Kavitha, S. Karthikeyan, and B. Chitra, "Efficient intrusion detection with reduced dimension using data mining classification methods and their performance comparison," in Proc. International Conference on Business Administration and Information Processing, 2010, pp. 96-101.

[14] K. S. Desale and R. Ade, "Genetic algorithm-based feature selection approach for effective intrusion detection system," in Proc. 2015 International Conference on Computer Communication and Informatics (ICCCI), 2015, pp. 1-6.

Journal of Optimization of Soft Computing (JOSC)

Vol. 1, Issue 2, pp: (46-53), December-2023

Journal homepage: https://sanad.iau.ir/en/Journal/josc

Research paper

Comparison optimization Computational model between Cellular Automata and Genetic programming in dynamic response of guyed tower under vibration force

Kaveh kumarci[1]

*1. **College of skills and entrepreneurship, Islamic Azad University, Shahrekord branch, Shahrekord, Iran***

Article Info

Article History:
Received: 2024/3/2
Revised:
Accepted: 2024/4/3

Keywords:
guyed tower, Cellular automata, Genetic programming, optimization, vibration force

**Corresponding Author's Email Address:kumarci_kaveh@yahoo.com*

Abstract

In the telecommunication industry, guyed towers are one of the important structural subsystems. They support a variety of antenna systems at great heights to transmit radio, television and telephone signals over long distance, thus preserving them in events of natural disasters such as earthquake is of high priority. Also, domes and transmission stations functions depend on transmitted information by guyed towers. In this paper, seismic behavior of guyed towers are studied. For that, one guyed tower in 9 clusters of guys is studied under earthquake force. This research was accomplished on the base of wind and earthquake forces and tower interaction to these forces. Here, the effect of earthquake force and tower response to seismic events are studied. At first, time history analysis is used in determination of towers vibration natural modes, then, under time- acceleration components of El-Centro earthquake, spectral analysis are accomplished. Analysis outputs are two parameters including frequency and maximum lateral displacement which are provided using ANSYS software. The results are used in comparing two different calculation models: genetic programming and cellular automata.

1. Introduction

Cellular automata (CA) is a decentralized computing model providing an excellent platform for performing complex computation with help of only local information [1]. In this paper, using 1- D two and three state in CA some samples of towers are studied to find rules of simulation for frequency and maximum lateral displacement of towers[2]. For that, using provided algorithms of two and three state 1-D CA and using 5 parameters- horizontal state between guyed cables on the earth (X1), the height of the first tower guy and the earth (x2), the height of guy cable and tower (x3), the height of antenna on the tower (x4) and the number of guy levels surface (n) and maximum deflection- for 50 towers in programmed CA (C++ programming language) and after one billion accomplishments for two state rules and 20 million three state rules, some models were provided for each tower. These models were tested to other 50 tower sample and in terms of the error percent for frequency and maximum lateral deflection, there was comparison with obtained results via ANSYS software [3].

Genetic programming (GP) is an evolutionary algorithm-based methodology which is branch of genetic algorithms [4]. Genetic programming creates computer programs in scheme computer languages as the solution. Genetic algorithms create a string of numbers that represent the solution so they have ability to optimize complex structures and can use in different problems [5]. GP model was provided for 50 guyed towers on the basis of X1, X2, X3, X4 and n parameters which are provided using ANSYS software. Then using genetic operators such as generation, crossover and mutation, the final population was provided which

is maximum lateral displacement in guyed towers [6]. Finally, genetic programming algorithm has been introduced to dynamic response of guyed towers under earthquake force.

2. Materials And Methods

2.1. Guyed Tower

Guyed Towers are lightweight to heavyweight towers supported by guy wires and are designed with the ability to carry light to heavy antenna loads. Guyed Towers are typically made of a mast in triangular cross-section and hinged support[7]. A cluster of guy lines are used at various elevations and angles to tower shaft. These guy lines maintain the stability of tower and provide tower (MAST) lateral stiffness. The various components of a typical tall guyed tower are shown in Figure1. The structural behaviour of guyed towers is complex; this arises from significant geometric nonlinearity, in the first order, the sagging tendency of the guy cables and the interaction between the cables and the towers; and in second order, the slenderness of the mast[8].

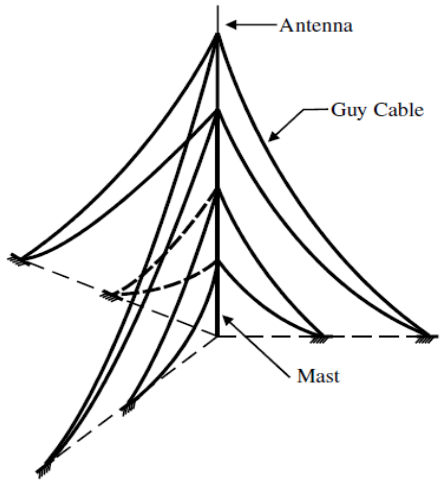

Figure 1: Guyed tower

2.2. Guyed Tower Modeling Using ANSYS

At this phase, analysis of guyed towers using ANSYS may be performed and different samples for CA and GP computing models are provided (Figure 2).

Variable modeling parameters in ANSYS software are described as follows[9]:

X1: horizontal distance between guy lines on the earth (12m)

X2: The distance from the first guy line to the base tower (11.4m)

X3: the distance between guy lines on tower (9.6m)

X4: the height of antenna on the tower (1.8m)

n: number of guy lines (9 level)

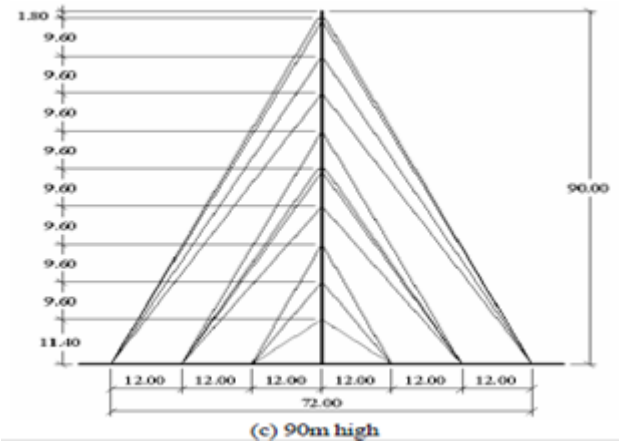

Figure 2: Geometry of 90-m guyed tower

All the above mentioned parameters are used to provide different guyed tower samples, X1: 7-12 m, X2: 11-14 m, X3: 8-11 m, X4: 1-3 m and n:7-12 m. All towers have a truss type geometry with a square cross section. Angles are connected to each other using bolt. Effective self weight in analysis includes self weight if structure, stairs, antenna, guy lines and etc. Also in modeling, the beam 3-D finite element with rigid connection is used. Vertical elements in four points of square section are solid and are 25 cm in diameter. Horizontal elements are varied by height of towers and include L 80×80×8mm, channel beams No. 300 and 120. Diagonal elements are used as truss bracing in which elements section change by tower height. They are made of thin casting pipe (No. 3.75×70mm and 3.25×76mm) and angle of 120×120×80 L mm. Young's modulus and steel materials density are 2.88×10^{6} kg/cm^2 and 7.85×10^{-3} kg/cm^3, respectively. All of angle elements (section No. 12) are made of steel ST32 and tensile strength equal 3600 kg/cm$^{3.}$ Also, section angles (No. 8) are made of steel ST37 and tensile strength equal 2400 kg/cm^3. All of cables are 40 mm in diameter and Young's modulus, ultimate strength and permissible strength are 688×1010kg/m^2, 10460kg/cm^2 and 4700 kg/cm^2, respectively.

3. Results And Discussion

3.1. Dynamic analysis of guyed tower

To find of earthquake effects on guyed towers and their maximum lateral deflection, horizontal component of north- south El- Centro earthquake (1940) has been studied in which time, maximum acceleration, maximum speed and maximum deflection are 31.98s, 0.31g, 33 cm/sec and 21.4 cm, respectively (Figure3).

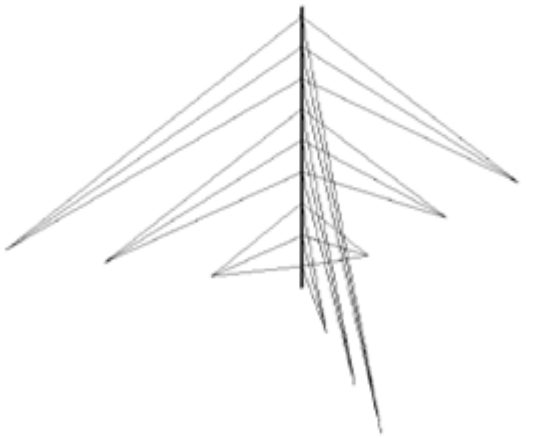

Figure 3: Accelerogram component of El- Centro earthquake

3.2. Data modeling using GP

At the second phase, using C++ programming language, the main GP steps are performed[10]:

1- Initial population of rules including potential solutions is provided potentially.
2- Provided rules are valued using fitting function in training collection.
3- Some of rules are chose on the basis of valuation of previous stage to provide reproduction mechanism.
4- Crossing over, reproduction and mutation operators are applied on current rules.
5- New generation choose to provide new generation.
6- Stages 3 to 6 will repeat to find suitable classification rule or provide maximum determined reproduction.
7- Stage 2 to 7 will repeat to find a suitable rule in data collection.
8- In training and experimental collection, each sample belong to a specific group.

These process is shown in Fig. 4.

.

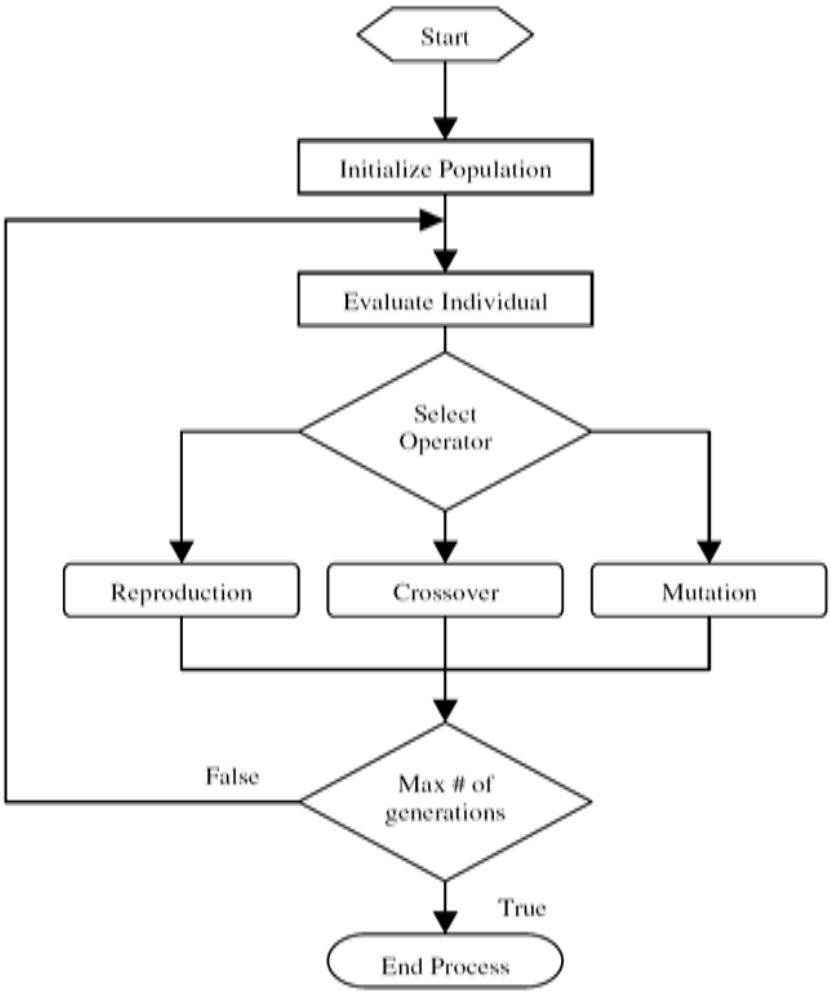

Figure 4: GP programming process

Using ANSYS, top maximum lateral displacement data (50 samples) of guyed tower were provided in which initial population are determined by X1, X2, X3, X4 and n. Also, details of mathematic formula, gen expression tree and the codes of the best GP model in top maximum lateral displacement

modeling on the basis of mathematic operators are introduced as output. Then using GP modeling programming codes in C++ programming language, other 50 analyzed samples (secondary population) are analyzed by ANSYS and the error percent of maximum lateral displacement and the best GP model is determined. Details of mathematic formula are output which determine tower maximum lateral displacement using X1, X2, X3, X4 and n.

Briefly, the second phase of paper deals with GP regarding to geometric parameters such as horizontal distance between guy lines, the height of the first guy lines on the tower, height of antenna on the tower and the number of guyed level surface. After few generations and using GP operators, final population is provided which determine tower maximum lateral displacement [11,12]. In modeling using GP, it is necessary to determine suitable frame including primary regulation (App. I) and determination of mathematic operators (App. J). At the beginning of modeling using GP, 50 guyed towers are analyzed.

4. Test of models

To test of GP models, it is necessary to provide secondary population of data which don't belong to initial population. Then the results of analysis of top lateral displacement using ANSYS and the best GP model are compared (Fig. 5 and 6).

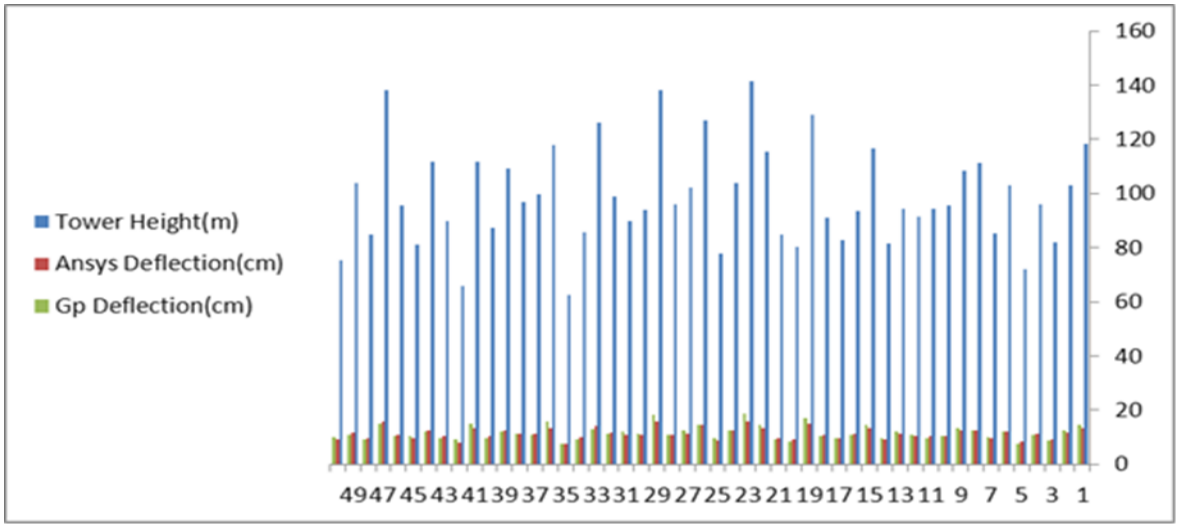

Figure 5: Comparison of maximum lateral displacement using ANSYS and GP on the basis of height of tower

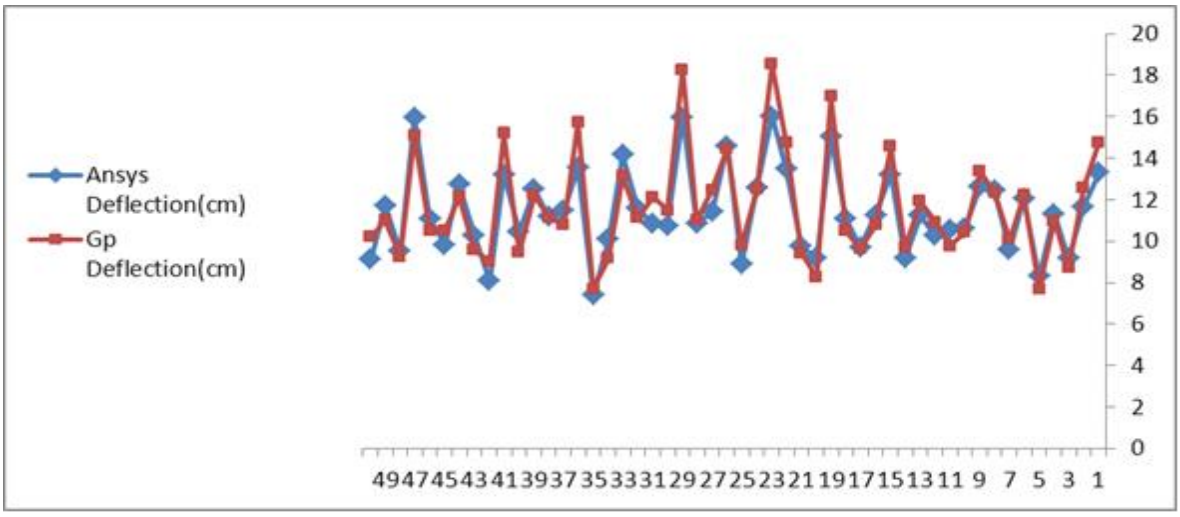

Figure 6: Comparison of maximum lateral displacement using ANSYS and GP

5. Data modeling using CA

Cellular automata (CA) are discrete, abstract computational systems[14]. The consists networks of similar and distinct places which includes finite sets of integers. According to algebraic rules, integers are depends on different steps. CA can use as an idealization of partial differential equations and usually describes natural systems. In addition, their partial natures leads to providing an important scale with the aid of digital computers. CA can be evaluated as a computer-like processing with a simple design[15,16]. CA includes two parts:

cellular space and local transition rules. Different types of each cell and cell type in time are shown with I index and S_i^t. In cellular automata, there are neighbors surrounding a central cell. For each cell, a set of cells called its neighborhood is defined relative to the specified cell. Cell neighbors (i) in time (t) and neighborhood radius of each cell are indicated in η_i^t and r, respectively[17]. Local transition rule is a function of η_i^t which is applied in cells simultaneously ($\phi(\eta_i^t)$). Finally, for each CA, there is a formula:

$$CA = (\varepsilon, d, V, \emptyset) \quad (1)$$

in which Σ, d, V and ø are Possible conditions of cell, CA dimension, CA neighborhood structure and Local transition rule, respectively.
One- dimensional CA is shown in Figure 7.

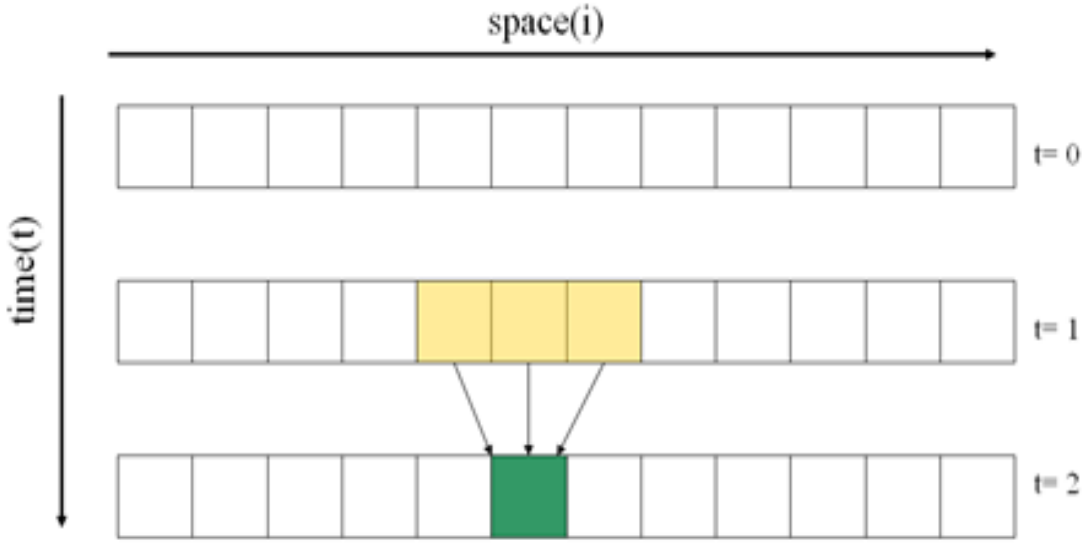

Figure 7: 1-D cellular automata

In this phase, using 1-D two and three state CA, provided tower samples in ANSYS are analyzed to find rules for simulation of maximum lateral displacement of guyed towers[18]. For that, by using 1-D two and three state CA algorithms and considering 5 parameters (X1, X2, X3, X4 and n) and calculated maximum displacement by ANSYS in C++ programming, one billion accomplishments for two state rules and 10 million rules for tree state rules, some tower models were provided. Then, these models were compared with other 50 samples and the error percent was determined for each analyzed tower in ANSYS. Analysis and modeling process by using cellular automata are shown in Figure 8.

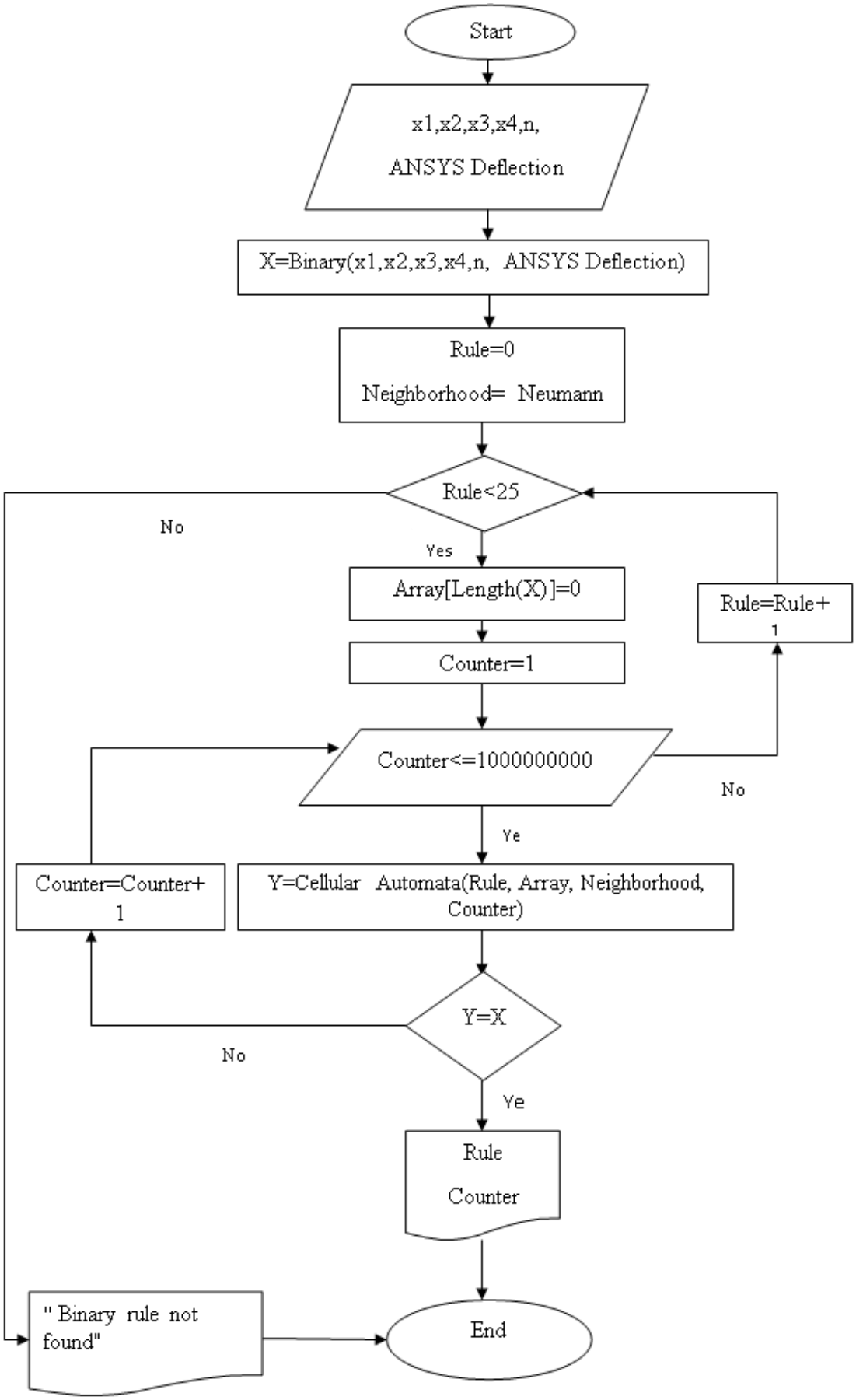

Figure 8: CA programming process

6. Analysis of guyed tower using CA

By using 50 data obtained by calculation of maximum lateral displacement in ANSYS, CA model was provided and after one billion accomplishments for 256 two state rule and one trillion performance three state rule follow data were provided. (figure 9 and 10)

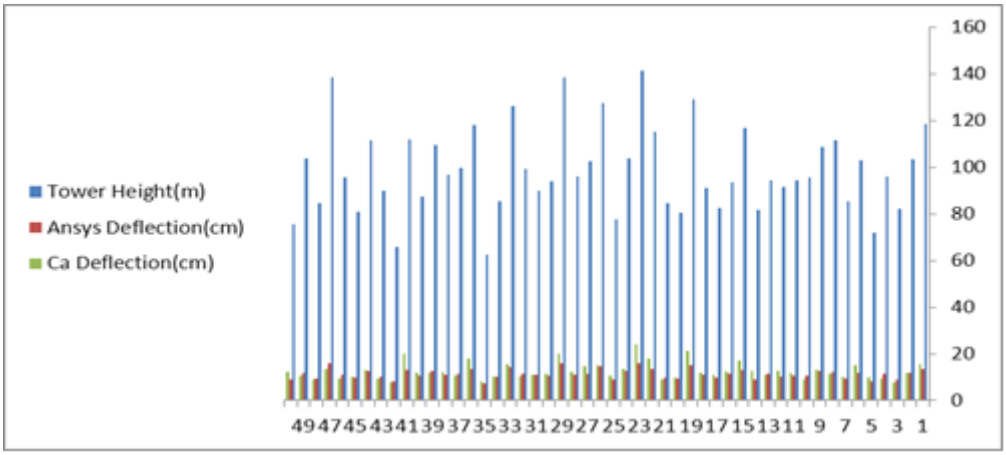

Figure 9: Comparison between maximum lateral deflection by ANSYS and CA due to height

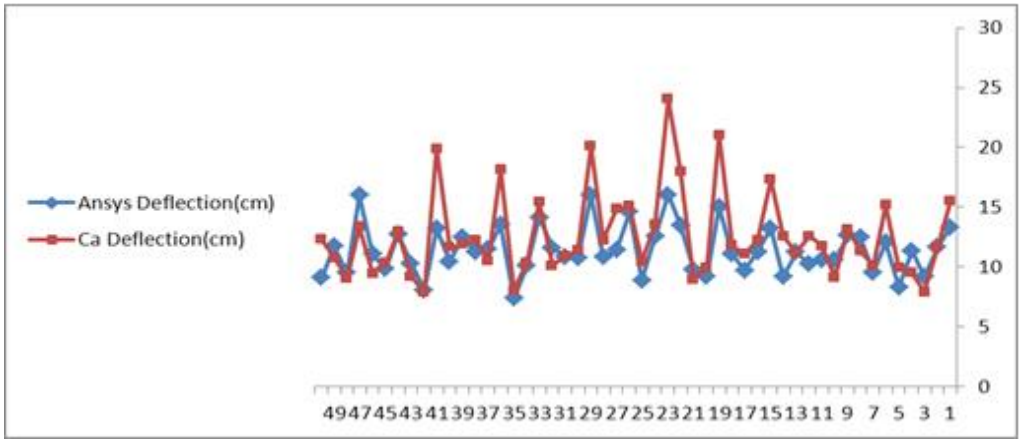

Figure 10: Comparison between maximum lateral deflection using ANSYS and CA

8. CONCLUSION

According to analysis of top displacement in guyed towers by using CA and GP, the average of error percent in GP and CA are 6.81 and 12.86 percent, respectively. The results are shown in figure 11 and 12.

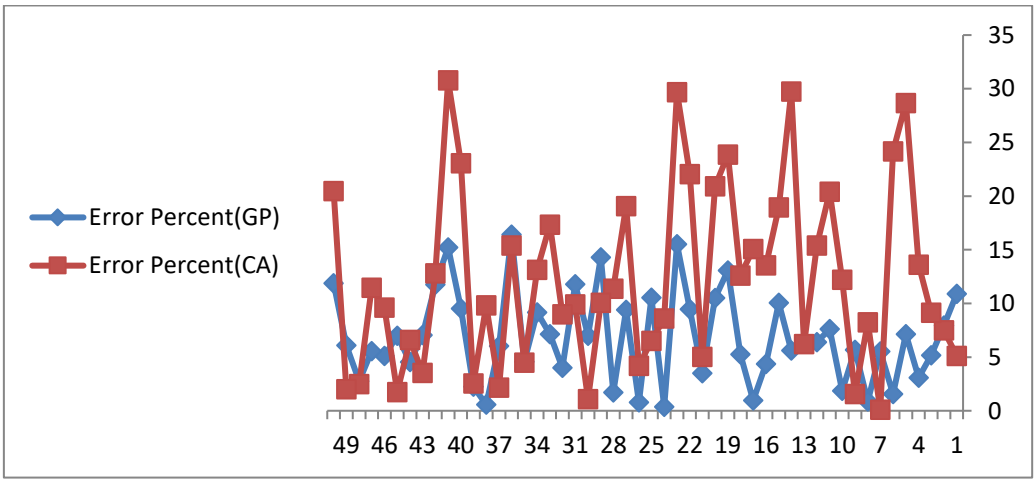

Figure 11: Comparison between error percent in calculation of maximum lateral deflection of guyed tower using CA and GP

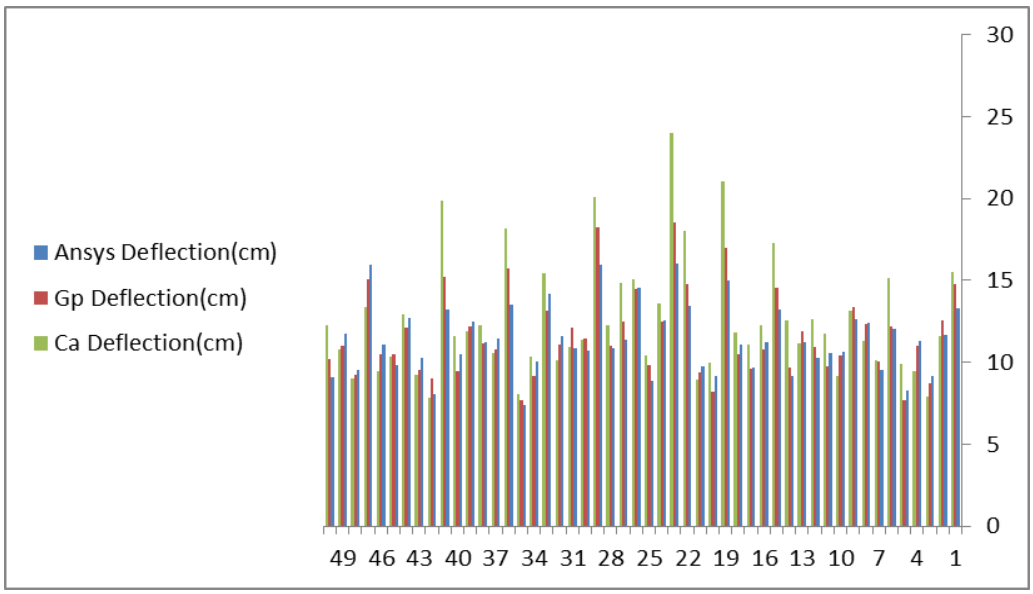

Figure 12: Comparison between maximum lateral deflection using CA, GP and ANSYS software

References

[1] Kaveh Kumarci, Afsaneh Banitalebi, Pooya Khosravyan. (2011). "Cellular Automata In Optimum Shape Of Brick Masonary Vault Under Dynamic Loads", International Journal Of Advanced Structural Engineering professional Journal Of Azad University, South Tehran Branch

[2] Kaveh Kumarci, Pooya Khosravyan, Issa Mahmodi, Arash Ziaie, Mehran Koohi Kamali. (2009), " Optimum Shape In Brick Masonry Arches Under Dynamic Loads By Cellular Automata", Journal Of Civil Engineering (IEB), 37 (1) (2009) 73-90

[3] Wolfram, E. (2002). A New Kind of Science, Wolfram Media, Inc.

[4] Koza, J., R., (1992). Genetic programming: On the programming of computers by means of natural selection. Cambridge, MA: MIT Press.

[5] Ferreira, C., (2006). Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence, 2nd Edition, Springer-Verlag, Germany.

[6] Ferreira, C., (2001). Gene Expression Programming: A New Adaptive Algorithm for Solving Problems, Complex Systems, 13 (2): 87-129.

[7] Kewaisy,T.H., 2001," Nonlinear Dynamic Interaction Between Cables And Mast Of Guyed-Tower Systems Subjected To Wind-Induced Forces, "P.H.D Thesis Department Of Civil Engineering, Texas Tech University.

[8] Shi,H.,2007," Nonlinear Finite Element Modeling And Characterization Of Guyed Towers Under Severe Loading," P.H.D Thesis Department Of Civil Engineering, Missouri-Columbia University.

[9] Rohit Kaul,B.E.,1999," Dynamic Analysis Of Guyed Towers Subjected To Wind Loads Incorporating Nonlinearity Of The Guys," M.S. Thesis, Department of Civil Engineering, Texas Tech University.

[10] Koza, J., R., Keane, Martin A., Streeter, Matthew J., Mydlowec, William, Yu, Jessen, and Lanza, Guido., (2003). Genetic Programming IV. Routine Human-Competitive Machine Intelligence. Kluwer Academic Publishers.

[11] Mitchell, M., (1996). An Introduction to Genetic Algorithms, MIT Press, Cambridge.

[12] Sette, S., and Boullart, L., (2001). Genetic programming: principles and applications. Engineering Applications of Artificial Intelligence, 14, 727–736.

[13] Tsakonas, A., (2006). A comparison of classification accuracy of four genetic programming-evolved intelligent structures. Information Sciences, 176, 691–724.

[14] Moore, A. (2003). New Constructions in Cellular automata, Oxford University Press.

[15] Neumann, V. (1993). Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components, on Neumann's Collected Works, A. Taub (Ed).

[16] Neumann, V. (1996). The Theory of Self-Reproducing Automata, A. W. Burks (ed.), Univ. of Illinois Press, Urbana and London.

[17] Wolfram, E. (1984). Universality and Complexity in Cellular Automata, Physical D.

[18] Wolfram, E. (1983). Statistical Mechanics of Cellular Automata, Rev. Mod.

Journal of Optimization of Soft Computing (JOSC)

Vol. 1, Issue 2, pp: (54-63), December-2023

Journal homepage: https://sanad.iau.ir/en/Journal/josc

Research paper

An optimal approach to detect anomalies in intrusion detection systems

Afsaneh Banitalebi Dehkordi[1*]

1. Department of Computer Engineering, Payame Noor University (PNU), P.O.Box 19395-4697, Tehran, Iran.

Article Info

Article History:
Received: 2024/3/3
Revised:
Accepted: 2024/4/3

Keywords:
Machine Learning,Security, Intrusion Detection System,Software Defined Network

***Corresponding Author's Email Address:** abanitalebi@pnu.ac.ir*

Abstract

Software Defined Networking (SDN) is considered as an innovate architecture of computer networks by using the central controller. Any modification in network data and its arrangement can be effortlessly executed in software via the controller in these networks. Consequently, the identification and timely response to Distributed Denial of Service (DDoS) attacks can be achieved, which is not the case in conventional networks.This paper uses the α-Entropy statistical method considering a threshold and machine learning techniques, K-Nearest Neighbor (KNN), Random Forest (RF) and Support Vector Machine (SVM) to increase the accuracy of detecting DDoS attacks. In this method, the results are evaluated by 10-fold cross validation. The used dataset is ISOT, CTU-13 and UNB ISCX. The results of evaluation with a precision of 99.84% and FPR value of 0.10% indicate the high efficiency of the proposed model in SDN networks.

1. Introduction

SDN is constructed as new network architecture to provide more flexibility in software control of network. In the SDN architecture, the data and control layers are separated, the intelligence of network is centralized and the network infrastructure is separate from the applications [1]. Figure 1 shows the overall architecture of the Open Flow network.

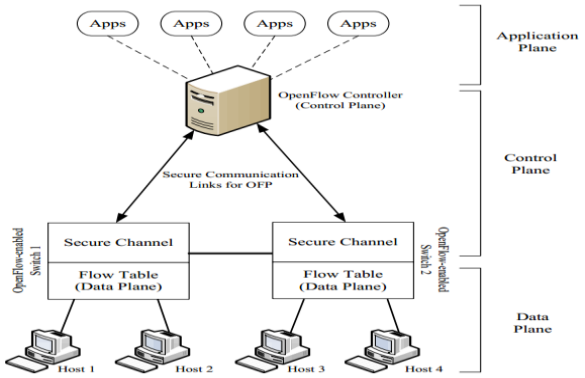

Figure 1. SDN architecture

In this architecture, the flow control is removed from the hardware level in network nodes and is taken centrally and separately by the controller.
The network operating system manages the SDN switches and gathers information from the APIs to create the data plane. [2].An essential situation in SDN networks, which distinguishes them from traditional networks, entails the ability of DDoS attacks to directly target the controller through the transmission of an excessive number of packet-in packets. Consequently, the controller must take into account certain rules for these packets in the flow table of switches and subsequently populate it with counterfeit flow entries[3]. In this research, a hybrid technique is presented to enhance the accuracy of detecting the DDoS attacks. Other sections of this article include the detection of DDoS attacks and their methods, the importance of security issues in SDN networks, the proposed method of this paper to detect DDOS attacks on SDN networks with a combination of statistical methods and machine learning including support vector machine, and a solution to obtain threshold dynamic, introduction of the UNB-ISCX dataset

used in this article, the evaluation of the proposed solution, the results of the statistical analysis and machine learning on the given dataset are evaluated, and finally concluding remarks.This paper is segmented into the subsequent divisions. We articulate certain techniques regarding the identification of DDoS in Division 2. Division 3 formulates a novel approach to identify DDoS assaults through utilization of the α-entropy methodology and classification algorithms. Division 4 discloses the collections of data. Division 5 encompasses the assessments of the performance of our proposed identification techniques on UNB-ISCX, CTU13, and ISOT datasets and elucidates the implementation surroundings. Division 6 recommends the experiments. The outcomes of this manuscript are expounded in division 7. Division 8 contrasts the techniques introduced in this study with some existing techniques. Ultimately, Division 9 presents a concise conclusion and future prospects. Numerous symbols are employed in this manuscript as depicted in Table 1.

Table 1. Major Notations used in this paper

Symbol	Explanation
DDoS	Distributed Denial Of Service
SDN	Software Defined Network
TF	Tolerance-Factor
TP	Time Period
RBF	Radial Basis Function
SVM	Support Vector Machine
KNN	K-Nearest Neighbor
RF	Random Forest
GE	Generalized Information Entropy
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
IP	Internet Protocol
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
TPR	True Positive Rate
FPR	False Positive Rate
AC	Accuracy
AR	Alarm Rate
F_i	Flow number i
T_1	Threshold

2. Related Work

Diverse methodologies were examined in the studies conducted on SDN networks to enhance the security measures implemented on these networks. The ensuing instances illustrate these approaches. In [4], Baczak et al. investigated the data mining methods and machine learning in detecting cyber space penetration. These methods were compared in different aspects such as precision, period of model training, time to classify an unknown sample by the trained model, and the impact of each one in detection of attacks. Yan et al. in [5] proposed a lightweight technique to detect the DDoS attacks. This method is based on hybrid fuzzy evaluation of decision-making model. For example, in case of the denial of service attacks, the network service may be slightly affected, but may still indicate normal status. In this case, 15% of DDoS attack occurs, rather than without any attack. This study uses a variety of features to identify whether an attack occurs or not. Using the formulas presented in the article, the denial of service attacks are detected. Stoyanova et al in [6] proposed a method of detecting a DDoS attack in VANET architecture. The test involves normal and DDoS attack traffics with IP addresses of spoofed resources. Based on traffic features such as the IP address of destination, and through the contribution of three windows with 30 packets, the entropy is computed. The attacks can be detected by calculating the entropy and defining a proper threshold. The provided solution is efficient and adds a low amount of code to the controller module, leading to little impact on the use of CPU.

In another research, Kia in [7] introduced a credible and lightweight solution to detect the denial of service attacks. The proposed method consists of calculating the entropy of the destination IP and the initial rate of flow and examining the characteristics of the packets, by allocating little time to detect attacks. The reduction method involves reducing the idle timer of flow, which is implemented in the case of an attack to contribute the switches to be less under-attack. The use of a single static threshold in this research increases the chance of FP and FN reports. Therefore, the need for a system, which uses optimal thresholds, is felt.

A new detecting method of denial of service attacks, involves calculating the entropy of destination IP to detect attacks in [8]. The provided solution is not only efficient, but also includes a low amount of code in implementation of the controller module, and does not increase the CPU load either in normal or in the event of an attack. In this research, the method of attack detection is based on entropy. One of the limitations of the mentioned method in this paper is that the entropy value cannot accurately detect the attack when the entire network and all hosts are under DDoS attack. A combination of statistical and machine learning methods is considered as the proposed method in this research to identify DDoS attacks. In the statistical section, three types of , Hartley entropy, Shannon entropy and collision entropy method are examined. In the machine learning section, the Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and Random Forest (RF)

techniques are used to enhance the accuracy of DDoS attack detection.

3. Proposed Method

In order to detect DDoS attacks in SDN networks, the proposed method is the combination of α-entropy statistical method and the SVM, KNN and RF Models to increase the accuracy of attack detection. This method is a two-class problem and detects normal flows and attacks. The flowchart, as shown in Figure 2 represents the method. The approach illustrated in Figure 2 comprises various applications. These applications collaborate in order to identify DDoS attacks occurring within the Floodlight controller. Each individual section is introduced in the subsequent text.

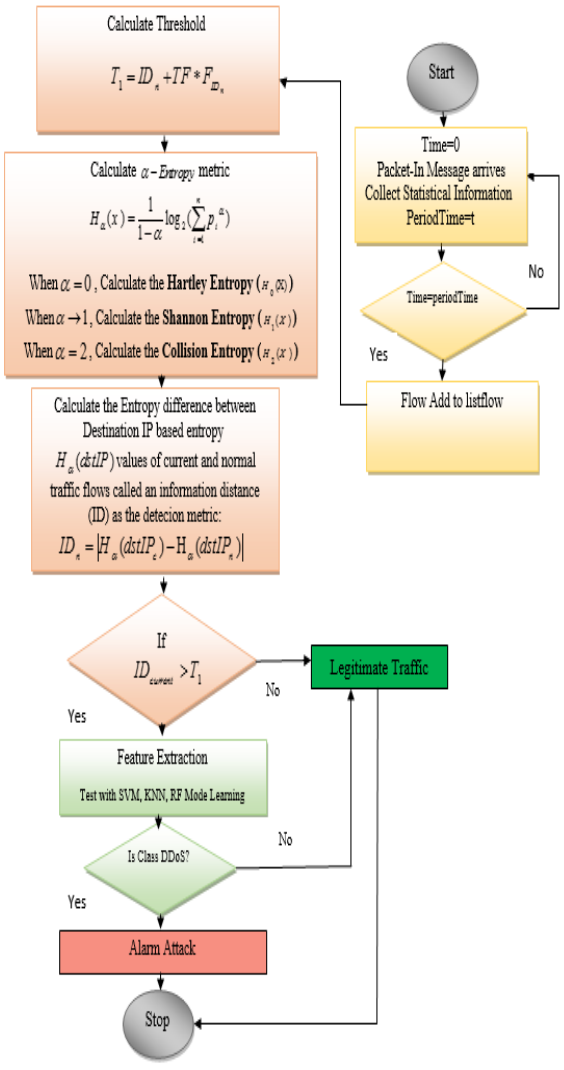

Figure 2. The Proposed Method

3.1. Proposed Statistical Method of α-Entropy

In 1948, Alfred Renyi introduced a more comprehensive definition of Shannon Entropy called generalized information entropy (GE) of the alpha grade, known as α-Entropy or Renyis' Entropy [9], as shown in equation 1.

$$H_\alpha(x) = \frac{1}{1-\alpha}\log_2(\sum_{i=1}^{n} p_i{}^\alpha) \quad (1)$$

Where $\alpha \geq 0$ and $\alpha \neq 1$.

- When $\alpha=0$, Hartley entropy of X is as equation 2:

$$H_0(X) = \log n = \log|X| \quad (2)$$

- When $\alpha \to 1$, the amount of Shannon entropy is as equation 3:

$$H_1(x) = -\sum_{i=1}^{n} p_i \log p_i \quad (3)$$

- When $\alpha=2$, Collision entropy is as equation 4:

$$H_2(X) = -\log \sum_{i=1}^{n} p_i^2 \quad (4)$$

In Figure 3, types of α-Entropy is shown

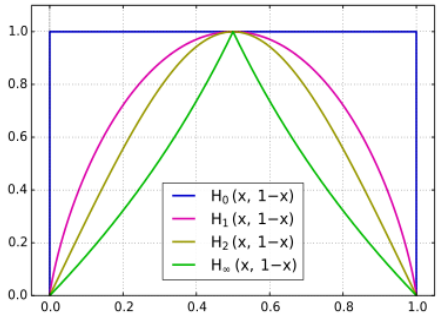

Figure 3. Renyi entropy of a random variable with two possible outcomes

This research addresses the detection of DDOS attacks and, attempts to select the best ID_n of α and increase the accuracy of attack detection by considering different values of α.This technique classifies the network flows in each time window and by calculating the ID_n that represents the difference value of the α-Entropy based on the destination IP address in two network traffic flow in normal and current modes, and it is recognized as detection metric. ID_n is defined in equation 5:

$$ID_n = |H_\alpha(dstIP_c) \_ H_\alpha(dstIP_n)| \quad (5)$$

Where, H_α indicates the α-Entropy in current and normal modes of network traffic flow.
Now, if this value is greater than threshold (T1) as shown in equation 6, attack is detected; otherwise, non-attack status is detected.

$$ID_n > T_1 \quad (6)$$

This technique is able to quickly detect DDoS attacks with low number of packets.

3.1.1. Calculation of Thresholds

In this study, a time-based computational method is used to compute the threshold. One of the aims of this threshold is to detect quickly DDoS attacks in time windows with small width. The relation of this threshold, T1 is presented in equation (7).

$$T_1 = ID_n + TF * F_{ID_n} \quad (7)$$

Where, ID_n value indicates the α-Entropy difference based on the destination IP address in two normal and current modes of network traffic flow as shown in equation 8. TF value is a fixed value. It belongs to the integer set, and is an experimental value used to adjust and improve the threshold dynamic values to detect DDoS attacks. F_{ID_n} represents the standard deviation of ID_n.

$$F_{ID_n} = \frac{1}{t}\sum_{i=1}^{t}(ID_n - \overline{ID}_n)^2 = \frac{1}{t}\sum_{i=1}^{t} ID_n{}^2 - \overline{ID}_n^2 \quad (8)$$

Where, F_{ID_n} represents the standard deviation of ID_n between normal flows.

3.2. Proposed Machine Learning Method

At the second phase of the proposed approach, a proportion of the dataset that was identified as an attack in the preceding phase, regardless of its accuracy, was forwarded to the machine learning component. The remaining dataset being detected correctly was filtered at this step. Different steps of machine learning are described below. Therefore, the pre-processing step and balancing normal data and attack are conducted. This research specifically focuses on the SVM, RF and KNN methods. The methodology used to detect DDoS attacks is analyzed. Then, the results of the SVM, RF and KNN algorithm are compared with other machine learning algorithms.

3.2.1. Extraction of characteristics

In this study, 11 features were considered for detecting the normal and attack packets. Table 2 introduced these features.

Table 2.Feature Extraction

Feature	Explanations
SumNeighborRelationshipsSrc	The number of neighbors connect to transmitter
CountUniDirectionalSrcSToD	The proportion of one-way connections that functioned as the origin for transmitting signals in relation to the overall number of connections to the desired node.
EntropyBytePerPacketSentSrc	Calculating the entropy
SumNeighborRelationdhipDst	The number of neighbors connect to recipient
CountUniDirectionalDstSToD	The proportion of one-way connections that served as the host for the recipient compared to the overall connections of the desired node.
CountSentSrc	The number of the flows which were the desired host of the transmitter.
CountReceiveSrc	The number of the flows which were the desired host of the recipient.
EntropyByte perPacketSentDst	Calculating the entropy of the flows which were the desired host of the recipient
CountSentDst	The number of flows which were the desired host of the transmitter
CountReceiveDst	The number of flows which were the desired host of the recipient
Packet In	This feature is the first packet which transmits a flow for any host in SDN.

These features can be extracted from the switches and controller of the SDN[10]. They are calculated for both ends of a flow and used for each flow in learning. Since separate rules are written in the flow table for two-way communications in software defined networks, the time and characteristics of the receiver and sender determine which flow is related to communication. After building all existing flows, the data between the two hosts in the form of an array which holds the data is given to the next step. These features can be extracted from the switches and controller of the SDN[11].

3.2.2. Support Vector Machine Model

SVM technique is developed by Vladimir Vapnik in 1961 and is based on the linear classification of data [12]. The SVM method uses structural risk minimization, while in other methods the empirical risk minimization is used. Some studies indicated that the SRM method had better performance than the ERM [13]. This paper uses the radial basis

function (rbf) kernel. This kernel function is considered as one of the best functions available in this field. Here, the main parameters of C and Gamma are shown. The parameter C indicates SVM penalty and gamma is presented as kernel coefficient. They are evaluated for the kernel function with the values of Table 3 by the corresponding Python code.

Table 3. C and Gamma values

C	Gamma
١	١
١٠	٠,١
١٠٠	٠,٠١
١٠٠٠	٠,٠٠١

As observed in Algorithm 1, the Python code in this algorithm is used to determine the best value of the C and gamma parameter in the set of determined data, and the best combination of C and gamma parameters in the SVM is obtained automatically by analyzing various combinations of them to find the best responses to the intended dataset, by using the GridSearchCV class in Python.

Algorithm 1.Find Best Parameter values in SVM model

```
From sklearn.model_selection import GridSearchCV
From sklearn.svm import SVC
param_grid = {'C': [1, 10, 100, 1000], 'gamma': [1, 0.1, 0.01,
   0.001], 'kernel': ['rbf']}
grid_svm = GridSearchCV(SVC(), param_grid, refit = True)
```

The tested parameters values are in the param_grid and the optimal ones are in the grid_svm.

3.2.3. K-Nearest Neighbor Model

In this section, the KNN Model is expressed to present the best value of K in set of specified values using elbow technique, For this, the value of 1 to 25 for k was investigated and the error rate was shown by the elbow technique as in Figure 4.

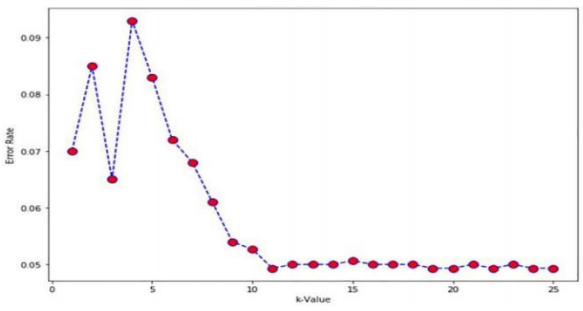

Figure 4. K-Value and error-rate

The results show that the optimum value of k=11 was selected by this technique because in this case the lowest error rate was obtained with value 0.049.In algorithm 2, best k-value is calculated.

Algorithm 2.Find Best K-value in KNN Model.

```
From sklearn.neighbors import KNeighborsClassifier
err_rate = [ ]
For k in range(1, 26):
   knn = KNeighborsClassifier(n_neighbors = k)
   knn.fit(X_train, y_train)
   k_opt = knn.predict(X_test)
   err_rate.append(np.mean(k_opt != y_test))
```

In this algorithm X_ train represents features, y_ train represents the value of output and y_test represents feature test.

3.2.4. Random Forest Model

With Random Forest Model, best response are fined with various decision trees. Four number of decision trees (n_ estimators) are investigated:

- n_ estimators = 10,100,500,1000

The best 'n_estimators' value is provided in Python code in algorithm 3.

Algorithm 3.Find the best decision trees

```
From sklearn.model_selection import GridSearchCV
From sklearn.ensemble import RandomForestClassifier
param_grid = {'n_estimators': [10,100,500,1000], 'criterion':
   ['gini']}
grid_rf = GridSearchCV(RandomForestClassifier(), param_grid,
   refit = True)
```

In this algorithm, the numbers of trees which is tested, holded in param_grid and the optimal response is in the grid_rf.

4. The Datasets

This study employs the UNB-ISCX dataset for the purpose of detecting Distributed Denial-of-Service (DDoS) attacks. This dataset is downloaded from the University of Canadian Institute of Cybersecurity website and is used to detect attacks. In the dataset presented in Table 4, a range of tools were employed to initiate different attacks. The resulting outcomes encompasses a period of network traffic, equating to a total size of 4.6 GB.

Table 4.Data statistics of the modified UNB-ISCX dataset

Application DDoS attack	Av duration	Av number of packets	Av number of flows	Av flow size (pkt)
DDoS improved GET(Goldeneye)	452s	6084	864	7
DDoS GET(ddossim)	138s	46081	22103	2
DDoS GET(hulk)	546s	8482	1085	8

The average rate of normal traffic in the UNB-ISCX dataset is 69 packets per second [14]. In addition to this dataset, the ISOT dataset is also used by the Internet of Things and Network Security Research Center at the University of Victoria. The captured ISOT dataset consists of a pcap file that is approximately 14.1 GB in size[15]. In this research, the normal part of this dataset is used. The details of the introduced attacks in the UNB-ISCX dataset are represented in Table 4.

5. Implementation Environment

The experiments were performed on an ASUS laptop equipped with an AMD (Bristl Ridge), FX-9830P CPU 2.8GHz processor and 12GB of RAM. The chosen operating system was Linux Ubuntu 14.04 LTS, which was executed on a Window 8.1 host machine. The network controller selected for the experiments was Floodlight [16], which relied on Mininet2.2.1 [17] for network simulation. This investigation illustrates the assessment findings of the proposed technique for identifying Distributed Denial of Service (DDoS) attacks. The training and assessment segment are chosen employing the K-

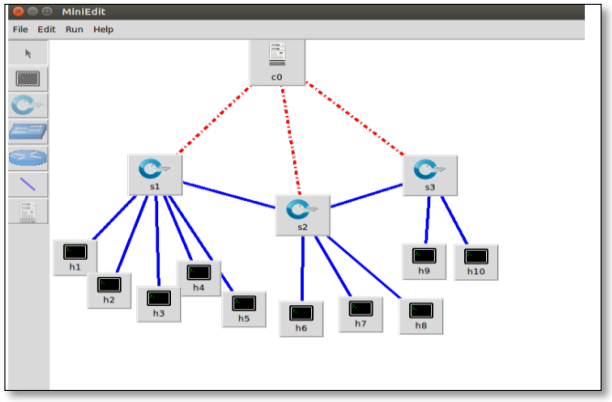

Figure 5.Simulated Topology

This topology is implemented using Mininet. It creates a network topology having 10 hosts connected with three OpenFlow-enabled switchs. All switches are interconnected with each other and are in turn connected with OpenFlow controller as shown. The controller must first collect the information needed from switches, hosts, and existing communications to detect the contraband attacks in order to be able to detect them. Distributed blocking attacks can be identified using different traffic features. The attributes of the headers are extracted using the toolbar. These features are collected every few seconds and the relevant features are calculated using this extracted traffic statistic. Figure 6 shows wireshark software that can be used to view and collect network information efficacy of the suggested resolution can be gauged through the utilization of the F-measure and precision parameters, as well as the accuracy parameter [18].

Fold approach, wherein the number of folds is regarded as ten in this particular investigation. The

6. Experiments

In this section, a clear implementation of the flow-based hybrid approach is presented. Figure 5 shows an overview of the implementation topology and simulation of the attack on this network.

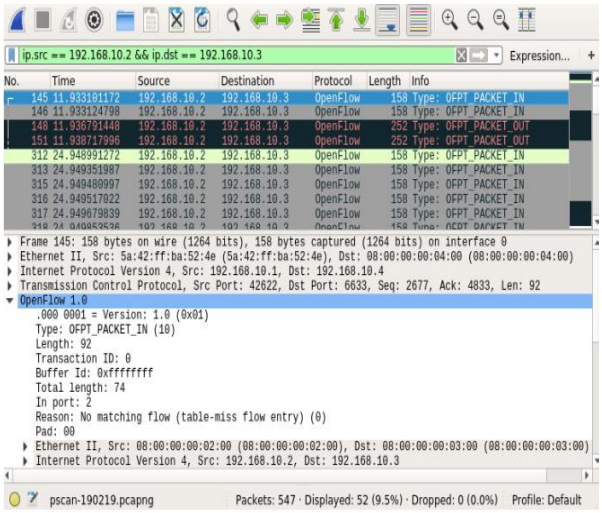

Figure 6.Flow analysis in Wireshark

In this Figure, using the OF filter can open packets and streams of SDN and analyze the statistical results.After the information extracted from the data set is sent to the controller, the Floodlight controller uses the written module to detect and detect the DDoS attack. Figure 7, part of this module, written in the Eclipse Neon software environment, is written in the Floodlight controller using the Java language.

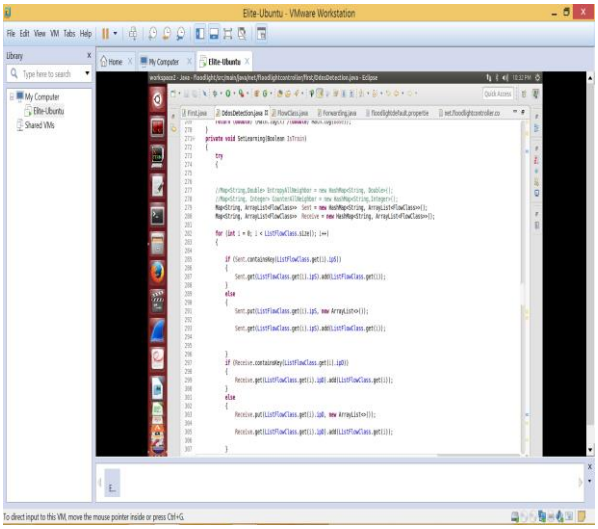

Figure 7. DDoS Detection Module in Floodlight

The implementation code written in Figure 7 shows part of the controller module for detecting attacks on the Floodlight controller. After the results of the statistical methods have been obtained, part of the results are sent to the machine learning section. This section deals with attack detection using classification algorithms in Software.

7. Results

Here, the results of proposed methods are presented for the intended dataset to identify DDoS attacks. For each dataset, the results are presented in two parts. First, the results are presented using the statistical method, and then the results of machine learning are demonstrated.

7.1. Results of Statistical Method

In this section, the results are evaluated in three stages for Hartley entropy, Shannon entropy and collision entropy, considering α=0, α=1 and α=2 values. The time periods are considered between 10 - 240 seconds, then among these periods of time, 15-45 seconds were selected as the best ones, because high level of attacks were detected during these periods, and also were detected earlier. Now, for each selected time, various w and m as tolerance-factors are evaluated. Since these variables are experimental parameters and play a decisive role in two threshold relations, the common values of -1, 0, and 1 are considered for three stages, and the detection results are obtained for each different state. In this table, for each stage of different tolerance-factors, the first mode in which TPR is 100 is considered, which is the case in all attacks in the dataset are correctly identified. Comparing this particular state with different tolerance-factors, the state where the FPR has the lowest value is selected as the chosen time period and its tolerance-factor value is set as the best value. The results of analysis for three Hartley, Shannon and collision entropies are presented in Tables 5, 6 and 7, respectively.

Table 5.Testing the accuracy of attack detection for UNB-ISCX dataset in $\alpha = 0$ For Hartly entropy

TF	TP	H_C	H_N	ID_n	T_1	TPR	FPR	AC
-١	١٥	٢,١٠	١,٨٨	٠,٢٢	1.76	81.45	11.34	**88.01**
	٢٠	٢,١٥	١,٩٤	٠,٢١	2.05	95.65	47.76	**56.15**
	٣٥	٢,٣٢	١,٩٨	٠,٣٤	2.12	100	89.37	**16.50**
	٤٥	٢,٥٧	٢,٢١	٠,٣٦	2.52	١٠٠	90.70	**15.96**
٠	١٥	٢,٢2	٢,١2	٠,١٠	٢,١١	84.18	47.75	**56.11**
	٢٠	٢.25	.٢17	٠,٠8	٢,٠٩	71.56	24.19	**75.63**
	٣٥	.٢30	.٢21	٠,٠٩	٢,١٧	86.04	53.52	**48.24**
	٤٥	٢,٣٩	٢,٢٤	٠,١٥	2.29	100	47.91	**55.01**
١	١٥	٢,١٥	٢,١١	٠,٠٤	2.10	٦٩,٦٤	١٤,٤٦	**٨٤,٣١**
	٢٠	٢,٢٩	٢,١٤	٠,١٥	2.19	٨٣,٥٣	١٧,٢٩	**٨٢,٧٥**
	٣٥	.٢31	٢,١٩	٠,١٢	2.13	٩٣,١٩	٤٩,٦٢	**٥٣,٦٥**
	٤٥	٢,٤٥	٢,٢٠	٠,٢٥	2.27	100	50.60	**52.48**

The less FPR value is obtained compared to other states by examining the results in the first mode for α=0 in Hartley entropy state and different tolerance-factors in different time periods in tolerance -factor = 1 when TPR = 100. Thus, the tolerance-factor = 1 in this mode, and accordingly the threshold value is calculated. In this case, by examining the results, it is observed that no good results are achieved, due to the α-Entropy value obtained for the current state and normal data. As observed, this value in the time period when both attacks and normal-flow are present is very close to the existing dataset, and the ID_n value is very small and will not distinguish the periods that the attacks are present. Therefore, the detection of DDoS attack traffic is difficult using α-Entropy and α=0 at first levels. Now, considering other values for α, we examine different values of α-Entropy. The results for α=1 andα=2 for Shannon and Collision entropy are presented in Tables 6 and 7.

Table 6.Testing the accuracy of attack detection for UNB-ISCX dataset in α=1 For Shannon entropy

TF	TP	H_C	H_N	ID_n	T_1	TPR	FPR	AC
-1	15	2.80	1.91	0.89	2.01	67.44	3.48	**93.55**
	20	2.87	1.95	0.92	2.56	86.95	6.68	**92.66**
	35	3.00	2.05	0.95	2.65	95.65	47.76	**56.15**
	45	3.06	2.15	0.91	2.78	100	38.83	**64.62**
0	15	2.94	1.99	0.95	2.60	71.56	24.19	**75.63**
	20	2.99	2.10	0.89	2.72	84.08	51.25	**50.90**
	35	3.04	2.17	0.87	2.29	98.09	51.88	**50.82**
	45	3.17	2.54	0.63	2.79	100	43.31	**59.87**
1	15	3.01	2.19	0.82	2.35	55.74	17.36	**80.99**
	20	3.14	2.11	1.03	2.57	65.69	23.23	**76.27**
	35	3.17	2.18	0.99	1.02	83.69	5.49	**93.45**
	45	3.21	2.13	1.08	2.19	100	19.10	**81.53**

Table 7. Testing the accuracy of attack detection for UNB-ISCX dataset in α=2 For Collision entropy

TF	TP	H_C	H_N	ID_n	T_1	TPR	FPR	AC
-1	15	3.05	1.90	1.15	2.01	70.00	3.17	**93.15**
	20	3.42	2.01	1.41	3.20	76.47	2.36	**95.13**
	35	4.06	2.16	1.90	3.26	81.01	26.55	**73.73**
	45	4.13	2.50	1.63	2.65	100	51.16	**51.65**
0	15	3.14	2.11	1.03	2.29	83.96	7.11	**92.05**
	20	3.23	2.22	1.01	2.76	87.87	25.79	**74.69**
	35	3.27	2.29	0.98	2.87	98.48	55.40	**46.49**
	45	3.32	2.30	1.02	2.76	100	35.57	**66.59**
1	15	2.85	2.01	0.84	2.67	66.44	15.46	**82.76**
	20	3.24	2.10	1.14	2.85	88.23	3.05	**96.00**
	35	3.35	2.27	1.08	3.02	91.17	23.54	**76.97**
	45	3.47	2.32	1.15	2.67	100	6.11	**94.42**

By examining the results in Tables 6 and 7 for different values of α=1and α=2, it is observed that the ID_n value in these two modes is higher than that of the state with α=0, indicating that with the increase of α the distance between the normal and attack traffic increases, makes the attack and normal traffic more distinguished than each other. The results are illustrated in Figures 8 -10.

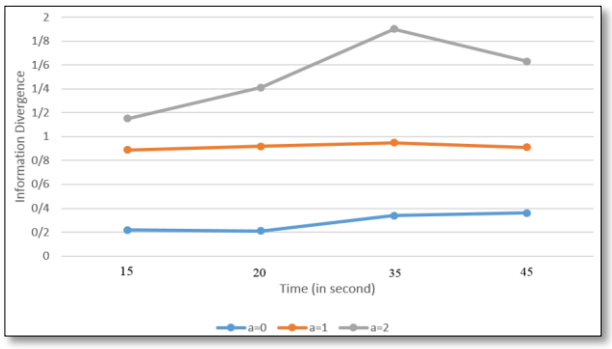

Figure 8. Comparison of ID_n for different modes of α in TF = -1

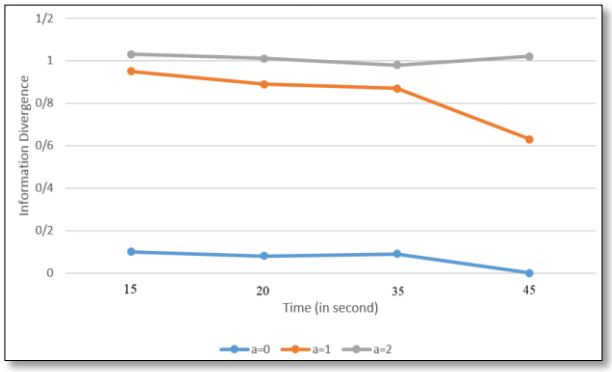

Figure 9. Comparison of IDn for different modes of α in TF = 0

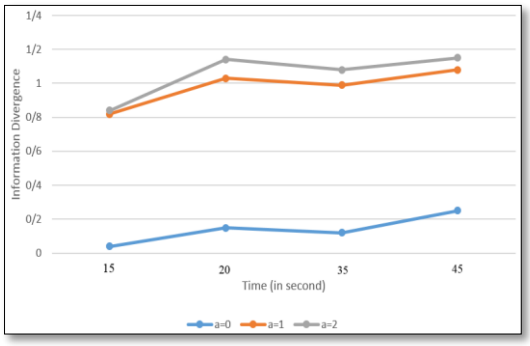

Figure 10. Comparison of ID_n for different modes of α in TF = 1

Based on the results in Figures 8-10, IDn in α=2 state is higher than the other states and is chosen as the selected α .Now, by examining the TPR and FPR values in this state, among different values, the TF =1 is chosen for the threshold relation because the FPR value is the least in this case. In general, by examining the obtained results in the statistical section, the selected state among the three modes is collision entropy with α=2 and TF = 1. In this case, FPR = 6.11% and TPR=100%. Unfortunately, these results indicate that the FPR value is high although TPR is high in this technique. Therefore, in order to eliminate this issue and refine the data and conduct pre-processing, the percentage of attacks alerted in this case are sent to the machine learning section and among different checked algorithms as shown in Table 8.

Model	Decisive Component	Optimized Parameter	TPR	FPR	AC	Precision
KNN	Euclidean distance	K=11	96.22	0.58	99.32	83.48
RF	Bagged Decision Tree	N_est=100	97.41	0.62	98.39	99.38
SVM	RBF Kernel	C=100.0 Gamma=1.0	99.68	0.10	99.84	99.66

As indicated in Table 8, the examined results for a variety of Models are discussed. Regarding the UNB-ISCX dataset, SVM algorithm is the best algorithm to detect DDoS attacks with a precision of 99.84% and a FPR value of 0.10%.

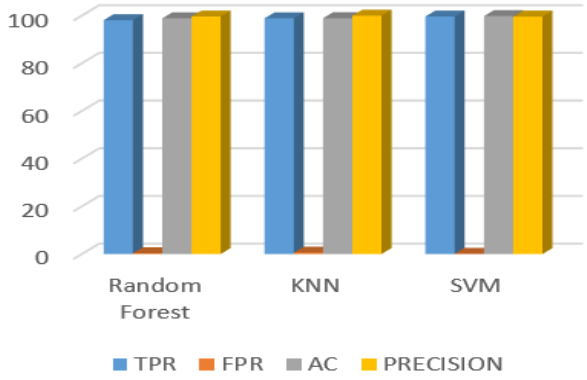

Figure 11. Results of Models for DDoS attacks Detection

As shown in Figure 11, the SVM algorithm has a better performance, compared to other discussed algorithms in machine learning method to detect DDoS attacks, which involves a higher accuracy.

8. Comparing the Method Presented In This Study To Other Methods

This section compares the proposed method in this study to other methods used in other studies. It should be noted that all these studies were conducted to detect DDoS attacks in the UNB-ISCX dataset being also explored in this study. The results on this dataset are summarized in Table 9 and then compared in the Figure 8.

Table 9. Comparing the accuracy of DDoS detection with different methods for the UNB-ISCX dataset

Detection Methods	Ref	AC	FPR
Computer Vision Technique	[19]	90.12	7.92
Cloud Computing	[20]	89.30	Not mentioned
K-Means+NBC	[21]	99	2.2
Neural Network	[22]	98	Not mentioned
Machine Learning Techniques	[23]	81.80	8.2
Ripper + C5.0	[24]	99	2
This Research		99.84	0.10

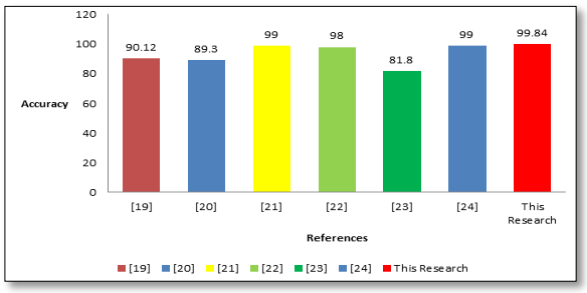

Figure12.Comparing the accuracy of the proposed method to other studies for the UNB-ISCX dataset

The results presented in Table 9 and Figure 12 demonstrate that the accuracy of the proposed method in this study surpasses that of comparable tasks.

9. Conclusion

In this paper, a novel approach was introduced to enhance the precision of identifying Distributed Denial of Service (DDoS) attacks on Software-Defined Networking (SDN) networks. This innovative technique encompasses a combination of statistical analysis and machine learning methodology.In the statistical portion, the three forms of α-Entropy, namely Hartly entropy, Shannon entropy, and Collision entropy, are derived from the examination of the destination IP. Subsequently, the IDn value is computed. By investigating three distinct values for α and examining the outcomes in 2=α for Collision entropy, a greater magnitude of true positive rate (TPR) and a reduced magnitude of false positive rate (FPR) are achieved. Consequently, α=2 is chosen from the scrutinized states. Subsequently, the TF parameter values are appraised to calibrate the desired threshold in this manuscript.A value is chosen that, for the initial instance, completely detected the regular traffic, and its True Positive Rate (TPR) should be 100 while simultaneously possessing the lowest False Positive Rate (FPR). Moreover, at this stage, a True Flag (TF) of 1 is opted for. Subsequently, the proportion of the dataset identified as an attack is transmitted to the machine learning phase in order to enhance the precision of detection.Among the techniques used in the machine learning section, SVM, RF, and KNN, the SVM technique, which has an accuracy rate of 99.84% and a false positive rate (FPR) of 0.10%, demonstrates the highest accuracy in detecting DDoS attacks.

References

[1] Yan, Q., et al., software defined networking (SDN) and Distributed DDenial of service (DDOS) Attacks in Cloud Computing Environments:A Survey ,Some Research Issue ,and Challenges. IEEE Communications Surveys and Tutorials, 2015.

[2] Lim, S., et al., A SDN-Oriented DDOS Blocking Scheme for Botnet-Based Attacks, in ICUFN 2014.

[3] Sahay, R., et al., Towards Autonomic DDoS Mitigation using SDN, in SENT 2015:NDSS Workshop on Security of Emerging Networking Technologies. 2015: United States.

[4] Buczak, A. and E. Guven, A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys and Tutorials, 2016. **18**(2): p. 1153-1176.

[5] YAN, Q., Q. GONG, and F. DENG, Detection of DDOS Attacks Against Wireless SDN Controllers Based on the Fuzzy Synthetic Evaluation Decision-making Model. Ad Hoc & Sensor Wireless Networks, 2016. **33**: p. 275-299.

[6] Todorova, M.S. and S.T. Todorova, DDoS Attack Detection in SDN-based VANET Architectures. 2016, AALBORG.

[7] Kia, M., Early Detection and Mitigation of DDoS In Software Defined Networks. 2015, Ryerson: Toronto,Ontario,Canada.

[8] Mousavi, S.M., Early Detection of DDoS Attacks in SDNController. 2014, Carleton: Ottawa,Ontario.

[9] Bolly, F. and I. Gentil, Phi-entropy inequalities for diffusion semigroups. 2018: Universit´e Paris-Dauphine, Ceremade, UMR CNRS 7534.

[10] Dhawan, M., et al., SPHINX:Detecting Security Attacks in Software Defined Networks, in Network and Distributed System Security Symposium. 2015.

[11] Hoque, N., H. Kashyap, and D. Bhattacharyya, Real-time DDoS attack detection using FPGA. Computer Communications, 2017.

[12] Yadav, A., et al., SDN Control Plan Security in Cloud Computing Against DDoS Attack. IJARIIE, 2016. **2**(3): p. 426-430.

[13] YANG, M. and R. WANG, DDoS detection based on wavelet kernel support vector machine. The Journal of China Universities of Posts and Telecommunications. **15**(3): p. 59-94.

[14] Hadian Jazi, H., H. Gonzalez, and N. Stakhanova, Detecting HTTP-based Application Layer DoS attacks on Web Servers in the presence of sampling. computer Networks, 2016.

[15] Bhamare, D., et al., Feasibility of Supervised Machine Learning for Cloud Security. IEEE, 2016.

[16] Asadollahi, S. and B. Goswami, Experimenting with scalability of floodlight controller in software defined networks, in International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT). 2017.

[17] Mininet ,An Instant Virtual Network on your Laptop (or other PC). Available from: http://mininet.org/.

[18] Cross-validation (statistics). Available: https://en.wikipedia.org/wiki/Cross-validatikon_(statistics).

[19] Tan, Z., et al., Detection of Denial-of-Service Attacks Based on Computer Vision Techniques. IEEE TRANSACTIONS ON COMPUTERS, 2013.

[20] Wang, B., et al., DDOS Attack Protection in the Era of Cloud Computing and Software -Defined Networking, in Network Protocols (ICNP), 2014 IEEE 22nd International Conference 2014.

[21] YASSIN, W., et al., ANOMALY-BASED INTRUSION DETECTION THROUGH KMEANS CLUSTERING AND NAIVES BAYES CLASSIFICATION, in 4th International Conference on Computing and Informatics, ICOCI. 2013.

[22] Saied, A., R. Overill, and T. Radzik, Detection of known and unknown DDOS attacks using Artifitial Neural Networks. Neurocomputing, 2015.

[23] Catania, C. and C. Garcia Garino, Towards Reducing Human Effort in Network Intrusion Detection, in The 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. 2013: Berlin, Germany.

[24] Fallahi, N., A. Sami, and M. Tajbakhsh, Automated Flow-based Rule Generation for Network Intrusion Detection Systems, in 24th Iranian Conference on Electrical Engineering (ICEE). 2016.