In the Name of God

Journal of
Information Systems & Telecommunication
Vol. 9, No.1, January-March 2021, Serial Number 33

Research Institute for Information and Communication Technology
Iranian Association of Information and Communication Technology
Affiliated to: Academic Center for Education, Culture and Research (ACECR)

Print ISSN: 2322-1437
Online ISSN: 2345-2773
Publication License: 91/13216
Editorial Office Address: No.5, Saeedi Alley, Kalej Intersection., Enghelab Ave., Tehran, Iran,
P.O.Box: 13145-799 Tel: (+9821) 88930150 Fax: (+9821) 88930157
E-mail: info@jist.ir , infojist@gmail.com
URL: www.jist.ir

Indexed by:
- SCOPUS www. Scopus.com
- Index Copernicus International www.indexcopernicus.com
- Islamic World Science Citation Center (ISC) www.isc.gov.ir
- Directory of open Access Journals www.Doaj.org
- Scientific Information Database (SID) www.sid.ir
- Regional Information Center for Science and Technology (RICeST) www.ricest.ac.ir
- Iranian Magazines Databases www.magiran.com

Publisher:
Iranian Academic Center for Education, Culture and Research **(ACECR)**

This Journal is published under scientific support of
Advanced Information Systems (AIS) Research Group and
Digital & Signal Processing Research Group, ICTRC

Acknowledgement

JIST Editorial-Board would like to gratefully appreciate the following distinguished referees for spending their valuable time and expertise in reviewing the manuscripts and their constructive suggestions, which had a great impact on the enhancement of this issue of the JIST Journal.

(A-Z)

- Abdolrazzagh-Nezhad, Majid, Bozorgmehr University,Ghayen, Suth Khorasan , Iran
- Agahi, Hamed, Islamic Azad University of Shiraz, Iran
- Ansari, Ebrahim, Institute for Advanced Studies in Basic Sciences, Zanjan, Iran
- Abbasi, Mahdi, Bu Ali Sina University, Hamedan, Iran
- Alizadeh Noughabi, Havva, Islamic Azad University of Gonabad, Iran
- Alavi, Seyed Enayatallah, Shahid Chamran University, Ahvaz, Iran
- Alam Tabriz, Akbar, Shahid Beheshti University, Tehran, Iran
- Afsharirad, Majid, Kharazmi University,Tehran, Iran
- Asgari Tabatabaee, Mohammad Javad, University Of Torbat Heydarieh, Razavi Khorasan, Iran
- Bouyer, Asgar Ali, Shahid Madani University, Azarbaijan, Iran
- Badie, Kambiz, Tehran University, Tehran, Iran
- Babaei, Shahram, Islamic Azad University of Tabriz, Iran
- Barekatain, Behrang, Islamic Azad University, Najafabad, Iran
- Darmani, Yousef, K. N. Toosi University of Technology, Tehran, Iran
- Ebadati, Omid Mahdi, Kharazmi University,Tehran, Iran
- Farsi, Hassan, University of Birjand, South Khorasan, Iran
- Fouladi, Kazim,University of Tehran, Tehran, Iran
- Fatemi Khorasgani, Afsaneh, University of Isfahan, Isfahan, Iran
- Fekri Ershad, Shervan, Islamic Azad University, Najafabad, Iran
- Golsorkhtabaramiri, Mehdi, Bobol University, Mazandaran, Iran
- Ghasemzadeh, Mohammad, Yazd University, Yazd, Iran
- Ghayoomi, Masood, Institute for Humanities and Cultural Studies, Tehran, Iran
- Ghaffari, Ali, Islamic Azad University, Tabriz Branch, Iran
- Ghasemzadeh, Mohammad, Yazd University, Yazd, Iran
- Haghighi, Hassan, Shahid Beheshti University, Tehran, Iran
- Kasaei, Shohreh, Sharif University, Tehran, Iran
- Kheirkhah, Fatemeh, ACECR, Tehran, Iran
- Mohajerzadeh, Amir Hossein, University of Birjand, Iran
- Mohammadzadeh, Sajjad, University of Birjand, South Khorasan, Iran
- Masoomi, Mohsen, Islamic Azad University, Tehran, Iran
- Mavaddati, Samira, University of Mazandaran, Iran
- Momtazi, Saeedeh, Amir kabir University, Tehran, Iran
- Mirroshandel, Seyed Abolghasem, University of Guilan, Rasht, Iran
- Momtazi, Saeed, Amir kabir University, Tehran, Iran

Table of Contents

Phase Transition in the Social Impact Model of Opinion Formation in Log-Normal Networks

Alireza Mansouri*
Department of Information Technology, ICT Research Institute, Tehran, Iran
amansuri@itrc.ac.ir
Fattaneh Taghiyareh
Department of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
ftaghiyar@ut.ac.ir

Received: 04/Aug/2020 Revised: 17/Dec/2020 Accepted: 20/Mar/2021

Abstract
People may change their opinions as a consequence of interacting with others. In the literature, this phenomenon is expressed as opinion formation and has a wide range of applications, including predicting social movements, predicting political voting results, and marketing. The interactions could be face-to-face or via online social networks. The social opinion phases are categorized into consensus, majority, and non-majority. In this research, we study phase transitions due to interactions between connected people with various noise levels using agent-based modeling and a computational social science approach. Two essential factors affect opinion formations: the opinion formation model and the network topology. We assumed the social impact model of opinion formation, a discrete binary opinion model, appropriate for both face-to-face and online interactions for opinion formation. For the network topology, scale-free networks have been widely used in many studies to model real social networks, while recent studies have revealed that most social networks fit log-normal distributions, which we considered in this study. Therefore, the main contribution of this study is to consider the log-normal distribution network topology in phase transitions in the social impact model of opinion formation. The results reveal that two parameters affect the phase transition: noise level and segregation. A non-majority phase happens in equilibrium in high enough noise level, regardless of the network topology, and a majority phase happens in equilibrium in lower noise levels. However, the segregation, which depends on the network topology, affects opinion groups' population. A comparison with the scale-free network topology shows that in the scale-free network, which have a more segregated topology, resistance of segregated opinion groups against opinion change causes a slightly different phase transition at low noise levels. EI (External-Internal) index has been used to measure segregations, which is based on the difference between between-group (External) links and within-group (Internal) links.

Keywords: Social Network; Segregation; Opinion Formation; Opinion Dynamics; Agent-Based Modeling.

1- Introduction

Analytical sociology has emerged as an approach for understanding the social world, concerning important social facts such as network structures, patterns of residential segregation, typical beliefs, cultural tastes, and common ways of acting [1]. Understanding the relationship between micro behavior and macro outcomes is one of the principal concerns of analytical sociology to explain relationships between properties of collectivities or aggregates (such as groups, organizations, markets, and cities) and individuals, their behavior, and how the interaction between them is organized [2].

Opinion formation, a collective behavior process, is a subject of interest in many areas, e.g., psychology, sociology, economics, finance, and politics, which describes group members' actions following a set of rules and its effects on social level [3, 4]. Several opinion formation models have been proposed for opinion formation [3, 5] since the first opinion formation introduced by French, the psychologist, in 1956 [6].

In this research, we used the social impact model of opinion formation [7], based on the psychological theory of social impact, formulated by Bibb Latané [8]. This model is a discrete opinion model, assuming opinion as binary values, e.g., agree/disagree. The model is appropriate for modeling social referendums such as Brexit [9] or investigating people's positive/negative opinions about presidential candidates. Like many other opinion formation models, noise is also considered in this model to describe individuals' stochastic behavior in opinion change[10].

A phase transition is a change of a whole system from one behavior to another [11], initially discussed in physics, e.g., magnetization and thermodynamics. Very analogous

* Corresponding Author

to the phases in physics' magnetization field, an opinion formed in the social level could be described as phases, including majority and non-majority phases at the highest level [12, 13]. Very analogous to the continuous (or second order) phase transition in magnetization [14], phase transition in opinion formation describes conditions where opinion phases may transfer to each other.

One of the key parameters in opinion formation is network topology[15]. Scale-free network with power-law node degree distributed network topology [16] has been widely used for modeling real time networks, including networks for opinion formation[17]. However, recent studies reveal that strongly scale-free structures are empirically rare, while for most social networks, log-normal distributions fit the data better than power-law distributions [18, 19].

In this research, we consider phase transitions in opinion formation using the social impact model in log-normal distribution networks. We have used agent-based modeling and simulation approach for this study.

We have organized the remainder of this paper as follows: first, we briefly review the related background in Section 2; then we explain the research method in Section 3; subsequently, we present the results in Section 4 and discuss the results in Section 5; and finally, we conclude the paper in Section 6.

2- Background

In this section, the main concepts of this study are briefly overviewed.

2-1- Analytical Sociology and Social Simulation

As a traditional discipline of social sciences, sociology studies all forms of human and social dynamics and organization at all levels of analysis, including cognition, decision making, behavior, groups, organizations, societies, and the world system [20]. Analytical sociology aims to explain complex social processes by dissecting them, focusing on their most important constituent components, and constructing appropriate models that help us understand why we observe what we observe [21]. Mathematics has sometimes been used as a means of modeling and formalization in the social sciences but has never become widespread. However, there are some reasons why simulation is more appropriate for modeling social science theories: simulation programming languages are more expressive and less abstract than most mathematical techniques; simulation programs can be modular so that major changes can be made in one part without changing other parts of the program; and simulation systems could include heterogeneous agents, while it is usually relatively difficult using mathematics [22]. Therefore, analytical sociology benefits widely from agent-based simulations as computational tools [2], and many researchers have used agent-based modeling approaches to study sociology phenomena, including opinion formation [23-25]. Over the last decade, the number of papers on using agent-based models to describe how opinions emerge in a group of people has grown at an overall annual rate of 16%, though not continually [3].

2-2- Opinion Formation Models

People may change their opinions due to their interactions with others. Therefore, opinions can be formed and revised through social influence. Opinion formation models describe opinion dynamics and deal with how opinions may be formed and evolved in a social network. Many researchers in social psychology, statistical physics, mathematics, and computer science have focused on the opinion formation models as an interesting challenge in the last few decades [26].

The French opinion formation model introduced in 1956 by French, the psychologist, is the first opinion formation model[6]. After French's model, some other opinion formation models have been introduced. Two main characteristics of every opinion formation model are the opinion space and time model. In the discrete opinion space, opinion values are from a set of discrete values, while in the continuous opinion space, opinion values are from a range of real values. Time modeling also includes continuous time and discrete time. In the continuous time models, time is considered a continuous range, and the opinion formation model is usually presented using a differential equation. While in the discrete time models, time is considered some (equal or non-equal) steps. The discrete time models are more suitable for simulation, including agent-based modeling and simulation; furthermore, difference equations are used instead of differential equations for mathematical representations. The most famous opinion formation models are summarized in Table 1, including their opinion space, time modeling, the main points(s) of opinion dynamics, publication year, and the reference(s). Each opinion formation model's opinion dynamics specify how interacting individuals/agents influence each other's opinions.

2-3- The Social Impact Model of Opinion Formation

In this research, we have used the social impact model of opinion formation [7], a discrete opinion model, with binary value for opinion, e.g., agree/disagree or yes/no. The model is also suitable for modeling opinion formation in online social networks and online communities in which a topic is raised, and users discuss for or against it.

Table 1: Some Famous Opinion Formation Models

Opinion formation model	Opinion Space: Continuous(C) / Discrete (D)	Time in model: Continuous (C)/ Discrete (D)	The main point(s) of opinion dynamics	Year	Reference (s)
French	C	D	average of the neighbors' opinions	1956	[6]
Abelson	C	C	weighted average of the neighbors' opinions	1964	[27]
DeGroot	C	D	weighted average of the neighbors' opinions	1974	[28]
Voter	D	D/ C	adopting the opinion according to the opinion of a randomly chosen neighbor	1975	[29]
Social impact	D	D/ C	impacts from two groups to change or persist on the current opinion	1981	[7]
FJ (Friedkin-Johnsen)	C	D	influencing by his own opinion with weight g_i and others' opinion with weight 1-g_i	1990	[30, 31]
Axelrod	D	D	diversity of opinions and cultures as a consequence of homophily	1997	[32]
Sznajd	D	D	based on Ising model, one dimensional	2000	[33]
Stauffer	D	D	based on Sznajd model, two dimensional	2000	[34]
Deffuant	C	D	opinions of two randomly selected individuals if are not far away from each other move toward each other	2000	[35-37]
HK (Hegselmann – Krause)	C	D	influence by weighted average of all others' opinions	2002	[38]
Majority rule	D	D	the individual adopts the opinion that has a larger value of the sum of the neighbors' opinions	2002	[39]
Altafini	C	D/ C	there are some antagonistic interactions	2012	[40-42]

The social impact model of opinion formation [7] is based on the social impact theory formulated by Latané [8]. According to this theory, the impacts on individuals are exerted by the real, implied, or imagined presence or actions of one or more people or even groups. The impact of source individuals on a subject individual depends on three factors: 1) the (spatial, closeness, time, or abstraction) distance of the source individuals from the subject individual, 2) the source individuals' strength of persuasion, and 3) the number of source individuals. The social impact model of opinion formation consists of N individuals or agents. Any agent i (i=1, 2, . . , N) is assigned one of two possible opinion values, -1 or +1 at any time step, o_i=±1. Moreover, any agent i is characterized by two strengths: persuasiveness strength (p_i) and supportiveness (s_i) strength. The p_i is the capability to persuade another agent with the opposite opinion to change its current opinion, and the s_i is the capability to persuade another agent with the same opinion to stay on its current opinion. Any agent i experiences total impact I_i from other interacting agents, js, formulated as (1), in which d_{ij} denotes the distance between two individuals i and j, and α determines how fast the impact decreases with the distance d_{ii}. The social impact between any two interacting agents is similar to the physical force that governs gravity between any two objects by Newton's law, $F=G(m_1m_2)/r^2$, in which G is the universal gravitation constant, m_1 and m_2 are masses of two objects (similar to persuasion strengths p_i and s_i in (1)), r is the separation between the objects (similar to d_{ij} in (1)). The power of 2 of d in Newton's law is similar to the parameter α in (1). Some implementations of (1) [43-45] have also assumed α=2.

$$I_i = \left[\sum_{j=1}^{N} \frac{p_j}{d_{ij}^{\alpha}}(1-o_i o_j)\right] - \left[\sum_{j=1}^{N} \frac{s_j}{d_{ij}^{\alpha}}(1+o_i o_j)\right] \quad (1)$$

The summations at the right-hand side of (1) calculate the impact of interacting agents trying to persuade agent i to change its opinion and the impact of interacting agents on agent i to persist in its current opinion, respectively. Thus, the overall impact on agent i to change (or persist on) its current opinion is calculated by (1).

Eq. (1) expresses the deterministic part of interacting agents' social impact on agent i, while there is a non-deterministic part affecting agent i, called noise, h_i. This non-deterministic part is initiated from the environment (e.g., public media) and the individuals' characteristics that determine how every individual is influenced by others (depending on many psychological factors). Thus, the social impact model of opinion formation formulates the opinion dynamics as (2), indicating the opinion of agent i at time step t+1 regarding the impact from interacting agents at time step t and all other non-deterministic factors summarized in noise parameter h_i. The non-deterministic part of the model usually has no bias toward any opinion. Therefore, h_i is usually regarded as white noise and is implemented as a random variable from a uniform distribution with a mean value equal to zero. The sign function in (2) maps negative values to -1 and positive values to +1.

$$o_i(t+1) = -sign\left[o_i(t)I_i(t) + h_i\right] \quad (2)$$

2-4- Log-Normal Distributed Network

Scientists from different fields are trying to understand the structure and properties of real-world networks. A new branch of mathematics called random graph theory focusses on the probabilistic methods to model real-world networks. In many scientific domains of networks, it is claimed that most of the real-world networks are scale-free, varying in some details, and generally, a network is scale-free if the fraction of nodes with degree k follows a power-law distribution $k^{-\alpha}$, where $\alpha>1$ [18]. However, scale-free networks' universality is controversial, and some recent studies reveal that log-normals often fit degree distribution as well or better than power-laws [18, 46, 47].

2-5- Phase Transition

The term phase transition was initially used by physicists to describe a change from one behavior to another in the thermodynamic or macroscopic limit [11]. Changing the values of a set of parameters, e.g., the temperature in physical systems may cause a transition from one phase to another. For example, changing between solid, liquid, and gaseous state of matter [48], or temperature may cause a change in the ferromagnetic state in materials such as iron, nickel, or cobalt [49]. Similarly, the phase transition is used in other sciences, including social systems [50, 51].
In this research, we consider the social phases from opinion formation viewpoint very similar to those defined in [12], as follows:

- Majority phase: population of agents with each opinion are not equal and could be recognized as a majority opinion and (probably) a minority opinion:
  - Consensus: All of the agents have the same opinion.
  - Frozen majority: continuing time steps cause no change in opinion of any agent.
  - Orderly fluctuated majority: some agents change their opinion in every time steps (and other agents do not change their opinion). Fig. 1 illustrates how this fluctuation may happen in a network of agents following the social impact model with the same persuasiveness and supportiveness strengths.
  - Non-orderly fluctuated majority: at least some agents change their opinions with no specific pattern.
- Non-majority phase: populations of agents with each opinion are (roughly) equal, such that no opinion could be recognized as the majority opinion of society. Very similar to the defined majority phase, the following three states may occur in this case:
  - Frozen non-majority,
  - Orderly fluctuated non-majority,
  - Non-orderly fluctuated non-majority.

Fig. 1 An example of orderly fluctuating agents with two opinions (black and white) in the social impact model with the same strengths

2-6- Segregation and Measuring using EI Index

One of the phenomena affecting opinion formation is segregation. Segregation is defined as "the degree to which two or more groups live separately from one another" [52]. Segregation depends on the network structure. In a highly segregated network with two opinion groups, there are two sub-network. All of the nodes (individuals/ agents) of each sub-network have the same opinion. There are some links between nodes of the same group and no link between nodes from different groups. Therefore, no opinion will change due to interaction with a node from the other opinion group.
There are various approaches to measuring segregation in social networks [52]. In this research, we have used the EI (External-Internal) index, which determines a value for the whole network based on the number of links between nodes from different groups, or external links (EL) and the number of links between nodes of the same groups, or internal links (IL), according to (3):

$$EI = (EL - IL)/(EL + IL) \quad (3)$$

Indeed, the EI index is defined as the difference between between-group links and within-group links, divided by the total number of links for normalization. EI takes a value between -1 (all links are within-group links, thoroughly segregated) and +1 (all links are between-group links, not segregated).

2-7- Noise in opinion formation models

The complex human being in various behaviors, including opinion formation, could not be represented by a simple deterministic model. Therefore, to be more realistic, almost opinion formation models have a non-deterministic or stochastic part. In the language of statistical mechanics, the stochastic component is called noise, which is added to the deterministic dynamics of an opinion formation model [10].
Some studies on the effect of noise in various opinion formation models have been published, mainly focusing on the effect of noise on phase transitions, e.g., [12, 53] for the social impact model, [10, 54-57] for the Deffuant model [35], [58] for the Sznajd model [59], and [60-62] for the HK model [38].

3- Method

To study phase transitions in opinion formation according to the social impact model in log-normal distributed random networks, we implemented an agent-based model. Some details of the method are described in this section.

3-1- Generating Log-Normal Random Networks

To generate random networks with N nodes and log-normal node degree distribution, we have used the Bianconi-Barabási algorithm [63], an algorithm using growth and preferential attachment mechanisms. The growth mechanism means that the network continuously expands gradually by adding new nodes to the network, attaching to the current nodes in the network, and preferential attachment means that a new node links with higher probability to the nodes with higher degrees. In the Bianconi-Barabási algorithm, m_0 and m determine the number of initial nodes and the number of edges added with any newly added node, respectively. By setting the algorithm's fitness parameter, the generated network's node degree distribution fits the desired distribution, including the log-normal distribution we used in this research. The generate_BB function from PAFit package [64], implemented in R has been used to generate networks in this research. Since the simulation has been considered for 1000 agents (N=1000) and every agent is assigned to one of the generated random network nodes, network size As shown in Table 2, In this research, we assumed m_0=2 and m=2.

3-2- Implementation of the Simulation

Eq. (1) and Eq. (2) for the social impact model contain some parameters. In this subsection, value assignments to those parameters are described. The persuasiveness and supportiveness parameters, p_i and s_i in Eq. (1), for each agent are assigned using a uniform random variable in the range (0..P_{max}) and (0.. S_{max}), respectively. We assumed both P_{max} and S_{max} are equal to 100.

In this research, the distance between any two connected agents i and j, d_{ij} in Eq. (1), equals 1. Therefore, regardless of the value of α, d_{ij}^{α}=1 for any connected agents i and j.

Since some random variables play important roles in the simulation, the simulation runs N_{run} (=30) times for each input parameter set, and the statistics of output values (β_{trend} and β_{final}), including the mean values and standard deviations, are calculated and reported.

Table 2: Constant Parameters of The Simulation

Parameter	Value	Parameter Description
N	1000	The number of agents
MaxStep	1000	Time steps for every simulation run
P_{max}	100	The maximum value of persuasiveness power
S_{max}	100	The maximum value of supportiveness power
d_{ij}^{α}	1	The same distance (=1) assumed between any two connected nodes (agents) i and j; therefore, equal d_{ij}^{α} values (=1) regardless of α value
m_0	2	The nunber of initial nodes for generating log-normal network
m	2	The number of edges added with any new added node during generating log-normal network
N_{run}	30	The number of runs for any unique parameter combinations to derive statistics

Table 2 summarizes the above mentioned parameters and the assigned values in the simulation.

The following parameters are used as the independent or input parameters:

- h: The noise level of the social impact model. Indeed, h_i for agent i in Eq. (2) is a random value from the uniform random variable Uniform(-h, +h), whose mean value equals zero. The simulation has been run for various noise levels from 0 to 2000 with steps 200.
- β: Indicates the percentage of agents with opinion '-1'; therefore, other agents' opinions are '+1'. Any simulation start with a β value showing the initial combination of both opinion groups. Then β values may change during simulation time steps until the last time step (*MaxStep*) at which opinion combination reaches βfinal. The values of initial β for various simulation runs are 0%, 10%, 20%, 30%, 40%, and 50%. The system behavior for initial β values more than 50% are the same as for 1-β and changing the initial assignment of opinions to the agents ('-1' instead of '+1' and '+1' instead of '-1').

Fig. 2 shows the pseudo code of the agent-based model. To more clarification, the equivalent flowcharts are also shown in Fig. 3 and Fig. 4.

For every N_{run} simulation repetitions with different random seed values of any combination of independent parameters h and initial β, the β values in every time steps are measured and saved as β_{trend}. Then for every input parameters h, initial β, and N_{run} the output β_{trend} is available for reporting statistics and drawing the figures showing the system behavior.

Algorithm 1: Pseudo code for the simulation

```
procedure SIMULATE
    for h from 0 to 2000 by 200 step do [noise levels]
        for β from 0% to 50% by 10% step do [initial percentage of agents with opinoin `-1']
            for run_counter from 1 to N_run do [simulation runs with different random number sequences]
                initialize rand_seed to a new seed value [to generate new random number sequence]
                β_trend[run_counter] = RUN_SIMULATION(h, β, run_counter)
            end for
            save h, β, and β_trend
        end for
    end for
    draw output diagrams using the saved variables h, β, and (the corresponding) β_trend
end procedure

Function RUN_SIMULATION (h, β, run_counter) [run_counter affects rand_seed which in turn affects creation of random
        network and assigning random values for persuasiveness and supportiveness strengths of the agents]
    N = 1000 [The number of agents]
    Log_Norm_Graph = Create_Log_Normal(N, m_0, m)[N nodes, m_0 initial nodes, m edges added with any new node]
    create N agents and randomly assign each agent to one node of Log_Norm_Graph
    randomly assign -1 opinion to β percent of the agents, assign others' opinions to +1
    initialize β_arr[0] to (the current initial) β value (at time step #0) [β_arr is an array of β values for each time step
        from initial time step (0) to MaxStep]
    for each agent A_i do
        generate p_i and s_i using uniform random distributions Uniform(0, P_max) and Uniform(0, S_max) respectively
    end for
    for time_step from 1 to MaxStep do
        for every agent A_i do
            A_i_con = agents connected to A_i according to Log_Norm_Graph [assume A_i connects to itself]
            I_i_pers = I_i_sup = 0 [initialize sum of the impacts from persuading and supporting agents]
            for every A_j in A_i_con do
                if A_j's opinion = A_i's opinion
                  I_i_sup = I_i_sup + s_j [to calculate the sum of supportive impacts]
                else
                  I_i_pers = I_i_pers + p_j [to calculate the sum of persuading impacts]
                end if
                I_i = 2*I_i_pers - 2*I_i_sup [Eq. (1)]
                h_i = a random value from Uniform(-h, +h)
                if I_i + h_i > 0 [opposite opinion overcomes the agent's current opinion]
                  A_i's next opinion = -1 * A_i's opinion [change the opinion in next time step] [Eq. (2)]
                end If
            end for [every A_j in A_i_con]
        end for [every agent A_i]
        for every agent A_i do
            A_i's opinion = A_i 's next opinion
        end for [every agent A_i]
        current_β = the percentage of the current agents with opinion '-1'
        β_arr[time_step] = current_β
    end for [time_step]
    return β_arr
end function
```

Fig. 2 Pseudo code of the simulation

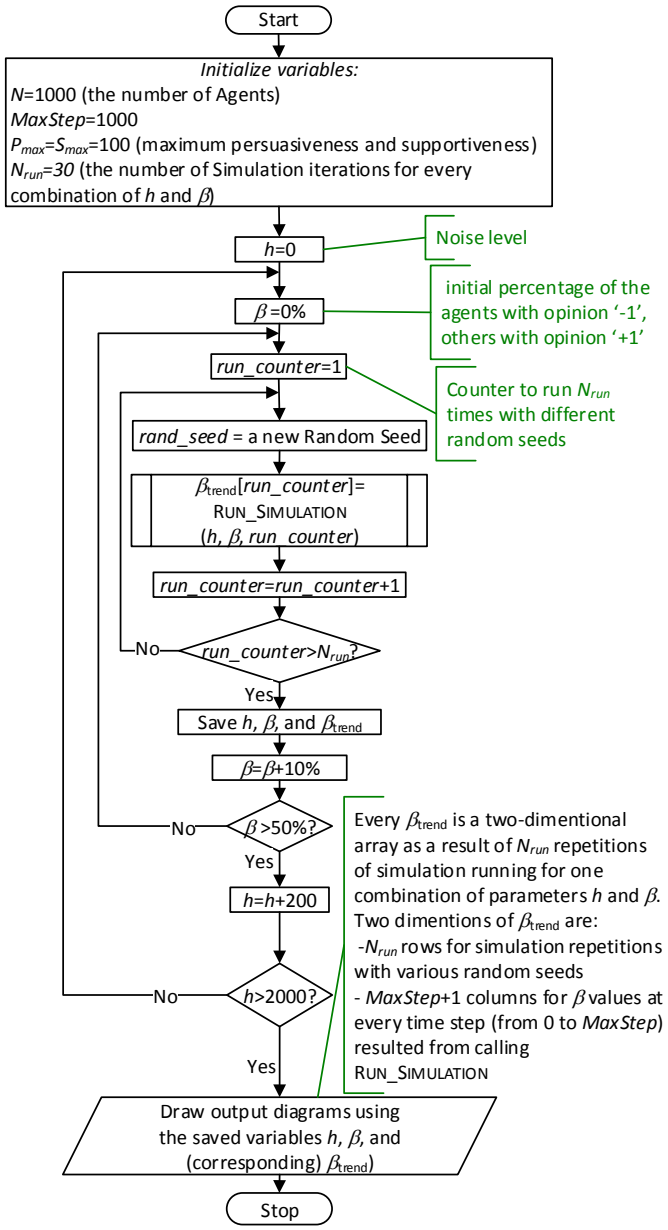


Fig. 3 Simulation flowchart, calling Run_Simulation function for N_{run} times for any combination of input parameters h, β, and finally drawing trend of β values

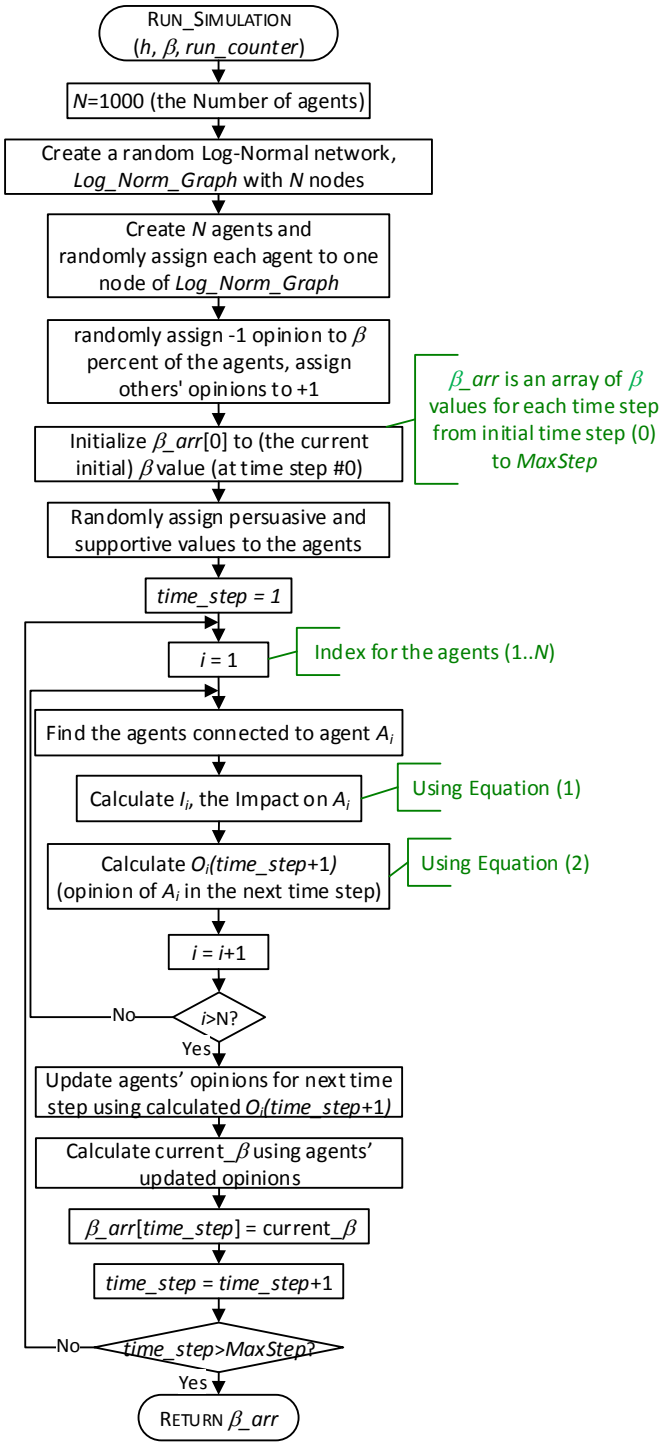


Fig. 4 The flowchart of Run_Simulation function of Fig.3 for calculating the agents' opinion at the next time step

4- Results

The results of running the simulation algorithm described in the previous section are presented in this section.

We use error fill plots to show the results. To clarify the presentation style of the results in error fill plots in this section, Fig. 5 shows the result of a subset of the results as an example, in which h=0, β starts from 30%, simulation repetitions is eight (instead of all 30 repetitions), and the number of time steps is 20 (instead of all 1000 time steps).

The top figure shows β values during time steps for each repetition of simulation. The down figure shows the corresponding error fill plot, which shows the curve of the mean values of the β values at each time step with a shaded area showing its standard deviation (SD). In this

case, an orderly fluctuation, discussed in section II (Fig. 1), causes a regular fluctuation after a few time steps. Such a fluctuation in some of the next figures for whole time steps (1000) are seen as thick lines. As another example, with the same parameters as Fig. 5 and changing starting β to 50% results in the plots shown in Fig. 6, where orderly fluctuation happens.

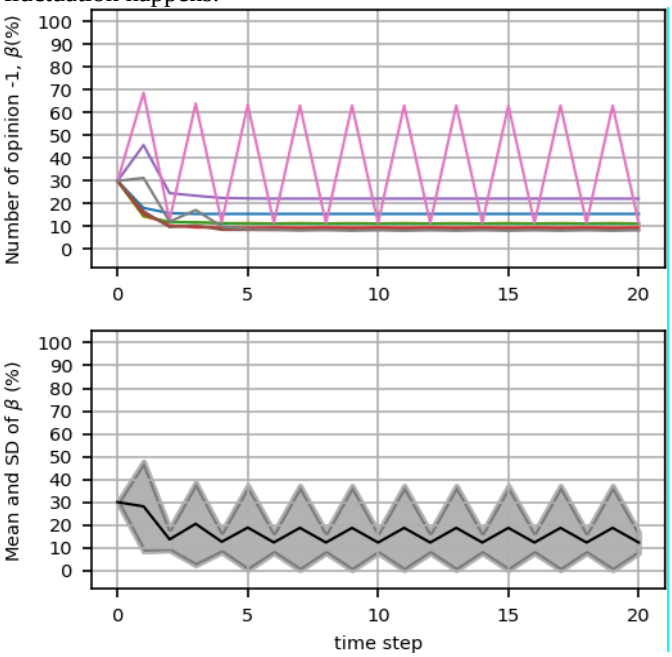


Fig. 5 Top: β values of eight simulation repetitions for h=0, initial β=30%, for 20 time steps; down: corresponding error fill plot.

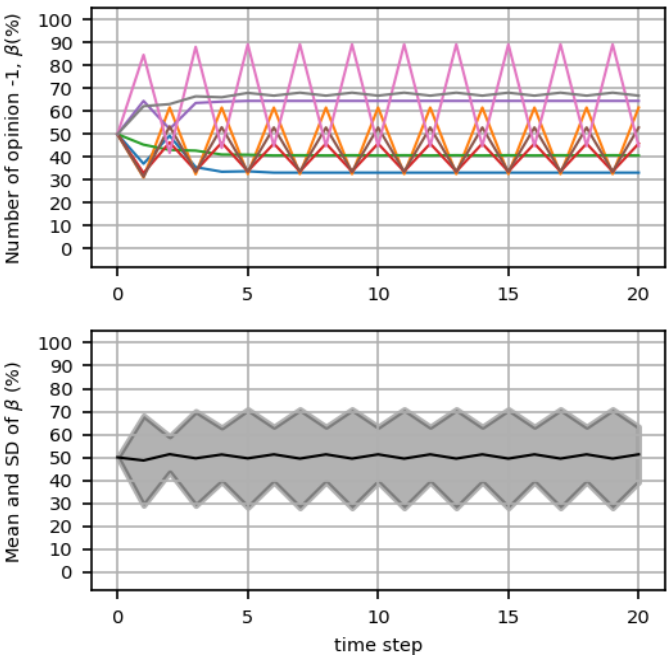


Fig. 6 A sample subset of simulation repetitions, the same parameters as Fig. 5, but β=50% (instead of 30%)

When the noise level increases, some agents randomly change their opinion due to the system's more stochastic behavior. Therefore, the observed fluctuation for the error fill curve is not regular. Fig. 7 and Fig. 8 show examples for this case for h=600 and starting from β=30% and β=50%, respectively.

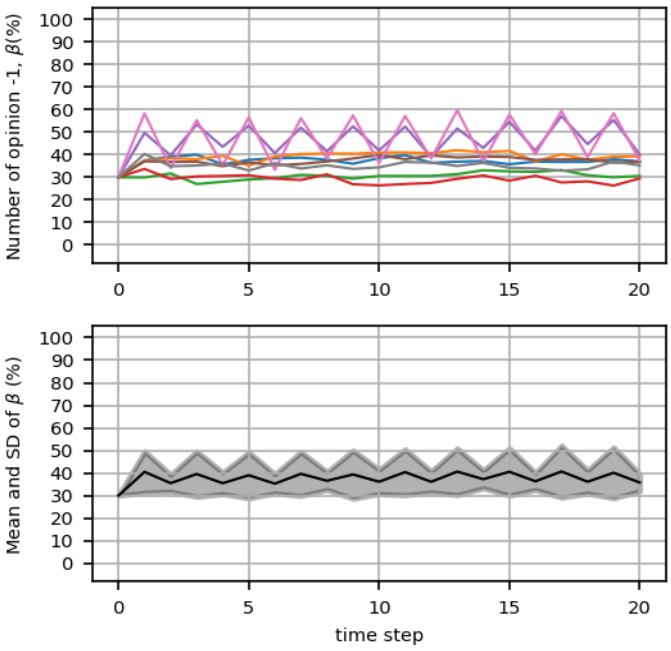


Fig.7 Top: β values of eight simulation repetitions for h=600, initial β=30%, for 20 time steps; down: corresponding error fill plot.

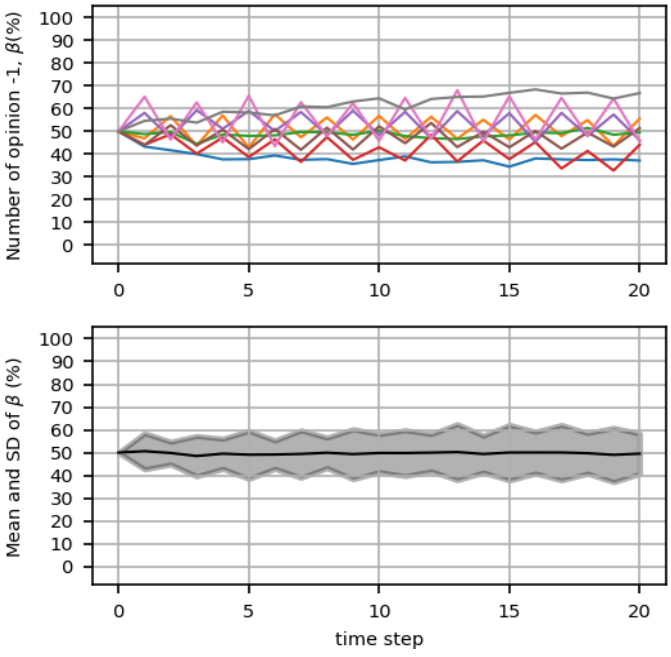


Fig. 8 A sample subset of simulation repetitions, the same parameters as Fig. 7, but β=50% (instead of 30%)

For high enough noise levels, the system behaves more stochastically. Fig. 10 and Fig. 11 show the results for h=2000, again for eight simulation repetitions for the first 20 time steps, starting from β =30% and β =50%, respectively.

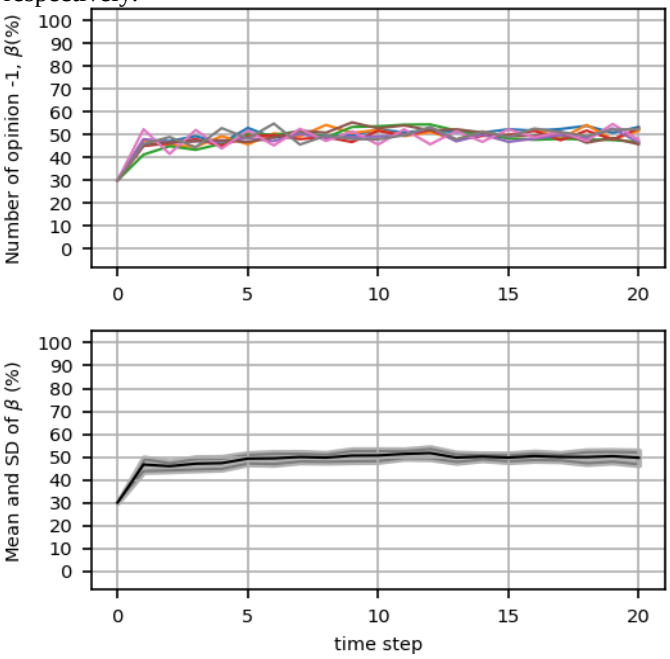


Fig. 9 A sample subset of simulation repetition for h =2000, starting from β=30%, running for 20 time steps; top: details of β values for eight sample run repetitions, down: corresponding error fill plot for the top run repetitions.

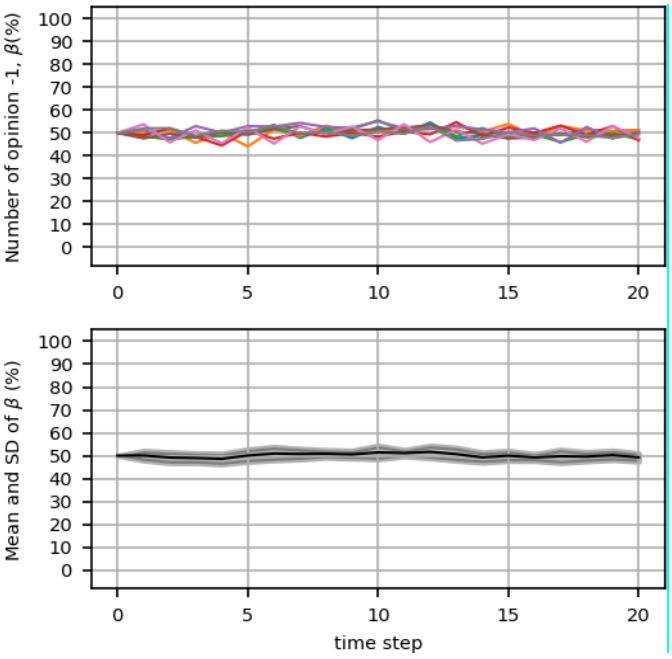


Fig. 10 A sample subset of simulation repetitions, the same parameters as Fig. 9, but β=50% (instead of 30%)

Now, the results for the whole 30 repetitions of simulation runs and all 1000 time steps are presented. Fig. 11 shows the values of β for the case where there is no noise, h=0. The left diagram shows the mean value and (shaded) standard deviation of β in every time step for N_{run} simulation run from start to *MaxStep*. Since the shaded area showing the standard deviations overlap and are not clear, the standard deviations of β_{final} (β values at the final time step, *MaxStep*) for each initial β is shown in the right diagram (black bars) in companion with min-max (red bars), and medians (blue bars). As the figure shows, since the system is noise-free in this case, the majority phase (frozen or orderly fluctuated) happens very soon in the few initial steps.

Similarly, for the next step of noise h, the simulations' result is shown in the following figures: Fig. 12 for h=200, Fig. 13 for h=400, Fig. 14 for h=600, Fig. 15 for h=1000, and Fig. 16 for h=2000.

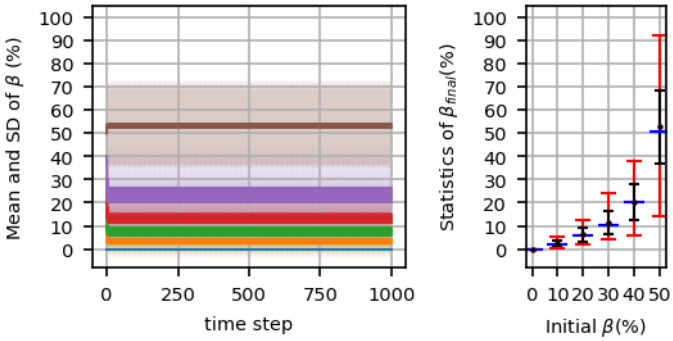


Fig. 11 h=0, noise-free simulation: Left: the mean value and (shaded) standard deviation of β for time steps, Right: Mean (circle marker), standard deviation (black bars), min-max (red bars), and median (blue bars) of $\beta_{final.}$

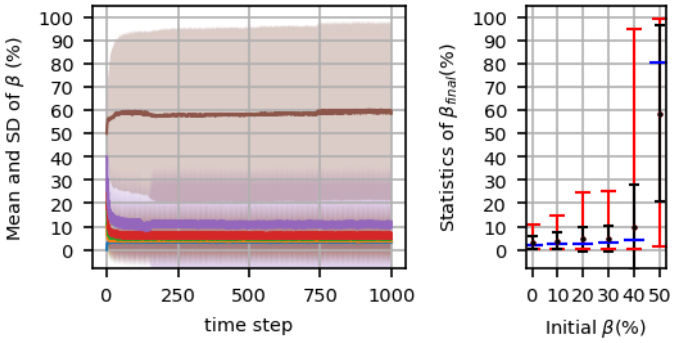


Fig. 12 h=200: Left: the mean value and (shaded) standard deviation of β for time steps, Right: Mean (circle marker), standard deviation (black bars), min-max (red bars), and median (blue bars) of $\beta_{final.}$

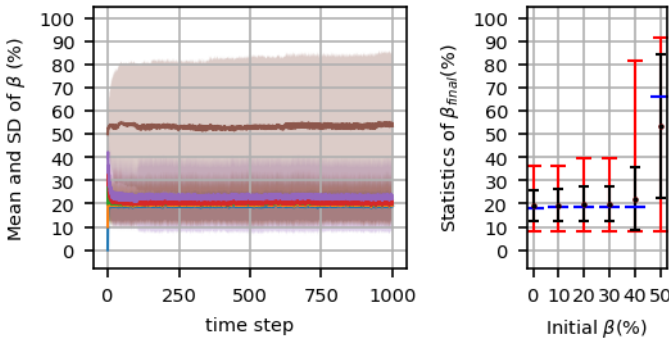


Fig. 13 h=400: Left: the mean value and (shaded) standard deviation of β for time steps, Right: Mean (circle marker), standard deviation (black bars), min-max (red bars), and median (blue bars) of $\beta_{final.}$

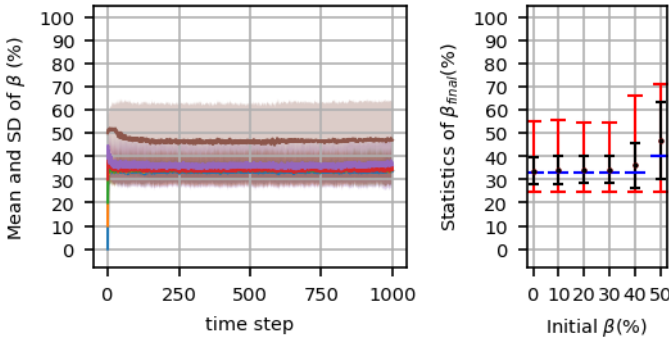


Fig. 14 h=600: Left: the mean value and (shaded) standard deviation of β for time steps, Right: Mean (circle marker), standard deviation (black bars), min-max (red bars), and median (blue bars) of $\beta_{final.}$

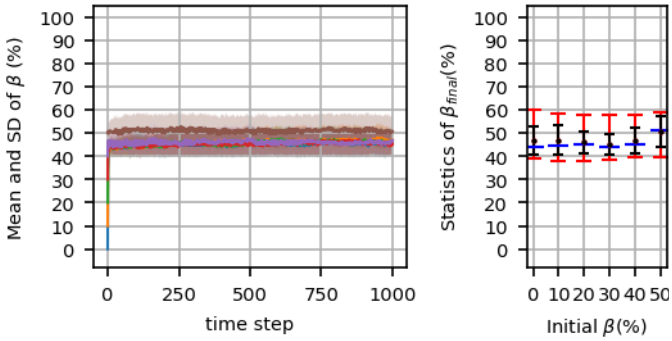


Fig. 15 h=1000: Left: the mean value and (shaded) standard deviation of β for time steps, Right: Mean (circle marker), standard deviation (black bars), min-max (red bars), and median (blue bars) of $\beta_{final.}$

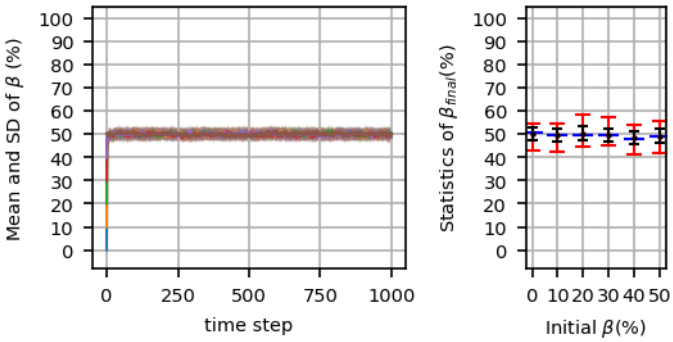


Fig. 16 h=2000: Left: the mean value and (shaded) standard deviation of β for time steps, Right: Mean (circle marker), standard deviation (black bars), min-max (red bars), and median (blue bars) of $\beta_{final.}$

Fig. 17 shows the results of the simulations from another viewpoint. In this figure, for each value of β a diagram is depicted. The horizontal axis shows initial β values, and the vertical axis shows β_{final} Values. The black curve shows the mean value of β_{final} with associated standard deviation in shaded form. Similarly, the red curve shows the result of simulations with the same parameter values, but network topology is scale-free Barabási-Albert random network as in [12] instead of log-normal. The results will be compared in the next section.

5- Discussion

As Fig. 11 shows, the behavior of the social system is fully deterministic in the noise-free case, starting from consensus (β=0%) results in a consensus (β_{final}=0%) because there is no opposite opinion and no stochastic behavior to cause any agent to change its opinion. For other β values, the system reaches the equilibrium state of a frozen or orderly fluctuated majority phase in a few time steps. Indeed, when the system starts with noise-free or very small noisy conditions, a frozen or orderly fluctuated phase happens in equilibrium because there is no noise to cause any spontaneous opinion change.

When the simulation starts from 0%<β<50%, it results in β_{final}<β because the majority opinion group dominates the minority group, and the minority group's population shrinks down, but the segregation phenomenon causes no consensus (except β=0%, which means starting from consensus). In most of the noise-free cases, starting from β=50% causes a majority phase, in some cases toward '-1' opinion, and in some cases toward '+1' opinion. In some rare cases starting from β=50, the system may reach (frozen or orderly fluctuated) non-majority phase with β_{final}=50%. As Fig. 11 shows, the mean value of β_{final} is roughly equal to 50%, but the min-max and standard deviation have a relatively wide range.

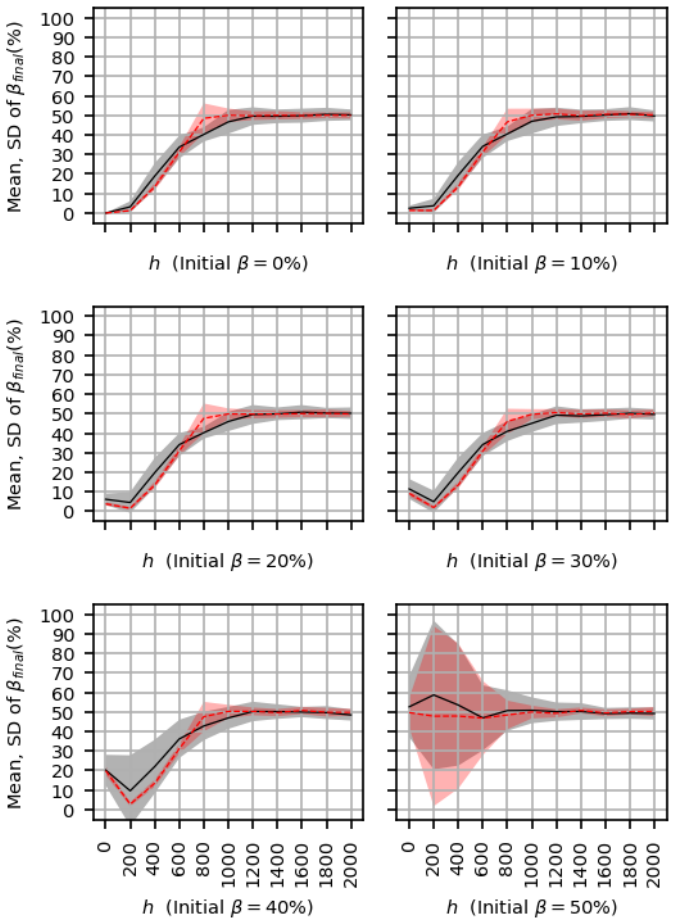


Fig. 17 The mean and (shaded) standard deviation of β_{final} for various β and noise levels for log-normal Bianconi-Barabási network (black) and scale-free Barabási-Albert network (red) topologies.

Increasing the noise level to 200 causes some segregated groups to break due to the agents' stochastic behavior. Therefore, in Fig. 12, more majority opinion groups emerge on average. Even starting from β=0%, it is less probable to reach a consensus phase. Moreover, the orderly fluctuation is less probable due to the stochastic behavior; therefore, the fluctuations are mostly non-orderly.

As the system noise level increases to 400 (Fig. 13) and 600 (Fig. 14), the non-orderly majority phase happens, but with less population in comparison with lower noise levels. In these cases, the more stochastic behavior of the agents causes some agents to change their opinion without a persuasiveness impact from interacting agents, and this spontaneous opinion changes may cause breaking the segregated groups the agents belong to. Therefore, more stochastic behavior causes less population of majority opinion group in the (non-orderly) majority phase.

Increasing the noise level to h=1000, as shown in Fig. 15, causes more domination of stochastic behavior, and the majority opinion group's population is not very discriminating from the population of the minority opinion group.

Finally, increasing the noise to higher levels, e.g., 2000, starting from any β values, the system reaches an equilibrium non-orderly non-majority phase, in which the population of both opinion groups is roughly the same with some small fluctuations.

Fig. 17 compares the system behavior starting from different β values with various noise levels for the log-normal network topology studied in this research (the black curves) and the scale-free Barabási-Albert network topology studied in [12] (the red curves). The networks in both studies have been generated using preferential attachment algorithms, starting from two nodes (m_0=2) and adding two edges (m=2) with every addition of a new node. For the scale-free network with power-law node degree distribution, the Barabási-Albert algorithm [16] has been used, which is very similar to the Bianconi-Barabási algorithm [63] we used for the log-normal networks. The networks of both topologies have the same number of nodes and the same number of edges; therefore, the mean value of node degrees are the same in both cases.

The comparison of simulation with both log-normal and scale-free network topologies is shown in Fig. 17. As the figure shows, starting from β=50% results in a non-majority phase (roughly β_{final}=50%) for both network topologies. The other starting β values are discussed as follows:

- For very high noise levels (>1200), a non-majority phase occurs (β_{final}=50%) in both topologies regardless of starting β values.
- For some lower noise levels (600< h <1200), β_{final} in scale-free topology is closer to 50% than log-normal topology. The reason is that in the scale-free topology with power-law node degree distribution, there are few highly connected nodes (strong hubs) which are more connected in comparison with highly connected nodes in the log-normal topology. The strong hubs in the scale-free topology in the presence of high enough noise cause more segregated groups break, and β_{final} is more closer to 50% in comparison to the log-normal topology. In other words, the stronger hubs in the scale-free network cause weaker majority phases than the same conditions in the log-normal topology.
- For even lower noise levels (0< h <600), the noise level is not enough to break many segregations. In this case, if a network is more segregated, it will more resist against opinion changes due to interactions. The mean value and standard deviation of EI indexes for measuring segregation of the N_{run} networks used for simulation sample runs are shown in Table 3. The table shows scale-free networks are more segregated at the starting point of the simulation runs than the log-normal networks due to more negative values of EI indexes. Therefore, since the scale-free networks are

more segregated than the log-normal networks, in the log-normal networks, β_{final} is closer to 50% compared to the scale-free networks. In other words, starting from the same β values, scale-free networks results in a more majority phase in equilibrium.

6- Conclusion

This research considered the phase transition due to the various noise levels in the social impact model of opinion formation in the log-normal network topology. Phase transition in the scale-free network topology, with power-law distributed networks, has been considered in previous studies[12]. Since some recent studies have revealed that the log-normal networks are more realistic models for real world social networks, we considered the log-normal network in this study on phase transition in the social impact model of opinion formation.

The results show that the segregation phenomenon is a main parameter affecting the phase transition for different noise levels, the level of the stochastic behavior of the social system. Two main phases are possible: the majority and the non-majority. The non-majority phase happens when there is high enough noise levels and causes approximately the same population of both possible opinions in the equilibrium. The majority phase occurs in lower noise and no-noise level, where the segregation of opinion groups inhibits consensus to occur, but the more populated opinion group becomes larger, and the less populated group shrinks down.

In this research, the log-normal network topology has also been compared with the scale-free network topology to study differences in phase transition behavior in both topologies based on EI index [52]. The experiments show that the scale-free networks are more segregated than the log-normal networks with the same number of nodes and edges. The more segregation in the scale-free topology causes more majority phases to occur in equilibrium in comparison with the log-normal topology, and weaker majority phases in some higher noise levels before enough high noise levels that cause non-majority phases.

Similar to many other studies in the complutational social science, the results of this study help us to understand some real world social behaviors. However, future studies may focus more on the various parameters assumed in this research, including persuasiveness and supportiveness strengths, determining individuals' leadership power. Furthermore, we assumed a fixed network topology during any simulation repetition; however, combining the opinion dynamics with the dynamics of the network structure (changing network links) could be a challenge to be studied in the future. Moreover, the results of similar studies using traditional sociological tools could be very worthy to be compared with the results of this study.

Table 3: The Mean Value(and the Standard Deviation) of EI Index for 30 Samples of Initial Log-Normal and Barabási-Albert Random Networks for Opinion Formation Simulation Repetitions

β	**Log-Normal Network**	**Barabási-Albert Network**
0	-1.00 (0.00)	-1.00 (0.00)
10	-0.62 (0.26)	-0.66 (0.03)
20	-0.34 (0.25)	-0.37 (0.04)
30	-0.12 (0.19)	-0.16 (0.03)
40	-0.03 (0.10)	-0.04 (0.02)
50	0.00 (0.02)	-0.01 (0.02)

References

[1] P. Hedström, and P. Bearman, "What is analytical sociology all about? An introductory essay," The Oxford handbook of analytical sociology, pp. 3-24, 2009.

[2] M. Keuschnigg, N. Lovsjö, and P. Hedström, "Analytical sociology and computational social science," Journal of Computational Social Science, vol. 1, no. 1, pp. 3-14, 2018.

[3] L. Mastroeni, P. Vellucci, and M. Naldi, "Agent-based models for opinion formation: A bibliographic survey," IEEE Access, vol. 7, pp. 58836-58848, 2019.

[4] B. D. Anderson, and M. Ye, "Recent advances in the modelling and analysis of opinion dynamics on influence networks," International Journal of Automation and Computing, vol. 16, no. 2, pp. 129-149, 2019.

[5] C. Castellano, S. Fortunato, and V. Loreto, "Statistical physics of social dynamics," Reviews of modern physics, vol. 81, no. 2, pp. 591, 2009.

[6] J. R. French Jr, "A formal theory of social power," Psychological review, vol. 63, no. 3, pp. 181, 1956.

[7] J. A. Hołyst, K. Kacperski, and F. Schweitzer, "Social impact models of opinion dynamics," Annual reviews of computational physics, vol. 9, pp. 253-273, 2001.

[8] B. Latané, "The psychology of social impact," American psychologist, vol. 36, no. 4, pp. 343-356, 1981.

[9] S. Hobolt, T. J. Leeper, and J. Tilley, "Divided by the vote: affective polarization in the wake of the Brexit referendum," British Journal of Political Science, 2020.

[10] M. Pineda, R. Toral, and E. Hernandez-Garcia, "Noisy continuous-opinion dynamics," Journal of Statistical Mechanics: Theory and Experiment, vol. 2009, no. 08, pp. P08001, 2009.

[11] L. P. Kadanoff, "More is the same; phase transitions and mean field theories," Journal of Statistical Physics, vol. 137, no. 5-6, pp. 777, 2009.

[12] A. Mansouri, and F. Taghiyareh, "Phase Transition in the Social Impact Model of Opinion Formation in Scale-Free Networks: The Social Power Effect," Journal of Artificial Societies and Social Simulation, vol. 23, no. 2, pp. 3, 2020.

[13] J. A. Hołyst, K. Kacperski, and F. Schweitzer, "Phase transitions in social impact models of opinion formation," Physica A: Statistical Mechanics and its Applications, vol. 285, no. 1-2, pp. 199-210, 2000.

[14] G. Jaeger, "The Ehrenfest classification of phase transitions: introduction and evolution," Archive for history of exact sciences, vol. 53, no. 1, pp. 51-81, 1998.

[15] M. Li, and H. Dankowicz, "Impact of temporal network structures on the speed of consensus formation in opinion dynamics," Physica A: Statistical Mechanics and its Applications, vol. 523, pp. 1355-1370, 2019.

[16] A.-L. Barabási, and R. Albert, "Emergence of scaling in random networks," science, vol. 286, no. 5439, pp. 509-512, 1999.
[17] T. Johansson, "Generating artificial social networks," The Quantitative Methods for Psychology, vol. 15, no. 2, pp. 56-74, 2019.
[18] A. D. Broido, and A. Clauset, "Scale-free networks are rare," Nature communications, vol. 10, no. 1, pp. 1-10, 2019.
[19] K. Sun, "Explanation of log-normal distributions and power-law distributions in biology and social science," Tech. Report, Department of Physics, 2004.
[20] C. Cioffi-Revilla, "Computational social science," Wiley Interdisciplinary Reviews: Computational Statistics, vol. 2, no. 3, pp. 259-271, 2010.
[21] P. Y.-z. Wan, "Analytical sociology: A Bungean appreciation," Science & Education, vol. 21, no. 10, pp. 1545-1565, 2012.
[22] N. Gilbert, and K. Troitzsch, Simulation for the social scientist: McGraw-Hill Education (UK), 2005.
[23] J. Hauke, I. Lorscheid, and M. Meyer, "Recent development of social simulation as reflected in JASSS between 2008 and 2014: A citation and co-citation analysis," Journal of artificial societies and social simulation, vol. 20, no. 1, 2017.
[24] E. Chattoe-Brown, "Why sociology should use agent based modelling," Sociological Research Online, vol. 18, no. 3, pp. 1-11, 2013.
[25] F. Bianchi, and F. Squazzoni, "Agent-based models in sociology," Wiley Interdisciplinary Reviews: Computational Statistics, vol. 7, no. 4, pp. 284-306, 2015.
[26] A. Jędrzejewski, and K. Sznajd-Weron, "Statistical physics of opinion formation: is it a spoof?," Comptes Rendus Physique, 2019.
[27] R. P. Abelson, "Mathematical models of the distribution of attitudes under controversy," Contributions to mathematical psychology, vol. 14, pp. 1-160, 1964.
[28] M. H. DeGroot, "Reaching a consensus," Journal of the American Statistical Association, vol. 69, no. 345, pp. 118-121, 1974.
[29] R. A. Holley, and T. M. Liggett, "Ergodic theorems for weakly interacting infinite systems and the voter model," The annals of probability, pp. 643-663, 1975.
[30] N. E. Friedkin, and E. C. Johnsen, "Social influence and opinions," Journal of Mathematical Sociology, vol. 15, no. 3-4, pp. 193-206, 1990.
[31] N. E. Friedkin, and E. C. Johnsen, "Social influence networks and opinion change," Advances in Group Processes, vol. 16, pp. 1-29, 1999.
[32] R. Axelrod, "The dissemination of culture: A model with local convergence and global polarization," Journal of conflict resolution, vol. 41, no. 2, pp. 203-226, 1997.
[33] K. Sznajd-Weron, and J. Sznajd, "Opinion evolution in closed community," International Journal of Modern Physics C, vol. 11, no. 06, pp. 1157-1165, 2000.
[34] D. Stauffer, A. O. Sousa, and S. M. De Oliveira, "Generalization to square lattice of Sznajd sociophysics model," International Journal of Modern Physics C, vol. 11, no. 06, pp. 1239-1245, 2000.
[35] G. Deffuant, D. Neau, F. Amblard, and G. Weisbuch, "Mixing beliefs among interacting agents," Advances in Complex Systems, vol. 3, no. 01n04, pp. 87-98, 2000.
[36] G. Deffuant, F. Amblard, G. Weisbuch, and T. Faure, "How can extremism prevail? A study based on the relative agreement interaction model," Journal of artificial societies and social simulation, vol. 5, no. 4, 2002.
[37] G. Deffuant, F. Amblard, and G. Weisbuch, "Modelling group opinion shift to extreme: the smooth bounded confidence model," arXiv preprint cond-mat/0410199, 2004.
[38] R. Hegselmann, and U. Krause, "Opinion dynamics and bounded confidence models, analysis, and simulation," Journal of Artificial Societies and Social Simulation, vol. 5, no. 3, 2002.
[39] S. Galam, "Minority opinion spreading in random geometry," The European Physical Journal B-Condensed Matter and Complex Systems, vol. 25, no. 4, pp. 403-406, 2002.
[40] C. Altafini, "Dynamics of opinion forming in structurally balanced social networks," PloS one, vol. 7, no. 6, pp. e38135, 2012.
[41] C. Altafini, "Consensus problems on networks with antagonistic interactions," IEEE transactions on automatic control, vol. 58, no. 4, pp. 935-946, 2013.
[42] C. Altafini, and G. Lini, "Predictable dynamics of opinion forming for networks with antagonistic interactions," IEEE Transactions on Automatic Control, vol. 60, no. 2, pp. 342-357, 2015.
[43] A. Nowak, J. Szamrej, and B. Latané, "From private attitude to public opinion: A dynamic theory of social impact," Psychological review, vol. 97, no. 3, pp. 362, 1990.
[44] A. Mansouri, F. Taghiyareh, and J. Hatami, "Improving Opinion Formation Models on Social Media Through Emotions," in 5th International Conference on Web Research (ICWR), 2019.
[45] A. Mansouri, F. Taghiyareh, and J. Hatami, "Post-Based Prediction of Users' Opinions Employing the Social Impact Model Improved by Emotion," International Journal of Web Research, vol. 1, no. 2, pp. 34-42, 2018.
[46] M. Golosovsky, "Power-law citation distributions are not scale-free," Physical Review E, vol. 96, no. 3, pp. 032306, 2017.
[47] A. Clauset, C. R. Shalizi, and M. E. Newman, "Power-law distributions in empirical data," SIAM review, vol. 51, no. 4, pp. 661-703, 2009.
[48] K. Binder, "Theory of first-order phase transitions," Reports on progress in physics, vol. 50, no. 7, pp. 783, 1987.
[49] A. Barrat, M. Barthelemy, and A. Vespignani, Dynamical processes on complex networks: Cambridge university press, 2008.
[50] P. Fronczak, A. Fronczak, and J. A. Hołyst, "Phase transitions in social networks," The European Physical Journal B, vol. 59, no. 1, pp. 133-139, 2007.
[51] M. Perc, "Phase transitions in models of human cooperation," Physics Letters A, vol. 380, no. 36, pp. 2803-2808, 2016.
[52] M. Bojanowski, and R. Corten, "Measuring segregation in social networks," Social Networks, vol. 39, pp. 14-32, 2014.
[53] A. Kowalska-Styczeń, and K. Malarz, "Noise induced unanimity and disorder in opinion formation," Plos one, vol. 15, no. 7, pp. e0235313, 2020.
[54] S. Grauwin, and P. Jensen, "Opinion group formation and dynamics: Structures that last from nonlasting entities," Physical Review E, vol. 85, no. 6, pp. 066113, 2012.

[55] M. Pineda, R. Toral, and E. Hernández-García, "Diffusing opinions in bounded confidence processes," The European Physical Journal D, vol. 62, no. 1, pp. 109-117, 2011.
[56] A. Carro, R. Toral, and M. San Miguel, "The role of noise and initial conditions in the asymptotic solution of a bounded confidence, continuous-opinion model," Journal of Statistical Physics, vol. 151, no. 1-2, pp. 131-149, 2013.
[57] J. Zhang, and Y. Zhao, "The robust consensus of a noisy deffuant-weisbuch model," Mathematical Problems in Engineering, vol. 2018, 2018.
[58] L. Sabatelli, and P. Richmond, "Non-monotonic spontaneous magnetization in a Sznajd-like consensus model," Physica A: Statistical Mechanics and its Applications, vol. 334, no. 1-2, pp. 274-280, 2004.
[59] K. Sznajd-Weron, "Sznajd model and its applications," arXiv preprint physics/0503239, 2005.
[60] W. Su, G. Chen, and Y. Hong, "Noise leads to quasi-consensus of Hegselmann–Krause opinion dynamics," Automatica, vol. 85, pp. 448-454, 2017.
[61] G. Chen, W. Su, S. Ding, and Y. Hong, "Heterogeneous hegselmann-krause dynamics with environment and communication noise," IEEE Transactions on Automatic Control, 2019.
[62] M. Pineda, R. Toral, and E. Hernández-García, "The noisy Hegselmann-Krause model for opinion dynamics," The European Physical Journal B, vol. 86, no. 12, pp. 490, 2013.
[63] G. Bianconi, and A.-L. Barabási, "Competition and multiscaling in evolving networks," EPL (Europhysics Letters), vol. 54, no. 4, pp. 436, 2001.
[64] T. Pham, P. Sheridan, H. Shimodaira, M. T. Pham, and I. Rcpp, "Package 'PAFit'," 2020.

Alireza Mansouri is a faculty member of ICT Research Institute (ex ITRC: Iran Telecommunication Research Center). He received his BSc and MSc from Sharif University of Technology, both in Computer Engineering- Software and his Ph.D. in Computer Engineering- Information Technology from University of Tehran. His research interests include computational social science, social networks, agent-based modeling, and Internet of Things.

Fattaneh Taghiyareh is associate professor of Computer Engineering- Software and Information Technology, at the University of Tehran, where she has served since 2001. She received a Ph.D. in Computer Engineering- Parallel Algorithm Processing from the Tokyo Institute of Technology in 2000. Her current research involves "Opinion formation in social networks", "Semantic web", "Social learning", and "Agent based modeling". She is a member of Editorial Board of *International Journal of Information & Communication Technology Research.*

Drone Detection by Neural Network Using GLCM and SURF Features

Tanzia Ahmed
Department of Computer Science and Software Engineering, Concordia University, Montreal, QC, Canada
tanzia.ahmed@mail.concordia.ca
Tanvir Rahman
Department of Computer Science and Engineering, Brac University, Dhaka, Bangladesh
tanvir.rahman@bracu.ac.bd
Bir Ballav Roy
Department of Computer Science and Engineering, Brac University, Dhaka, Bangladesh
roybirballav@gmail.com
Jia Uddin*
Technology Studies Department, Endicott College, Woosong University, Daejeon, South Korea
jia.uddin@wsu.ac.kr

Received: 21/Jan/2020 Revised: 20/Jan/2021 Accepted: 26/Feb/2021

Abstract

This paper presents a vision-based drone detection method. There are a number of researches on object detection which includes different feature extraction methods – all of those are used distinctly for the experiments. But in the proposed model, a hybrid feature extraction method using SURF and GLCM is used to detect object by Neural Network which has never been experimented before. Both are very popular ways of feature extraction. Speeded-up Robust Feature (SURF) is a blob detection algorithm which extracts the points of interest from an integral image, thus converts the image into a 2D vector. The Gray-Level Co-Occurrence Matrix (GLCM) calculates the number of occurrences of consecutive pixels in same spatial relationship and represents it in a new vector- 8 × 8 matrix of best possible attributes of an image. SURF is a popular method of feature extraction and fast matching of images, whereas, GLCM method extracts the best attributes of the images. In the proposed model, the images were processed first to fit our feature extraction methods, then the SURF method was implemented to extract the features from those images into a 2D vector. Then for our next step GLCM was implemented which extracted the best possible features out of the previous vector, into a 8 × 8 matrix. Thus, image is processed in to a 2D vector and feature extracted from the combination of both SURF and GLCM methods ensures the quality of the training dataset by not just extracting features faster (with SURF) but also extracting the best of the point of interests (with GLCM). The extracted featured related to the pattern are used in the neural network for training and testing. Pattern recognition algorithm has been used as a machine learning tool for the training and testing of the model. In the experimental evaluation, the performance of proposed model is examined by cross entropy for each instance and percentage error. For the tested drone dataset, experimental results demonstrate improved performance over the state-of-art models by exhibiting less cross entropy and percentage error.

Keywords: Feature Extraction; GLCM Method; Image Processing; Neural Network; SURF Algorithm.

1- Introduction

Drones are, in technical terms, unmanned aircraft. These are also known as unmanned aerial vehicles, or UAVs. Usually they are controlled by remote controlling systems, however, there are some drones that can fly by themselves. While there are numerous merits of drones that be uttered here, drones are also being used in several crimes these days. Crimes are being sophisticated day by day and this is just one angle of it. For instance, it is used for providing unwanted materials to prison [Telegraph, February 16, 2016]. FBI spokesperson says, drug cartel activists are being replaced by drones as there is little fear of getting arrested [1].

Image processing in the field of object detection is getting momentums, and for good reasons. The drones that are being used for crimes need to be detected while they are still in the air and prior to the time crime takes place. Traditional CCTV based security system is in cry of update, hence image processing can come in handy. For that, techniques of image processing have to be chosen carefully for extracting and analyzing features of the objects.

Object detection is one of the major divisions in the field

* Corresponding Author

of computer vision and there are many researches that have been conducted in this area. Object detection has been done using deep learning. The Support Vector Machine (SVM) was used for extracting feature in the field of emotion recognition [2], and also used for dimension reduction in the fields of machine learning [3]. Yan *et. al* detected anomalies using SVM [4]. SVM outperforms then the other state-of-art models in the field of object recognition using entropy theory [2-4]. Blob detection has also been quite popular in the field of feature extraction. Barbosa *et al.* showed a tool to extract meta-data for game sprite using Blob detection aka edge detection. Neural Networks are very sensitive, even to the lowest change of any properties of an object [5]. Often times, it may lead to inefficient generalization of their results. Tay and Lao present how the use of SVM leads to inefficient generalization in the field of financial time series forecasting [6].

Image processing has come a long way from detecting the overall features of a particular image, it is now a field that often works to detect the distinct features. However, this development has its own challenges. Different image processing systems have several challenges as to how they should be analyzed. Kumar and Bhatia showed how to use Fourier analysis to analyze the shapes. Moreover, they also showed the importance of Gray Scale character in both rotation variant and rotation invariant [7]. However, all distinct information may not be relevant. Choras showed how the relevant information can detect and identify the similar images in the field of biometrics. In [8], Content Based Image Retrieval (CBIR) is used for detecting the similarities in the images. Some basic architecture of ZF-NET, and deep normalization and convolutional layers (DNCNN) may use for automatic extracting features. Yin *et al.* suggested a model that shows the previous phenomenon [9]. In [10], authors present an analysis of various ways to recognize the aerial components from images taken by drones on power transmission lines using the neural network and SURF (Speeded-up Robust Features) and BoW (Bag-of-Words) methods as a feature extraction. In [11], Abuzneid *et. al* proposed an enhanced technique of face recognition using traditional methods like back-propagation neural network (BPNN) and feature extraction by correlation between training images of T-Dataset and BPNN.

Binary filtering and Circular Hough Transform (CHT) have been used for circular object detection. Firstly, they filtered the background and after that a gray scale filter is used to prepare the dataset for binary filter and circular Hough transform. It is successful in detecting an object but CHT may not exactly detect the circular object as sometimes it is connected with other object and give an inaccurate result [12]. Color offers potent object recognition data. Swain and Ballard's model is straightforward reconnaissance scheme is the representation of matching images based on RGB histogram [13]. This color-based recognition method has been extended by Funt and Finlayson and to get extensive flexibility they introduce illumination by indexing on a light-invariant color range [14].

Alex, Ilya and Geoffrey used a very modern technique to train up data's in neural network [15]. Their method of deep convolutional neural network has the capability of doing significant computational power. Their analysis also gives us an insight that regardless of the complication of the dataset it can achieve good result using supervised learning. But subsequent reduce of one layer hinders the performance and accuracy too.

LeCun, Yann, Fu Jie Huang, and Leon Bottou.in their paper points out the lack of flexibility and resource minimization of template-based approaches. Their proposed model is more feature extractable and robust [16].

Chae *et al.*, in their paper "A Wearable sEMG Pattern-Recognition Integrated Interface Embedding Analog Pseudo-Wavelet Preprocessing" have presented a wearable wireless surface electromyogram (sEMG) integrated interface that utilizes a proposed analog pseudo-wavelet preprocessor (APWP) for signal acquisition and pattern recognition [17].

Zupan, in his paper of "Introduction to Artificial Neural Network (ANN) Methods: What they are and how to use them" has explained the selection procedure of training dataset. He has emphasized on this step as to be very important and suggested to divide the dataset into not two but three datasets. According to him, the first dataset should be for training, the second one should be the control set or fine-tuning set and lastly, the third one should be the test dataset. He also suggested that the training dataset should be smaller in size than the test dataset. He also mentioned that the true test set should contain completely 'non-committal' or unbiased set of data [18].

Zupan, when trying to explain artificial neural network simply, he compared it to a black box having multiple input and multiple output which processes large number of parallelly connected simple arithmetic units. ANN methods work best when they are dealing with non-linear dependence between the inputs and outputs.

Youtang and Jianming, in their paper of "Air Target Fuzzy Pattern Recognition Threat-Judgment Model" have tried to establish a threat judgement model that has high reliability in air defense systems in the naval warships. They have used fuzzy pattern recognition model to identify threats from air targets. They have classified the threat degrees considering target distance, type, speed, advent time, cross-point distance and flight altitude. They have theoretically measured the threats using the parameters. For example, distance threat membership function has the feature that the threat degree is inversely proportional to

the distance between targets and warships. Therefore, they described it as a descending ridge distribution [19].
This paper explains the model we came up with after experimenting a number of methods in terms of detecting drones using Neural Network tool. Our goal was to find a better and faster way to detect the object (drones) which leads to better performance and lesser error. Here, we used a hybrid feature extraction method using the SURF and GLCM features which is utilized for detecting a drone by Neural Network. SURF method is popularly used for its faster image matching property and GLCM extracts the best features of a set of images. Thus, combining the two methods we were able to form a dataset that gave us our desired result. The complete process of combining the methods have been explained in proposed model section.
Rest of the paper is organized as follows. Chapter II presents the proposed model, details result analysis is in Chapter III. Finally, conclude the paper in Chapter IV.

2- Proposed Model

The proposed model is a hybrid model where both SURF and GLCM methods have been used to extract features from the input and target datasets. Then the newly created dataset has been fed into SCG function of neural network to obtain the output set.
Four different versions have been developed in order to achieve a better result on the basis of cross entropy and percentage of error. They include applications of various methods for feature extraction such as, MSER, SURF, GLCM and SURF and GLCM combined. Different versions gave different results but the best model that was observed for both SURF and GLCM feature extraction algorithms. At first, images for training were pre-processed, then feature extraction algorithms, SURF and GLCM were applied to extract attributes of drones. This dataset was fed into the neural network for training and testing. From the output, it was possible to analyze the performance and error percentage of the model. The model is described step by step with a flow chart in Fig. 1.

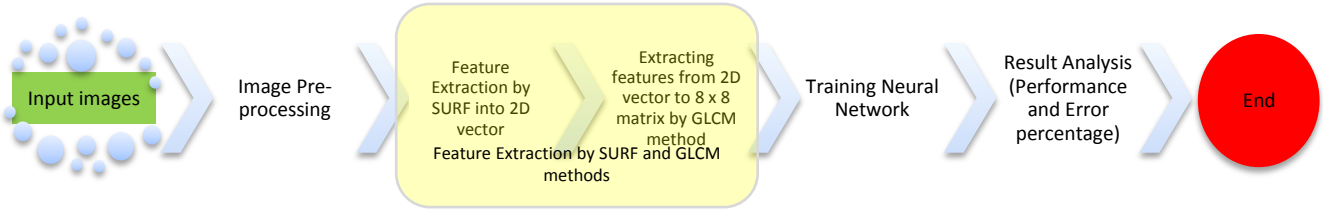


Fig. 1 Flow chart of the hybrid model of drone detection.

2-1- Image Pre-Processing

About 600 images of drones were collected and resized to 227 × 227 pixels. Among the total images, 75% were used for training and rest of the 25% for testing. All the images were converted into a uniform size. There are barely any scholarly articles to explain the reason for resizing the images to exactly same dimensions. Although, Nikhil *et. al.* mentioned that many Neural Network models expect or assume input images to be square shaped, therefore, images have to be reshaped or cropped [20]. The input images were of true color, having clear sky in the background so that no attributes of other components interrupt the feature extraction of the drones.

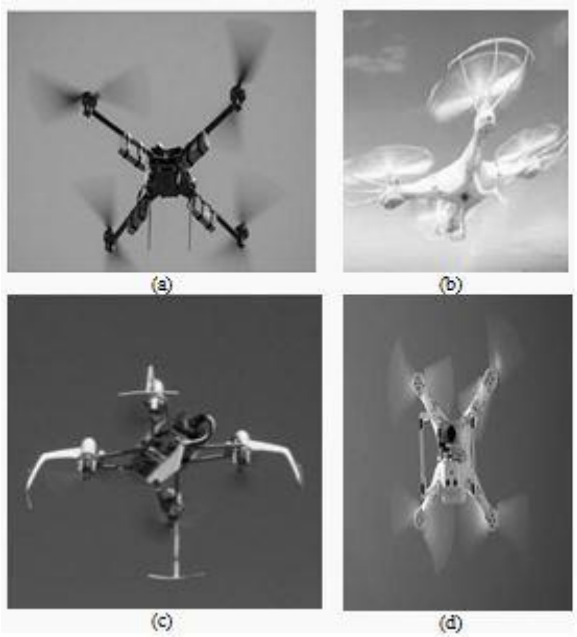


Fig. 2 (a), (b), (c) and (d) are the gray-scale images of drone #79, #98, #100 and #113, respectively.

Here in Fig. 2, the gray-scale or 2D images of few sample drones are shown. The test dataset was also of true color. At first, all the input (training) images have been taken randomly. Then each image was converted from true color or 3D to gray-scale or 2D image. 3D image is of RGB scale and consists of 3D numeric array and similarly, 2D image consists of 2D numeric array. The conversion removes the hue and saturation of the image keeping the luminance intact and returns a 2D array of double values. The method rgb2gray has been used for the conversion which is done by calculating a weighted sum of Red, Green and Blue. This follows an algorithm which is, 0.2989*R + 0.5870*G + 0.1140*B, where the values of R, G and B of a pixel are multiplied with their respective specific co-efficient and then summed together to provide a gray-scale pixel corresponding to that true color pixel [21]. Each value of the 2D array generated from this algorithm is in the range of 0 to 1, it can be positive or negative. The pixels with values greater than 0 are displayed as white and the pixels that are equal to 0 or less than 0 are displayed as black [22]. Similarly, the test image datasets are converted to gray-scale images one by one.

The 2D arrays were rotated to 90° angle for both datasets right after RGB to Gray-scale transformation. The rotation was required because in this model GLCM algorithm was applied as one of the feature extraction methods, where an offset had to be fixed that depends on the angle of rotation. This offset was later delivered to graycomatrix method [23].

Fig. 3 shows some of the images of drones which were used to train our network.

Fig. 3 Sample images of drones for the training dataset.

2-2- Feature Extraction by SURF and GLCM Methods

Since it is a hybrid model, the feature extraction took place in two steps. At first, attributes were extracted from the images using SURF algorithm. Then the 2D array of double values achieved by concatenating the features set of all the images was used to populate into GLCM method to obtain the best features, better performance and minimal error percentage. The steps are described elaborately below:

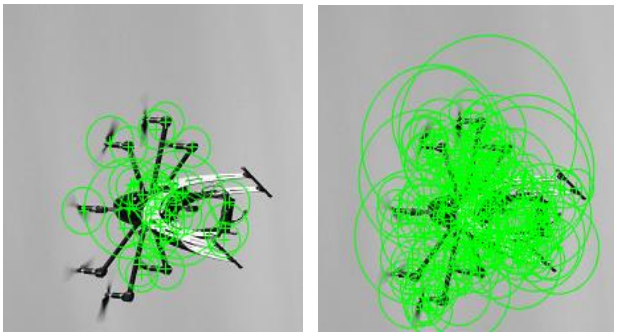
Speeded-up Robust Features (SURF) is a blob detection algorithm which means it detects the corners of the object and the locations where the reflection of light is higher (light speckles) [24]. This method is popularly used because of faster calculation of interest points due to use of integral images and it can detect locations best where there is illumination [25]. There are three main steps to this algorithm - interest point detection, local neighborhood description and matching. Firstly, the interest points are calculated using Hessian matrix. They can be found at different scales as the algorithm uses comparison images and the corresponding interest points can be found in different levels. To resolve this issue, Gaussian filter is used that smoothed the images repeatedly. Then they are subsampled to get the next level of the hierarchy of the pyramid (scale space) [26-29]. Since images of drones were taken from different scales, filters had to be used and the faster method to do that was provided by SURF algorithm. The levels are calculated by:

$$\sigma = \left(current\ filter\ size \times \frac{base\ filter\ scale}{base\ filter\ size} \right) \quad (1)$$

If p(x,y) is point in an image and σ is the scale where the Hessian matrix is H(p, σ) and $L_{ij}(p, \sigma)$ is the convolution of the second order derivative of Gaussian, then-

$$H(p,\sigma) = \left(L_{xx}(p,\sigma)\ L_{xy}(p,\sigma)\ L_{yx}(p,\sigma)\ L_{yy}(p,\sigma) \right) \quad (2)$$

Secondly, the local neighborhood descriptor is to be detected. Descriptors provide unique and robust features by describing the intensity or the orientation of the pixels. They are computed from the local neighborhoods of the interest points. To extract descriptors, a circular region around the point of interest of radius 6S is used, where S is the scale of the point. Then a square region is constructed around it aligned to the orientation to obtain scale invariance. Haar wavelet responses in horizontal and vertical directions are calculated within this squared region for each sample point [26], [27-30]. Finally, the descriptors are compared for matching among the images and the common points are taken as the matched attributes of the images [26, 28].



(a) (b)

Fig. 4 (a) Rotated 2D drone feature detected by SURF-strongest 30 points, (b) All possible points

Thus, the original image is fragmented into several squared regions for the use of integral image and then summed up for faster calculations [26-29]. To sum up the integral images, it uses equation (3), where I(i,j) is the interest point of the image I.

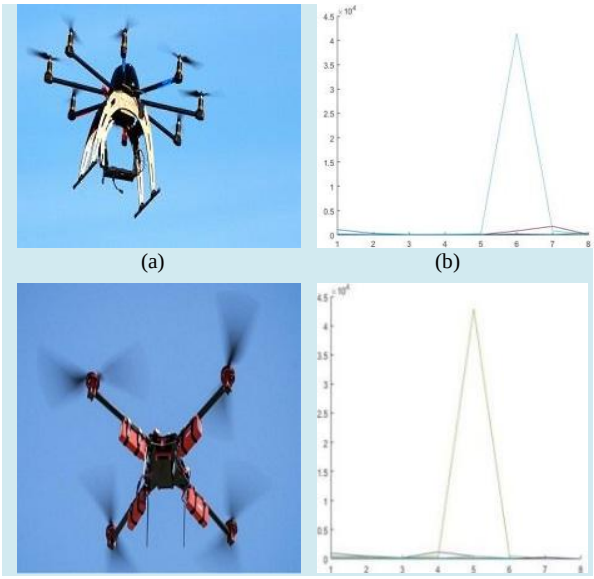
$$S(x.y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i,j) \quad (3)$$

Fig. 4 represents the extracted features of a rotated drone by SURF method. In Fig. 4(a) only 30 strongest features are detected and in Fig 4(b) all the features are detected. In this study, all the features have been used. After incorporating the processed image datasets into SURF extraction method, it was possible to obtain the best features out of each image which is represented by a 2D array. The features set of each image have double type values with a size of $n \times m$.

Once the features are collected in a 2D array, it is fed into GLCM algorithm along with proper offset. Gray-Level Co-occurrence Matrix (GLCM) is generated by calculating the number of times a pixel with the gray-level intensity value at *i* occurs in a specific spatial relationship with the pixel j[m], where, Q(x,y) = i and Q(x+1, y+1) = j when diagonally right pixels are considered, and Q(x+1, y+1) = j when horizontal neighboring pixel is considered. Here, x and y are offsets [23, 31-33]. The equation is determined by using the dimension of the offset matrix. An 'Offset' is the distance between a pixel of interest and its neighbor. It is a p×2 matrix, where p is the number of pairs that pixels of interest make with their neighbors [23]. By default, it is [0 1]. The graycomatrix function takes two parameters where the image points and the offset are used. For image points the feature set obtained from SURF feature extraction method is taken and for the second parameter [2 0] offset is populated and this is the reason why the images are rotated to 90-degree angle in prior [23]. [2 0]-offset means that the sequence of pair of adjacent pixels which is to be considered (as feature), lies in 2 consecutive rows of the same column. The size of GLCM matrix is determined by number of gray-level intensities which is by default 8. It usually returns an n × n matrix of extracted features. For this study, the function returned an 8 × 8 matrix that means the best 64 features were obtained for each image [31]. The equation (equation 4) for calculating GLCM features is given below [33-34]:

$$C_{\Delta x,\Delta y}(i,j) = \sum_{x=1}^{n} \sum_{y=1}^{m} \{1, if\ i\ and\ j\ true\ 0, if\ i\ and\ j\ false \quad (4)$$

Fig. 5 represents the extracted features graph of sample images of drones before pre-processing and their corresponding gplots of GLCM feature extraction method.



(a) (b)

(c) (d)

Fig. 5 (a) Drone #10 shown in RGB before pre-processing, (b) corresponding plot of features extracted by GLCM after pre-processing of (a), (c) drone #79 shown in RGB before pre-processing, (d) corresponding plot of features extracted by GLCM after pre-processing of (c).

The extracted feature set for each image is converted to 1D array. Then each of these arrays is arranged in another parent array which is the final dataset for training in the Neural Network. This is the newly created hybrid dataset that have to be populated in Scale Conjugate Gradient (SCG) function of the neural network. The separate datasets for training and testing are incorporated in the network, each having $n \times 64$ size 2D array where n is the number of images.

The dataset obtained have the final features of an image. It is arranged in a 2D array. This dataset is used for training the neural network.

2-3- Training Neural Network

A neural network is a collection of connected nodes called the artificial neurons loosely modeled like the neuron connections of the brain [35]. Like the biological neurons, the artificial neurons receive signal (input), combines it with their internal state (activation) and an optional threshold using an activation function and signal other neurons connected to it. The final output finishes the task, such as recognizing an object in the image. The important characteristic of the activation function is that it provides a smooth and differentiable transition as input value changes.

The network consists of connections, each connection provides an output of one neuron as an input to another. Each connection is assigned a weight that represents its relative importance [36].

Artificial neural network was chosen for the proposed model's dataset training. As a machine learning tool, neural network for pattern recognition algorithm has been used for this model.

It is a fully connected neural network which is open to various customization. It uses the basic equation of modelNN = learnNN(X, y) for training, and p = predictNN(X_valid, modelNN) for prediction. There is a chance for an arbitrary number of layers and different activation functions. We used an arbitrary number of layers and the activation function was set to default [42].

Pattern recognition is the algorithm which identifies or classifies object based on their key features [37]. For its fast and optimum classification method, it is not only used for object identification but also used in the fields like speech recognition, text classification, and radar processing. The classification by pattern recognition can be both supervised and unsupervised. Supervised classification is the one where classifiers are created from different object classes. On the other hand, unsupervised classification is the method where hidden structures or patterns are identified within the unlabeled data using segmentation and clustering techniques.

Since the aim of the study is to identify drones from unclassified images, we have used the unsupervised classification method of Pattern Recognition tool. The pattern of the images had to be trained to the system's network, so that on testing it could determine the drones with optimal performance and accuracy. Pattern recognition algorithm matches all the inputs' features with test images' features and try to calculate how much alike they are, considering their statistical variation [38]. And pattern recognition, when implemented with neural network, resolves complex recognition in real time. Real time response is what we need in case of a drone is identified in the clear sky. Moreover, neural network is well known for its adaptive learning which other tools offer less. No wonder, the leading companies like DeepMind, Google AI, Facebook uses neural network as a machine learning tool. The datasets prepared in earlier step, are passed to the network which have 10 neurons or hidden layers and trained by 'trainscg' function (which uses SCG algorithm) suitable for low memory usage [39]. Scale Conjugate Gradient (SCG) Backpropagation function is an algorithm with superlinear convergence rate. It requires $O(n)$ space complexity, where n is the number of weights in the network, therefore, it is suitable for the system also to get a faster result [29]. SCG is evaluated considering 3 algorithms' performance as standard they are – Backpropagation algorithm (BP), the Conjugate Gradient Propagation (CGP), and the one-step Broyden-Fletcher-Goldfarb-Shanno memoryless quasi-Newton algorithm (BFGS). The speed-up of SCG depends on convergence criterion. If the demand for reduction in error is more, the speed-up will be boosted. SCG is user independent unlike CGP and BFGS, and the weight complexity also favors SCG in terms of showing long ravines in sharp curvature than BP where the ravines are short. Therefore, the overall performance of SCG is better than other training functions considering the low memory space and that is why it has been chosen as the training function of this network [40].

The network took training dataset and trained itself to recognize the pattern of the images of drones. Then by testing with the test dataset, it learned as well as gave output to the number of drones it could detect. The system, however, cannot determine the type of the drone but can identify drones and differentiate between other aerial objects – the output will show greater cross-entropy. The goal is to find the better method to extract features for training the system and it is possible to come up with a better algorithm.

This proposed model is using the best of two already very popular models. It was proven that extracting blob features into a 2D array was necessary. Hence the usage of SURF came into action, however, detecting a drone is a different matter altogether. Hence the GLCM method was thought of.

3- Result Analysis

In the results analysis, we have considered the performance and error percentage of the network. The performance is calculated by cross-entropy per epoch; the minimum is the cross-entropy, the better is the performance [41]. If the system takes all the properties into account then the performance will be 0. If it does not take any properties into account then the performance will be 100. Low performance mean the system works with the high number of properties when it runs the algorithm. Our

focus is to have this performance as low as we can, that means we wanted to take higher number of properties while detecting the drones. The percentage of error is calculated as,

$$percentError = \left(\sum (tind \cong yind) \right) / numel(tind) \quad (5)$$

Here tind is a 2D target vector indices and yind is 2D output vector indices. While running the algorithm our model leaves some portion of the dataset that is to say we cannot consider their properties; that portion is our percent error. We have to keep that low as much as we can. Now we give performance preference over percent error; because performance deals with all the properties and percent error deals with portion of the data set. If we do not take all the properties into account of a dataset, it does not matter how big our dataset is.

3-1- Comparative Analysis

Table 1 presents a comparative performance comparison of proposed hybrid models with other state-of-art models for our tested drone dataset.

Table 1 Comparative analysis of state-of-art models

No.	*Version No.*	*Performance*	*Error percent*
1.	MSER [v.3]	29.09	3.5
2.	GLCM Method [v.4]	1.88e-16	89
3.	SURF Feature [v.5]	30.5	2.34
4.	Proposed Model [v.6]	7.34e-17	33

With the trend analysis in Fig. 6, we can assume that SURF feature with GLCM is the better way to detect drones while it is in the air. This way, we can detect the drones with a minimum amount of time and less complexity; that too with accepted error percentage rate. Here, one question may arise how 33% error is better than having 2.34% error. The answer to the question is, it is not better. However, the performance is better when we use the proposed hybrid model. Moreover, 33% of error means the system leaves 33% of the input dataset while matching. It is acceptable because it still is compared with 67% of the dataset where we know all of the pictures are of drones.

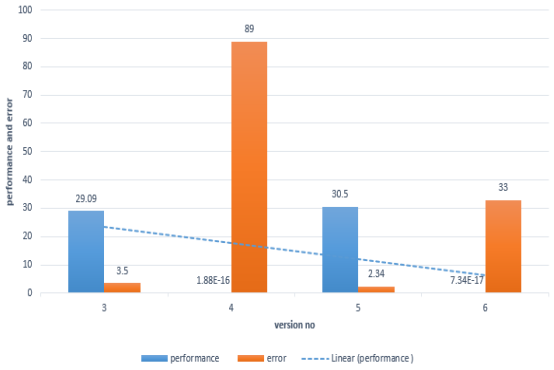


Fig. 6 Comparative analysis with state-of-art models by side with trend analysis.

Besides, we are doing it with all possible extracted features. We are giving priorities to the extracted features rather than how many of the dataset does it go through while comparing them and it does not leave too many of the dataset either in the proposed model. Therefore, the proposed hybrid model is better. Fig. 6 shows the trend analysis of performance and error percentage of all the four state-of-art models.

4- Conclusions

This paper presented a hybrid model to detect drones by extracting the attributes from the image dataset provided by SURF feature and populating the extracted information by GLCM algorithm. This dataset is fed into the network that uses scale conjugate gradient algorithm to recognize the pattern of the drones. The SCG function makes the system faster to detect the desired components. As a result of this model, the system is able to capture any kind of image of drones in the air and it can identify those with as much accuracy as possible and as fast as it can while it remains in the sky. Although, there are various modern neural networks like Alex-Net, ZF-Net, VGG Net, etc., we have chosen to provide a better way for drone detection using the traditional methods and tools for higher performance and lower percentage error. In addition, the proposed model exhibits better results than the state-of-art models.

Acknowledgments

This research is funded by Woosong University Academic Research in 2021.

References

[1] M. Hicks, "Criminal Intent: FBI Details How Drones are used in crime," Techradar-the source for tech buying advice, May 2018. [online]. https://www.techradar.com/news/criminal-intent-fbi-details-how-drones-are-being-used-for-crime.

[2] F. P. George, I. M. Shaikat, P. S. F. Hossain, M. Z. Parvez, and J. Uddin, "Recognition of emotional states using EEG signals based on time-frequency analysis and SVM classifier," International Journal of Electrical and Computer Engineering, 2019, vol. 9, no. 2, pp. 1012-1020

[3] R. Dong, H. Meng, Z. Long and H. Zhao, "Dimensionality reduction by soft-margin support vector machine," IEEE International Conference on Agents (ICA), Beijing, China, 2017, pp. 154-156.

[4] G. Yan, "Network Anomaly Traffic Detection Method Based on Support Vector Machine," 2016 International Conference on Smart City and Systems Engineering (ICSCSE), Zhangjiajie, Hunan, China, 2016, pp. 3-6.

[5] M. d. Barbosa, C. d. Barbosa, and A. F. Barbosa, "MuSSE: A Tool to Extract Meta-Data from Game Sprite Sheets Using Blob Detection Algorithm," 14th Brazilian Symposium on Computer Games and Digital Entertainment (SBGAMES), Piauí, Brazil, 2015, pp. 61-69.

[6] F. E. H Tay and L. Lao, "Application of support vector machines in financial time series forecasting," omega, vol. 29, no. 4, Aug. 2001, pp. 309-317.

[7] G. Kumar, P. K. Bhatia, "A detailed Review of Feature Extraction in Image Processing Systems," Fourth International Conference on Advanced Computing and Communication Technologies, IEEE Computer Society, Washington DC, USA, 2014, pp. 5-12.

[8] R. S. Choras, "Image Feature Extraction Techniques and Their Application for CBIR and Biometric Systems," International Journal of Biology and Biomedical Engineering, 2007, vol. 1, no. 1, pp. 6-16.

[9] Z. Yin *et al.*, "A Deep Normalization and Convolutional Neural Network for Image Smoke Detection," IEEE Access, vol. 6, 2018, pp. 4287-4296.

[10] J. Chen *et al.*, "Analysis of the recognition and localization techniques of power transmission lines components in aerial images acquired by drones," The Institute of Engineering and Technology Journals, IEEE Access, 2017, pp. 29-32.

[11] M. A. Abuzneid and A. Mahmood, "Enhanced Human Face Recognition Using LBPH Descriptor, Multi-KNN, and Back-Propagation Neural Network," IEEE Access, vol. 6, 2018, pp. 20641-2065.

[12] R. Hussin, M. R. Juhari, N. W. Kang, R. C. Ismail, A. Kamarudin, "Digital Image Processing Techniques for Object Detection from Complex Background Image," Procedia Engineering, 2012, pp. 340–344.

[13] M. J. Swain and D. H. Ballard, "Color indexing," International Journal of Computer Vision, vol. 7, no. 11, 1991, pp. 11-32.

[14] B. V. Funt and G. D. Finlayson, "Color constant color indexing," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 5, May 1995, pp. 522-529.

[15] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Neural Information Processing Systems, 2012, pp. 1-9.

[16] M. Nakib, R. T. Khan, M. S. Hasan and J. Uddin, "Crime Scene Prediction by Detecting Threatening Objects Using Convolutional Neural Network," International Conference on Computer, Communication, Chemical, Material and Electronic Engineering, Bangladesh, 2018, pp. 1-4.

[17] H. Y. Chae, K. Lee, J. Jang, K. Park, and J. J. Kim, "A Wearable sEMG Pattern-Recognition Integrated Interface Embedding Analog Pseudo-Wavelet Preprocessing," IEEE Access, 2019, vol. 7, pp. 151320-151328.

[18] J. Zupan, "Introduction to Artificial Neural Network (ANN) Methods: What They Are and How to Use Them," Acta Chimica Slovenica, 1994, vol. 41, no. 3, pp. 327-352.

[19] T. Youtang and W. Jianrning, "Air target fuzzy pattern recognition Threat-judgment model," Journal of Systems Engineering and Electronics, 2003, vol. 14, no. 1, pp. 41-46.

[20] B. Nikhil, "Ïmage Data Pre-Processing for Neural Networks," Becoming Human: Artificial Intelligence Magazine, 2017.

[21] rgb2gray-Convert RGB image or colormap to grayscale, MathWorks, v: R2018a, 2018. [online]. https://www.mathworks.com/help/matlab/ref/rgb2gray

[22] mat2gray-Convert matrix to grayscale image, MathWorks, v: R2018a, 2018. [online]. https://www.mathworks.com/help/images/ref/mat2gray.

[23] R.M. Haralick and L.G. Shapiro, "Computer and Robot Vision," vol. 1, Addison-Wesley, 1992, pp. 1-459

[24] A. Xu and G. Namit, "SURF: Speeded-up Robust Features," 2008, Project Report: McGill University.

[25] T. Das, R. Hasan, M. R. Azam and J. Uddin, "A Robust Method for Detecting Copy-Move Image Forgery Using Stationary Wavelet Transform and Scale Invariant Feature Transform," International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), Bangladesh, 2018, pp. 1-4.

[26] G. S. Rabbani, S. Sultana, M.N. Hasan, S. Q. Fahad, J. Uddin, "Person identification using SURF features of dental radiograph," 3rd International Conference on Cryptography, Security and Privacy, 2019, pp. 303

[27] Detect SURF Features-Detect SURF features and return SURF Points object MathWorks, v: R2018a, 2018. [online]. https://www.mathworks.com/help/vision/ref/detectsurffeatures.

[28] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, "SURF: Speeded Up Robust Features," Computer Vision and Image Understanding, 2008, vol. 110, no. 3, pp. 346–359.

[29] E. Oyallon and J. Rabin, "An Analysis of the SURF Method," 2015, Image Processing On Line (IPOL), pp.176-218.

[30] B. Fan, Z. Wang, F. Wu, "Local Image Descriptors: Modern Approaches," Springer, 2015, vol. 12, pp. 1-99.

[31] R. M. Haralick, K. Shanmugan, and I. Dinstein, "Textural Features for Image Classification," IEEE Transactions on Systems, Man, and Cybernetics, 1973, vol. SMC-3, pp. 610-621.

[32] A. Uppuluri, "GLCM texture features- Calculates texture features from the input GLCMs," version:1.2.0.0, MathWorks, v: R2018a, 2018. [online].

https://www.mathworks.com/matlabcentral/fileexchange/22187-glcm-texture-features.
[33] P. Cosman, "Gray-Level Co-occurrence Matrices (GLCMs)," [online]. http://www.code.ucsd.edu/pcosman/glcm.pdf.
[34] J. Uddin, R. Islam, J. M. Kim, "Texture Feature Extraction Techniques for Fault Diagnosis of Induction Motors," Journal of Convergence, 2014, vol. 5, no. 2, pp. 15-20.
[35] Image Recognition- Recognition methods in image processing, MathWorks, v: R2018a, 2018. [online]. https://www.mathworks.com/discovery/pattern-recognition.
[36] "The Machine Learning Dictionary," available at: www.cse.unsw.edu.au. Retrieved at 4 November 2009.
[37] A. Zell, "chapter 5.2," Simulation neuronaler Netze [Simulation of Neural Networks] (in German) (1st ed.), Addison-Wesley, 2003.
[38] C. M. Bishop, "Pattern Recognition and Machine Learning," Springer, 2006, pp. 1-758
[39] Classify Patterns with a Shallow Neural Network, MathWorks, v: R2018a, 2018. [online]. https://www.mathworks.com/help/nnet/gs/classify-patterns-with-a-neural-network.
[40] M. F. Moller, "A Scale Conjugate Gradient Algorithm for Fast Supervised Learning," Neural Networks, 1993, vol. 6, no. 4, pp. 525-533.
[41] Crossentropy- Neural network performance, MathWorks, v: R2018a, 2018. [online]. https://www.mathworks.com/help/nnet/ref/crossentropy.
[42] V. Tshitoyan (2021). Simple Neural Network (https://github.com/vtshitoyan/simpleNN), GitHub. Retrieved January 20, 2021.

Tanzia Ahmed received the Bachelor of Science in Computer Science and Engineering degree from BRAC University, Dhaka, Bangladesh in 2019. She has experience on Software Analysis and Software Development for about two years. Currently, she is a master's candidate of Concordia University, Canada. She is enrolled in the program of Master of Applied Computer Science under Department of Computer Science and Software Engineering. Her research interests include Machine Learning, Image Processing, Computer Interface and Software Design and Analysis.

Tanvir Rahman received the B.Sc. degree in Computer Science and Engineering from BRAC University, Dhaka, Bangladesh in 2018 and M.S. degree in Computer Science and Engineering from BRAC University, Dhaka, Bangladesh in 2021. Currently He is a full-time lecturer in BRAC University, Dhaka, Bangladesh. His research interests include Information retrieval, Forecasting, Algorithms, Machine Learning and Data mining.

Bir Ballav Roy received the B.Sc. degree in Computer Science from BRAC University, Dhaka, Bangladesh in 2019. Currently he is working as a Software Engineer in a data centric software Firm. He is a data enthusiast and likes to get insights from data using machine learning models and Deep Learning and Statistical Models. His research interest includes Data Mining, Computer Vision, NLP, Artificial Intelligence and Machine Learning.

Jia Uddin is an Assistant Professor of Technology Studies Department in Endicott College, Woosong University, Daejeon, South Korea. He is an associate professor and undergraduate coordinator (On leave) in the department of Computer Science and Engineering, BRAC University, Dhaka, Bangladesh. He did Ph.D. in Computer Engineering from University of Ulsan, South Korea and M.Sc. in Electrical Engineering emphasis on Telecommunications from Blekinge Institute of Technology, Sweden. His research interest includes Multimedia Signal Processing, Fault Diagnosis, Bangla Language Processing, IoT based Intelligent System Design.

Confronting DDoS Attacks in Software-Defined Wireless Sensor Networks based on Evidence Theory

Reyhane Hoseini
Computer Engineering Department, Imam Reza International University, Assar St., Daneshgah St., Mashhad, IRAN,
reyhane_hoseini71@yahoo.com
Nazbanoo Farzaneh*
Computer Engineering Department, Imam Reza International University, Assar St., Daneshgah St., Mashhad, IRAN
Nazbanou.farzaneh@imamreza.ac.ir

Received: 30/May/2020 Revised: 26/Dec/2020 Accepted: 05/Apr/2021

Abstract
DDoS attacks aim at making the authorized users unable to access the network resources. In the present paper, an evidence theory based security method has been proposed to confront DDoS attacks in software-defined wireless sensor networks. The security model, as a security unit, is placed on the control plane of the software-defined wireless sensor network aiming at detecting the suspicious traffic. The main purpose of this paper is detection of the DDoS attack using the central controller of the software-defined network and entropy approach as an effective light-weight and quick solution in the early stages of the detection and, also, Dempster-Shafer theory in order to do a more exact detection with longer time. Evaluation of the attacks including integration of data from the evidence obtained using Dempster-Shafer and entropy modules has been done with the purpose of increasing the rate of detection of the DDoS attack, maximizing the true positive, decreasing the false negative, and confronting the attack. The results of the paper show that providing a security unit on the control plane in a software-defined wireless sensor network is an efficient method for detecting and evaluating the probability of DDoS attacks and increasing the rate of detection of an attacker.

Keywords: Software- Defined Wireless Sensor Networks; Distributed Denial of Service; Entropy; Dempster-Shafer Theory; Evidence Theory.

1- Introduction

A wireless sensor network is made of several wireless nodes able to collect data in non-accessible areas in which human interference could be impossible. In wireless sensor network, there exist numerous limitations because sensor nodes have limited processing power, energy, storing, and bandwidth in wireless links, potential of failure [1, 2].
Software-defined network has been developed as a promising and modern mechanism in improvement of the network. A central software program, called controller, generally manages the network behavior. The software-defined network controller is able to add, update, and remove a flow. Every reaction is actively done in response to information packets using pre-defined rules. Therefore, the software-defined network would be able to quickly react to security threats and traffic filtering, and determine dynamic security policies [3-7].
The software-defined network has emerged as a solution and has been integrated with the wireless sensor network offering it as a modern technology called SDWN (software-defined wireless network). SDWN is not resistant against new attacks due to the physical separation between control plane and data plane. SDWN model uses the software-defined network technique in order to solve many basic problems in the wireless sensor networks; SDWN is facing many challenges though [2-4, 6, 8-18]. In SDWN, the controller sends policies and orders the transportation devices how to face the flows [3, 4, 8-10, 14, 16, 17].
Security is vital for any network; SDWN is not an exception. However, security in SDWN is still in its early stages. Some security solutions can be applicable to SDWN, some cannot. Hence, solving the SDWN security problems is a challenge. Moreover, DDoS attacks are among common attacks in software-defined networks based on wireless sensor networks. DDoS attacks grow in the SDWN environment considering the characteristics of such networks and remain one of the greatest security concerns [3, 15, 16, 18-26].
The present paper, with the help of a security framework on the controller, detects the DDoS attacks which could occur on switches or different nodes. In the proposed method, by a security structure, the suspicious traffic is detected, DDoS attacks are detected, and attacks are

* Corresponding Author

stopped from entering the network main part. The present paper proposes a method for dealing with DDoS attacks in the controller plane using an entropy-based method as a quick and light-weight detection mechanism in its early stages, the history of the flows, and Dempster-Shafer theory for a more exact detection.

The present paper is organized as follows: section 2 offers a literature review. Section 3 provides a brief explanation of the basic concepts of the proposed method. Section 4 is about the proposed method and its importance in the DDoS attacks detection mechanism and the process of detecting such attacks. Section 5 deals with the simulation and evaluation of the proposed method for detecting the DDoS attacks in SDWN environment. It also discusses the simulation results. Finally, section 6 offers the conclusion and the future works areas.

2- Research Background

[15] has offered some solutions to distinguish the slow DDoS attack traffic from lawful traffic. However, nobody has distinguished flash attacks from quick DDoS attacks which are more common. They have used the information theory based on general entropy.

[18] has offered a systematic investigation of various kinds of DoS attacks in software-defined network, and has offered MLFQ to deal with the attack; it allows the queue to develop dynamically and integrate regarding the controller load. [21] has proposed Bohatei method which is a defense method against DDoS using software-based network and NFV. In designing Bohatei, the key takes the address into consideration which is related to scalability, responding, and resisting against attacker. Bohatei simply uses its resources management section to stand against various DDoS attacks; using a few network resources, it can effectively resist attacks.

[26] has offered a defense mechanism against DDoS attacks called CoFence which facilitates the domain-helps-domain cooperation. CoFence is a collaborative network for resistance against DDoS attacks based on network functions virtualization in which the under-attack domain can send the excessive traffic to other collaborative domains for filtering. Specifically, this paper focuses on resource allocation mechanism. It, to allocate the resources, uses the multi-leader-follower Stackelberg game in order to collaborate fairly and share the resources.

Few authors use statistical methods for detecting attacks in software-defined networks. [22] proposed a solution based on entropy. The main purpose of this method is to enable itself to detect the attack up to 500 traffic packets. To do so, it uses the central controller of the software-defined network to detect the attacks. A suggestion for detecting such attacks based on entropy changes is using the destination IP. [24] offers solutions for dealing with DDoS attacks. It uses the flow entropy combined with average entropy technique for detecting attacks. [25] has proposed a mechanism to track DDoS attacks which, according to entropy changes, is between normal traffic and DDoS traffic, basically different from normal traffic. Tracking is done through information theoretical parameters. [23] offers a method similar to techniques like entropy-based system and system anomaly detection for detecting DDoS attacks and preventing them. To investigate a lot of data flow in such environment, a multi-thread IDS approach is proposed. Calculation of entropy for the packets is done using IP address, ports, and flow size. The method presented in [27] used delivery ratio and control packets overhead to detect DDoS attacks in SDWNs. The change point (CP) detection algorithm is used to detect an attack. Method [28] uses several modules to detect the attack and counter the attack in SDIoT networks. If the number of traffic flows exceeds a threshold, the algorithm detects the attack and tries to identify the source of the traffic. If traffic flows are sent from one source, then the attack is definite and that source is blocked. Proposed method in [29] is able to detect an attack in either centralized or distributed mode. The centralized detector has great recognition rates and can differentiate the type of the attacks. The distributed detector offers information that lets to recognize the nodes beginning the attack.

In [30], the method of common intrusion detection is determined for detecting attacks in a cloud. It is a preventive model in which the responsibility of the cloud elements management is distributed among several supervision nodes. In order to detect the common intrusion, Depmster-Shafer evidence theory has been used through the cloud broker role. [31] has proposed a new light-weight trust mechanism called TEDS to detect and increase the effects of Blackhole attacks. The new idea is a combination of entropy function and Dempster-Shafer theory to gain validity for a node. If the validity of a node is less than the threshold amount, it is put on the black list and separated from the network. In [32] Dempster-Shafer theory and combined evidences are used to detect the internal attacker by detection mechanism with the help of neighboring node parameters in wireless sensor network.

[19] focuses on the detection and analysis of DDoS attacks in the environment of cloud calculations. The proposed solution is combining the evidences gained from intrusion detection systems. In the virtual machines, cloud systems are used along with data fusion method. A method has been proposed using a quantitative solution for detecting and analyzing the DDoS attacks in the environment of cloud calculations with the help of Dempster-Shafer theory.

3- Basic Concepts

In this section, the concept of Entropy and Dempster-Shafer theory used in this paper are explained

3-1- Entropy

Entropy or Shannon-Wiener is a significant concept in information theory which measures the amount of uncertainty in the network by an accidental variable or data. The amount of hidden entropy is $[0,\log_m]$ where m is the number of accidental elements [33].
Considering an accidental processing, entropy rate H (X) is calculated from two accidental processes using eq. 1.

$$H(x) = -\sum_{i=1}^{n} Pi \log Pi \quad (1)$$

Entropy H (X) takes an accidental variable X with possible amounts {X1, X2, …, Xn} with the probability of {P1, P2, …, Pn} [33]. And the conditions of eq. 2 are correct for Pi s'.

$$0 \le P_i \le 1 \quad ; \quad \sum_{i=1}^{n} P_i = 1 \quad (2)$$

3-2- Dempster-Shafer Theory

Dempster-Shafer theory is a popular theory used in modeling and reasoning at the time of uncertainty or lack of precision in smart systems. Dempster combination rule is a powerful device used in combining evidences from different information resources. It is used in evaluating the risk and trust capability in engineering issues when the exact measurement of experiments and gaining knowledge from experts' inferences is impossible. A significant aspect of this theory is the combination of evidences gained from various resources and modeling the contrast among them which allows us to reach a belief degree (which is known via a mathematical object called belief function) [32, 34].
Suppose θ is a finite set of elements; an element can be a hypothesis, an aim, or a situation of a system. θ is called frame of discernment. The power set of θ is determined by Ω(θ). For example, if $\theta = \{a,b,c\}$, the amount of Ω(θ) is defined in eq. 3 [35].

$$\Omega(\theta)=\{\{\emptyset\},\{a\},\{b\},\{c\},\{a,b\},\{a,c\},\{b,c\},\{a,b,c\}\} \quad (3)$$

Empty set that shows the flawless system situation $A = \{a,b\}$is a subset of θ : A□θ . A states the system flaw in a or b. θ states the system flaw in a, b, or c.
There are three important functions in this theory which are the base of calculations and equations. They are

- Likelihood function (m)
- Belief function (Bel)
- Likelihood function (Pl)

The probability of the occurrence of the predicate is shown by mass function which is briefly called m and defined as eq. 4 [35].
The mass m(A) of A, a given member of the power set, expresses the proportion of all relevant and available evidences that support the claim that the actual state belongs to A and to no particular subset of A. The value of m(A) pertains only to the set A and makes no additional claims about any subsets of A, each of which has, by addition its own mass.

$$m: \Omega(\theta) \to [0,1], \quad \sum_{i}^{n} m(A) = 1 \quad , \quad m(\theta) = 0 \quad (4)$$

Bel(A) function measures the amount of all probability that should be in elements of A which means certainty about A belief and is the lower bounds on the A probability. The belief function is defined as eq. 5 [35].

$$Bel\ (A): \Omega(\Theta) \to [0,1], \quad Bel\ (A) = \sum m(B) \quad (5)$$

Pl(A) function measures the maximum amount of probability that can be distributed among the elements of A which describes the total belief degree related to A and is the upper bound on A probability. It is defined as eq. 6 [35].

$$Pl\ (A): \Omega(\Theta) \to [0,1], \quad Pl\ (A) = 1 - Bel(A) = \sum_{B \cap A \neq \emptyset} m(B) \quad (6)$$

Where A is the intersection of subsets B and C.

4- The Proposed Method

In this paper, a defense mechanism against DDoS attacks is proposed for the controller. The proposed security model is placed on the controller plane. The purpose is to offer a platform with the help of software-defined network technology as a mechanism for detecting suspicious traffic and attacks and confronting them. Also, the other purpose is to offer a method for providing security on the control plane in order to stop the DDoS attacks from entering the network and, also, to take care of the whole network and the controller as its main part due to being the single point of failure

4-1- The Network Model

The network model of the proposed method is shown in Fig. 1. More explanation about the network model will follow.
SDWN is made of numerous sensor nodes and a sink node. The proposed architecture includes data plane and control

plane. Data plane is made of some wireless sensor nodes; the main part of the network is on the control plane. A set of these nodes connect to the surrounding environment and send the collected information to the controller. Therefore, in the proposed architecture, the connection among sets made of nodes is taken into consideration in which the controller affected by DDoS attack receives malicious traffic flow from these nodes.

For connecting the nodes to the controller, a secure channel is considered in a way that has the smallest delay and the highest trust capability. Secure channel is used for sending control messages and transportation rules from the controller to the sensor nodes, info alert, and changing the message topology from sensor nodes to controller.

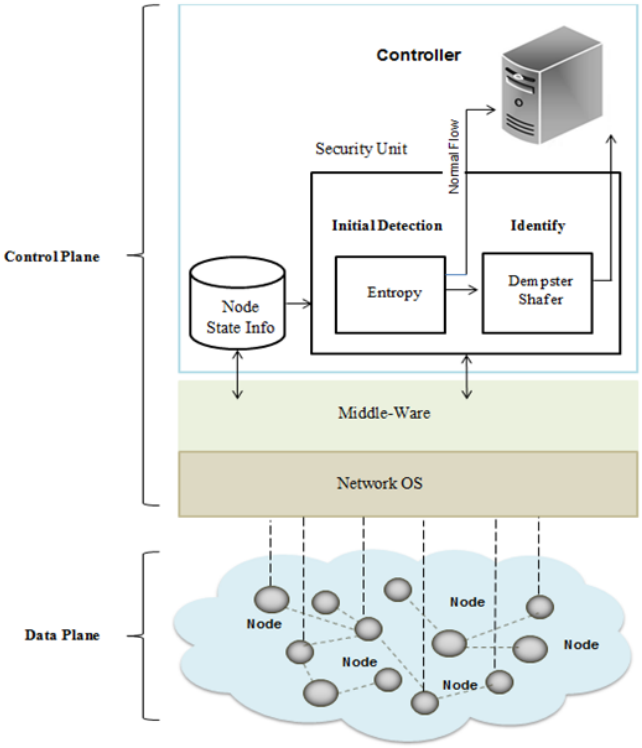


Fig. 1: the network model in the proposed method

Every controller is made of the following components:

Middle-ware

This module is responsible for analyzing and extracting data and updating the database.

Information storage unit

This unit stores the information about the nodes. Such information includes the node number, the node kind, situation, the energy left, etc. When the packets enter the middle-ware of the network, the middle-ware extracts the node information and updates or registers it in the database.

Security unit

Sensor nodes send the flows with various data to the controller. Every traffic flow, after arriving at the control plane and before arriving in the controller, enters the security module to be processed and investigated by this module. Security module is made of the two following parts:

- **First Module:** early detection of suspicious traffic (via entropy module)
- **Second Module:** detecting and confronting the attack (via Dempster-Shafer module)

4-2- The Hypotheses

Bearing in the mind that the main purpose of this research is detecting malicious flows and offering a security model for detecting DDoS attacks, the following hypotheses exist:

- The logical unit of the control plane is considered a part of the sink.
- It is supposed that in this network (software-defined wireless sensor network) the network nodes are fixed.
- The channel is supposed to be secure.
- The controller has to manage the nodes; in fact, the controller can generally view the whole network via flow tables.
- The controller is supposed to be secure, and the flows sent from sensor nodes could be malicious.

4-3- The Details of the Proposed Method

The security unit is made of two main parts. The aim of devising two parts is a more exact detection of attacks. In the first module, considering the input flow and usage of entropy, the traffic flows are classified. Then, suspicious flows are sent to the second module for more investigation. Fig. 2 shows the security unit functioning.

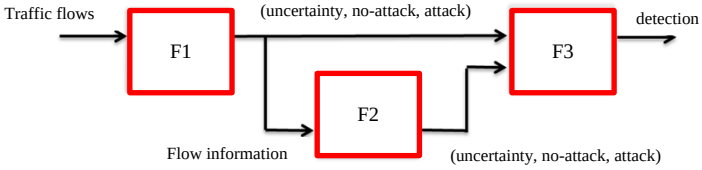


Fig. 2: security unit datagram

As seen,

- F1 is Entropy function:
  - ✓ Input: various traffic flows from certain nodes
  - ✓ Security first-level function: the entropy module measures the entropy value for each input flow

- ✓ Output: a percentage of probability in a certain range (attack, no-attack, uncertainty)
- F2 function (the history of flows)
  - ✓ Input: entropy module output
  - ✓ Function: the information of the flows and the sender of each flow are kept in a memory.
  - ✓ Output: a set of information in n number of various rounds with a percentage of probability (attack, no-attack, uncertainty)
- F3 Dempster-Shafer function:
  - ✓ Input: the information gained from the entropy output and the history of flows
  - ✓ Security second-level function: combination of views and offering the final view about the attack flows
  - ✓ Output: determining the lawfulness or maliciousness of the flow (a percentage of the attack probability, no-attack, uncertainty)

In the following, these two levels of security will be explained in more details.

4-3-1- Entropy: The first level of security

In the first level, the entropy method has been used in the software-based wireless sensor network to detect DDoS attacks. Before the controller is completely attacked, a quick and effective method is needed to work at the control plane. The main reason of choosing entropy used for detecting DDoS attacks is its ability to do accidental measurement in the flows entering the network. However, to stop the overuse of the processing power, the attack detection method has to be light-weight while the attack is occurring.

The entropy measures the probability of an event considering the total number of the events. To detect DDoS attacks, three phases are considered: flow input, traffic analysis, determining whether the flow is malicious or not. The entropy receives the input flows from different nodes and investigates them in a short time. In the simulation process, the flow entropy is used as a measurement tool for detecting attacks because it is efficient and trustworthy. The flow entropy defines the traffic distribution based on certain characteristics such as resource address, destination address, flow ID, etc. It is a significant parameter in traffic analysis. It is important to distinguish lawful and unlawful flows for detecting attacks. To track malicious flows, single flows are calculated and analyzed.

The entropy can, by processing the header of each flow and collecting the flow statistics, quickly detect the suspicious traffic. If the entropy finds a suspicious attack, it can be measured by the entropy.

In DDoS attacks, a great number of fake packets are sent from a group of hosts to the controller. These fake packets can occupy the controller resources and ruin them due to constant processing. Entropy can measure the received packets.

According to the formula, entropy in eq. 1, measures the amount of the flows disorder. In this regard, the entropy keeps 3 amounts for each flow including attack probability, no-attack probability, and uncertainty. The probability amounts in the entropy output determine whether the flow is malicious or lawful.

After the system's early investigation and according to the simulations done in it, a threshold amount is chosen for the entropy. The suitable threshold is chosen through simulation. If the entropy amount is less than the threshold, we consider it as attack flow; if more than the threshold, as lawful flow and normal input flow. Also, the numbers between these two intervals are considered as the probability of uncertainty. Programming is one of the main advantages of software-defined network so when the network construction changes, threshold could be readjusted.

For uncertainty, weighted average has been used as shown in eq. 7.

$$AL(t) = \alpha \times L(t) + \sum_{j=1}^{T} (1-\alpha)^{j} \times L(t-j) \qquad t \geq 1 \quad , \quad 0 \leq \alpha \leq 1 \tag{7}$$

Where AL(t) is the uncertainty weighted average at the time of t, and L is the uncertainty value for various flows. α is the effect and weight of the L values in the past rounds; the older (less) and closer to zero L is, the less effective on the present AL it would be. t is the present time. T is the number of the rounds based on which the weighted average is considered.

After coming to the final conclusion by the entropy plane, the normal flows that have been considered as no-attack enter the main part of the controller, and the first level of security sends the flows considered as attack and uncertain to the second level for more investigation in order to be exactly analyzed by Dempster-Shafer theory and make sure that the suspicious flows detected by entropy have been attacked or not.

In addition to investigation of the flow by the entropy, a history of the flows in n number of the rounds is kept in a memory which is called M2. We calculate the history of the flows for investigating each flow and each sender. In this history, the nodes function and the flows sent by each node in the recent rounds are kept. Hence, the information of various nodes and the amount of malicious node detection and the flows produced by that are registered.

The received information is sent to the second module (Dempster-Shafer) as the output of this section.

Using the history of the flows, as a method in data combination, increases the precision. Entropy measurements and flows history help find the attack resources and stop the DDoS attacks from entering.

4-3-2- Dempster-Shafer: The second level of security

Dempster-Shafer theory is an effective solution for evaluating the probability of DDoS attacks. The purpose of offering this module is having more time for traffic analysis and a more exact detection of attack and confronting it. Based on Dempster-Shafer theory, the information received is combined using different combination rules coming to a final conclusion.

Dempster-Shafer combination rules are shown in the following equations. The combination amount of m(a) is gained from integration of two basic assignment probabilities (m1, m2) and could be calculated according to eq. 8.

$$m(A) = \frac{1}{1-K} \sum_{B \cap C = A} m1\,(B) \times m2(C)$$
$$K = \sum_{B \cap C = A} m1\,(B).\ m2\,(C) > 0 \qquad (8)$$
$$m\,(\emptyset) = 0$$

Where K represents the basic probability mass associated with conflict, and (1-K) denominator in eq. 8 is the normalizing factor.

According to Dempster-Shafer combination method (Monte Carlo), different views will be investigated and, finally, three amounts will be received as output (attack probability, no-attack, and uncertainty). The module results indicate that DDoS attack has been applied onto the input flow, and the malicious flow and lawful flow are detected.

The controller, using the Dempster-Shafer results, decides upon the input flow. If the flow is detected as malicious, it puts it on the black list and, after investigating its function, removes the malicious node from the network.

The proposed method is made of the following stages:

- ✓ **Stage one:** the flows sent from certain senders enter the entropy plane and are calculated and measured by this plane. If the entropy is less than the minimum threshold, the flow enters stage two. Otherwise, the flow is considered lawful and sent to the controller.
- ✓ **Stage two:** the flows history is calculated.
- ✓ **Stage three:** the information taken from the entropy output based on attack and uncertainty is combined with history of flows by the second level of security (Dempster-Shafer) in order to gain a total probability of attack and uncertainty.
- ✓ **Stage four:** the malicious flow is detected and blocked.

The proposed algorithm attempts to be applied with optimum design without nested loop or complicated functions. The application time of the proposed algorithm is O(n) where n is the number of input flows.

The present paper, using a combination of Dempster-Shafer theory and entropy approach, has proposed a method for finding the uncertainty interval for various situations in systems. One of the reasons of using Dempster-Shafer is its theoretical development among non-traditional theories of investigating uncertainty. Also, the combination of different evidences gained from various resources is another advantage.

On the other hand, entropy is one of the ways of measuring uncertainty in finite sets of evidences using their probability distribution function; it can state the incompatibility among evidences resources probable distributions. Therefore, we expect these two methods along with the flows history to be able to detect the changes in traffic behavior of such events and affect the increase of uncertainty.

Using the mentioned levels of security, in addition to detection of the attack, the malicious node could, also, be detected ending in the prevention of the attack. In fact, in the first level, the traffic flows are quickly investigated on-line; in the second, the suspicious flows are analyzed more exactly, in a longer time, and off-line.

4-4- An Example of the Proposed Method

In this section, an example is provided to understand the proposed method more. For instance, the traffic flows including 20 various flows exist from wireless sensor nodes (ex: there are 4 sensor nodes called X1, X2, X3, and X4). We suppose that X2 and X3 are malicious and they send flows containing DDoS attacks sent to the controller. These flows are investigated in the security unit.

In the first level of security, the entropy module, measures the disorder amount of the input flows according to eq. 1 formula; attack probability is 0.7, no-attack is 0.2, and uncertainty is 0.02. The flows which are detected as normal by the entropy are sent to the controller; the flows which are detected as suspicious (attack and uncertainty) are sent to the second level of security for more investigation.

The flows history registers and saves the information and the flows senders, as follows, in a memory.

$$\begin{bmatrix} X3 & 0\,1\,0 \\ X2 & 0\,1\,0 \\ X2 & 0\,0\,1 \\ X3 & 0\,1\,0 \\ X1 & 1\,0\,0 \end{bmatrix}$$ Round 1

$$\begin{bmatrix} X3 & 0\,1\,0 \\ X4 & 1\,0\,0 \\ X2 & 0\,1\,0 \\ X3 & 0\,1\,0 \\ X1 & 1\,0\,0 \end{bmatrix}$$ Round 2

$$\begin{bmatrix} X3 & 0\,1\,0 \\ X2 & 0\,1\,0 \\ X2 & 0\,1\,0 \\ X4 & 1\,0\,0 \\ X4 & 1\,0\,0 \end{bmatrix}$$ Round 3

$$\begin{bmatrix} X3 & 0\,1\,0 \\ X2 & 0\,1\,0 \\ X1 & 1\,0\,0 \\ X3 & 0\,1\,0 \\ X1 & 1\,0\,0 \end{bmatrix}$$ Round 4

As you can observe, for each flow and each sender, the information is registered in the flows history. The attack result is [0 1 0], no-attack is [1 0 0], and uncertainty is [0 0 1]. The flows history keeps 5 stages of input flows in a memory, and when a new flow enters, according to circular shift, replaces the first input flow; finally, the view average of these 4 stages with the a amount of 0.7 (it means that the past flows had less effect on the final results of flows history) as the probability (attack, no-attack, and uncertainty) is considered as output; here, the attack probability is 0.8, and no-attack is 0.2; according to the registered information, X2 and X3 are the nodes attacked by DDoS and send malicious flows. This detection could be the result of over-population or other factors, so these flows are sent to the second level of security for more exact detection and results.

Attack and no-attack results in entropy and the flows history are considered as Dempster-Shafer module input. Dempster-Shafer combines the various views according to combination rules and comes to a complete and final conclusion about malicious flows.

As shown in Table 1, according to Dempster-Shafer theory, after combining the views, attack probability is 0.93, no attack probability is 0.065, and uncertainty is 0. Therefore, DDoS attack caused by these flows is detected. The controller can decide upon these flows and block them.

Table 1. Dempster- Shafer results in the example of the proposed method

	Likelihood amount pl_i		Belief amount bel_i		Mass functionality	
	Ent*	His**	Ent	His	Ent	His
Attack	0.781	0.8	0.77	0.8	0.77	0.8
No-attack	0.219	0.2	0.21	0.2	0.21	0.2
Uncertainty	0.998	1	0.99	1	0.002	0
Attack probability: 0.93 No-attack probability: 0.065 Uncertainty probability: 0 * Ent = Entropy ** His = History						

5- Simulation and Evaluation of the Proposed Method

To simulate, MATLAB has been used. Controller's characteristics is Intel(R) Core(TM) i7 CPU 1.80 GHz, 12GB RAM, windows 10, 64 bit. KDD CUP 99 has been used to send data from various nodes to the controller in the evaluation environment. This database includes a standard set of data investigated by us; it includes numerous simulated intrusions.

In this database 9998 flows exist. Normal flows include 7793 flows and malicious ones include 2205 in this dataset. To each flow, one sender (among 50 wireless nodes) is attributed. Traffic flow from different security unit nodes on the control plane enters the entropy module in order to be processed and investigated by this module in a short time and on-line. Entropy measures the attack probability, no-attack probability, and uncertainty for each flow at a high speed and in 1.56 ms. Table. 2 shows the practical parameters in various functions in the proposed method.

Table 2: Introduction of practical parameters

Parameter name	***Description***
M1	Entropy output
M2	Flow history
DS	Dempster_shafer output
W	Features
a	Suitable features
A	Attack probability
N	No-attack probability
AN	Uncertainty probability
H	Weighted average
Bel_i	Belief amount
Pl_i	Likelihood amount

Each input flow contains 40 features. We, here, have considered 4 of them as more weighted in comparison with others as shown in Fig. 3.

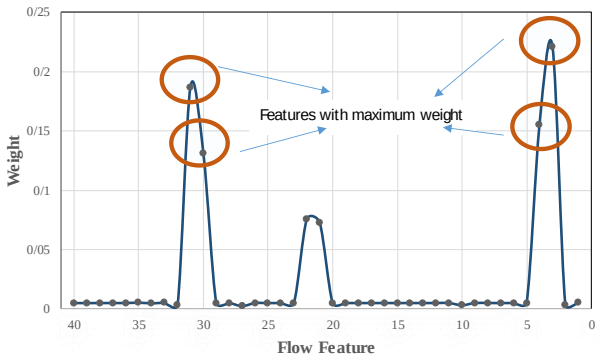


Fig. 3: Flow features weight

As observed in the figure, 40 features are shown on the horizontal pivot. The vertical pivot shows the weight of each feature. The amounts 31, 30, 4, and 3 which have more weight in comparison with others are considered the best features. Every flow containing the above 4 features enters the entropy module for investigation from certain senders (wireless sensor nodes). The entropy amount for each flow is calculated by eq. 1.

A threshold has been considered for the entropy. The minimum is considered 0.2; the maximum is considered 300. This number has been calculated according to the attacks detection rate via trial and error. Amounts less than min threshold are considered as attack probability and amounts more than max threshold as no-attack. The numbers between these two intervals which are less than 300 and more than 0.2 are considered as uncertainty probability. Fig. 4 and Fig.5 show the results of calculating different amounts of threshold.

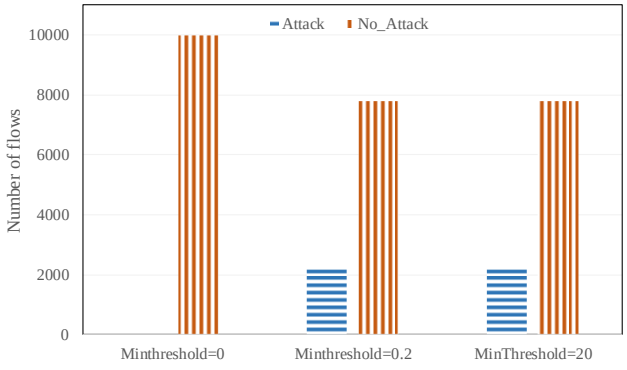


Fig. 4: Calculation of Min Threshold

As observed in Fig. 4, the horizontal pivot is considered as attack, no-attack, and uncertainty. To find the min threshold, 0, 0.2, and 20 are considered. 0 is not suitable for threshold because it cannot correctly detect the attacks. 20 is also not good because it may show more attack probability and more false positive. 0.2 is the best because the number of the attacks it has correctly detected is mainly true positive with less false positive.

Moreover, for max threshold, 3 different amounts are investigated as shown in Fig. 5. Among these amounts, 300 is the best for max threshold because of having less false positive.

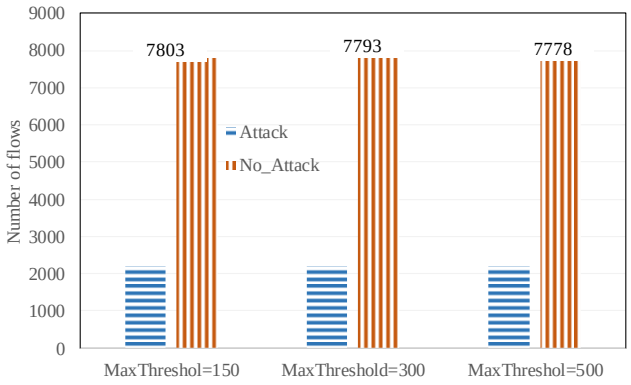


Fig. 5: Calculation of Max Threshold

To calculate the uncertainty eq. 7 is used as shown in Fig. 6. Weighted average calculates the uncertainty for the recent 10 rounds. The weighted average rate (α) is considered 0.5, that is, numbers gained in the past and present are equal. Amounts more than 0.5 mean that the past flows are less effective on final results. In Fig. 6, the amounts of weighted average for different α amounts are shown.

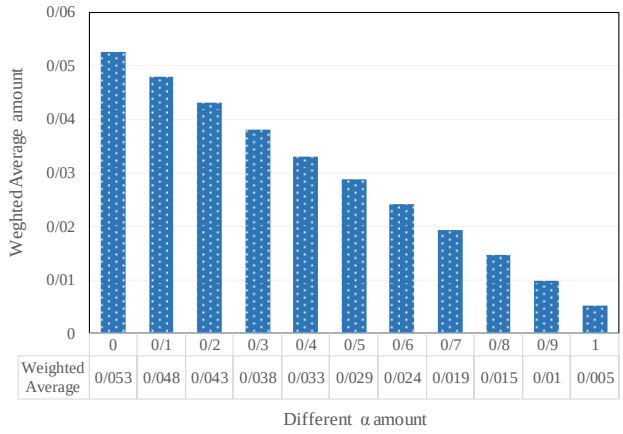


Fig. 6: The effect of α to weighted average formula

The first level of security output includes the calculation of entropy for each flow which is considered as a triple of attack, no-attack, and uncertainty; this output is kept in an array. The results are transformed according to probability

of [0, 1]. The entropy output is shown in Tables 3 and Fig. 7.

Table 3: Entropy module output

	Entropy module output		
	Attack	***No_Attack***	***Uncertainty***
Real amount	2205	7793	0
Real amount (probability)	0.22	0.77	0
Simulation output	2179	7796	23
Simulation output(probability)	0.21	0.787	0.002

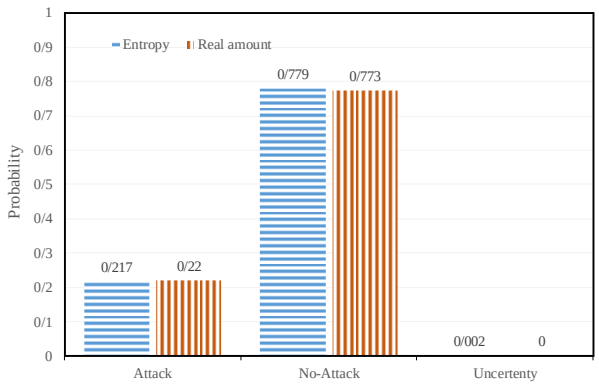


Fig.7: Entropy results

As observed, entropy has been able to detect the amount of normal flows with a high probability percentage. Also, the suspicious flows detected in this module are detected very precisely.

The traffic rate detected by entropy is shown as Fig. 8. As depicted in Fig. 8, in 7000 to 9998 interval, the traffic rate has increased. It means that the attack probability exists in this interval.

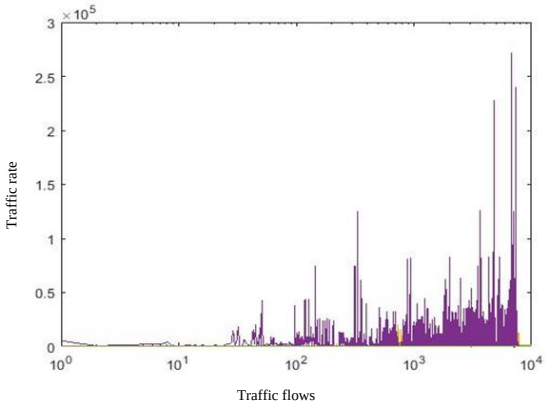


Fig. 8: The traffic rate detected by Entropy

Doing a test on the whole dataset, in the first module, 4 parameters are calculated in terms of efficiency.

- ✓ True positive: the number of the attacks detected correctly
- ✓ False negative: the number of the attacks not detected correctly
- ✓ True negative: the number of flows not attacked and not detected as attack
- ✓ False positive: the number of flows detected as attack while they were not so

Table 4 shows the amount of 4 parameters in terms of efficiency in Entropy

Table 4: Calculation of 4 parameters in terms of efficiency in Entropy

True positive	***False negative***	***True negative***	***False positive***
0.99	0.002	0.96	0

The amounts gained from these parameters are shown in Table 4 which entropy %98.82 has correctly detected the attacks; only 0.2 % of attacks were not correctly determined. Moreover, it has detected the normal flows correctly with %99.6.

The amounts gained from the entropy module output show that entropy can detect the amount of the flows produced by DDoS attacks in an acceptable way. To make sure about the flows considered suspicious by entropy, the suspicious traffic is sent from flows including attack and uncertainty to Dempster-Shafer plane for more analysis.

Moreover, we take the flows history into consideration. A history of flows including 7 rounds is kept in a memory. This history is considered for investigating each flow and, also, each sender which can save the information of the last 7 rounds relative to an event; finally, the views average in these 7 rounds is considered as probability (MT). Table 4 shows the flow history in 7 rounds in simulation test. At each round, it has been assumed that there are 5 input traffic flows and the diagnosis associated with these flows is written in the Table 5. The output amount for each flow from a certain node is considered [0 1 0] for the attack, [1 0 0] for no-attack, and [0 0 1] for uncertainty.

Table 5: The amount of flows history in 7 rounds

	Attack	**No_attack**	**Uncertainty**
Round 1	0	1	0
	1	0	0
	0	1	0
	1	0	0
	1	0	0
Round 2	0	1	0
	1	0	0
	0	1	0
	1	0	0

	1	0	0
Round 3	1	0	0
	1	0	0
	1	0	0
	0	1	0
	1	0	0
Round 4	0	1	0
	1	0	0
	1	0	0
	0	1	0
	1	0	0
Round 5	1	0	0
	1	0	0
	0	1	0
	1	0	0
	0	0	1
Round 6	0	1	0
	1	0	0
	1	0	0
	1	0	0
	0	1	0
Round 7	1	0	0
	1	0	0
	1	0	0
	0	1	0
	1	0	0

As shown in Table 5, in the first round, the first flow has been registered in the memory with the attack probability of [0 1 0]; the second flow with [0 1 0]; the third flow with [0 1 0]; the fourth flow with [1 0 0]; the fifth flow with [0 1 0]. Next rounds are shown in the same way.
When a new flow enters, according to circular shift, it is placed in the first input flow; finally, the average of all 7 rounds of views, with α=0.7, is determined as the triple of attack, no-attack, and uncertainty. Fig. 9 shows the probability values for 7 recent rounds.

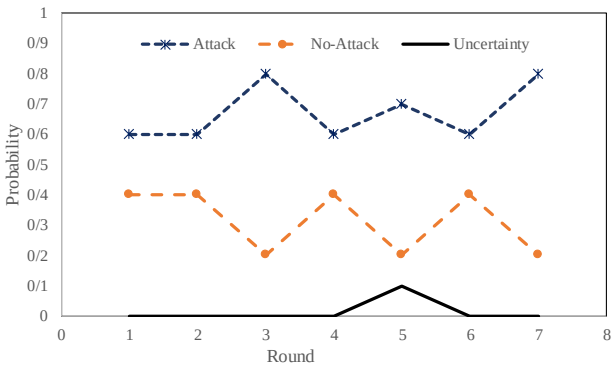


Fig. 9: The flows history in recent 7 rounds

Considering an amount for α, we try to make the numbers gained from the previous rounds affect the present responses. Therefore, we took different amounts of α into consideration; here, 0.9 and 0.7 are considered as the representative of maximum and 0.2 as minimum as shown in Fig. 10.

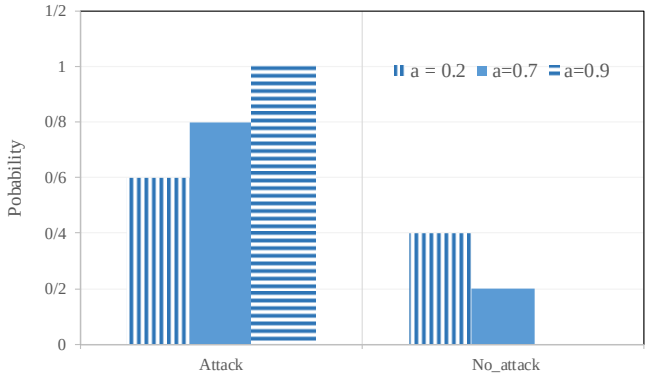


Fig 10: The effect of α amount on flows history formula

As observed, the horizontal levels of the diagram show the attack, no-attack, and uncertainty. For instance, Table 5 shows the results obtained from Dempster-Shafer based on input parameters. Considering α equal to 0.7, this amount shows more attack probability relative to 0.2, that is, the past numbers have less effect on the new responses as shown in Fig. 10. If α equals 0.9, the attack probability is 1. Therefore, the entropy module output and the flows history are sent as second module input (Dempster-Shafer) for a more exact detection and decision-making according to this theory and, finally, coming to a complete conclusion.
For instance, Table 6 shows the results gained by Dempster-Shafer based on input parameters.

Table 6: Demster_Shafer results

	Likelihood amount pl_i		**Belief amount bel_i**		**Mass functionality**	
	Entropy	History	Entropy	History	Entropy	History
Attack	0.22	0.6	0.22	0.6	0.22	0.6
No-attack	0.78	0.4	0.78	0.4	0.78	0.4
Uncertainty	0.99	1	0.99	1	0.002	0

According to the Monte-Carlo combination method, Combination m_1 &m_2
Attack probability: 0.29
No-attack probability: 0.7
Uncertainty probability: 0

The details of combining the results of Dempster-Shafer functions are depicted in Table 7. The results combination is done by Monte-Carlo.

Table 7: Combination of the results of Dempster_Shefer function

	Dempster_Shefer	*M_2 (Flow history)*	*M1 (Entropy)*
Attack	0.22	0.6	0.29
No_Attack	0.78	0.4	0.7
Uncertainty	0.002	0	0

As seen in Table 7, Dempster-Shafer, after combining M1 and M2 belief degrees, has detected the attack amount as being equal to 0.29, that is, it has attributed the DDoS attacks to the malicious flows with higher probability. As the Dempster-Shafer results show, no-attack has decreased relative to entropy results. It has increased the attack probability in the final results.
The simulation results and each function's output result are shown in Fig. 11.

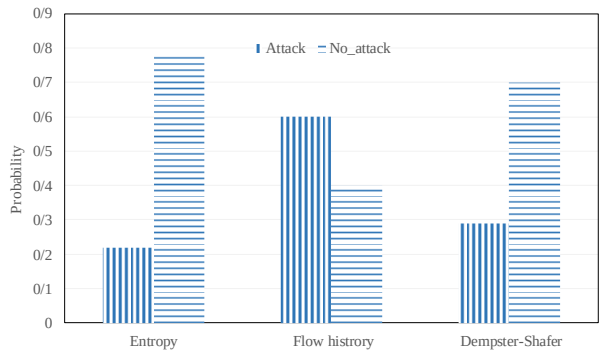


Fig. 11: Comparison of Output Entropy and Dempster_Shafer

As observed, Dempster-Shafer theory, after combining M1 and M2 belief degrees, comes to a certain conclusion about the event. In this example, the second level of security detects the suspicious flow as DDoS attack with greater certainty. According to these results, the controller blocks the detected malicious flow and, by detecting the malicious nodes, stops the malicious sent flow from entering the controller.

6- Conclusions

In SDWN, software-defined network technology is used for solving many basic problems of the wireless sensor network. DDoS attacks are serious threats to modern networks. The main purpose of the present paper is detecting DDoS attacks in their early stages. In the proposed method, to detect DDoS attacks, the central controller of the software-defined network is used, and the entropy approach, as an affective, truly light-weight, and quick solution is also made use of. Also, to confront DDoS attacks, Dempster-Shafer theory is used; it is an effective device for detecting DDoS attacks and finding the attacker. In this paper, in addition to attack detection, the malicious node is also detected ending in the prevention of the attack. By calculating the criteria based on evidence theories like entropy and Dempster-Shafer, the changes differences in the traffic behavior of such events could be detected. The present paper has accomplished the evaluation of attacks aiming at increasing the attack detection rate, maximizing the true positive, decreasing the false negative, and detecting the attack.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," Computer Networks, vol. 38, pp. 393-422, 2002/03/15/ 2002.
[2] F. Losilla, C. Vicente-Chicote, B. Álvarez, A. Iborra, and P. Sánchez, "Wireless Sensor Network Application Development: An Architecture-Centric MDE Approach," in Software Architecture, Berlin, Heidelberg, 2007, pp. 179-194.
[3] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in Software Defined Networks: A Survey," IEEE Communications Surveys & Tutorials, vol. 17, pp. 2317-2346, 2015.
[4] A. Akhunzada, E. Ahmed, A. Gani, M. K. Khan, M. Imran, and S. Guizani, "Securing software defined networks: taxonomy, requirements, and open issues," IEEE Communications Magazine, vol. 53, pp. 36-44, 2015.
[5] I. T. Haque and N. Abu-Ghazaleh, "Wireless Software Defined Networking: A Survey and Taxonomy," IEEE Communications Surveys & Tutorials, vol. 18, pp. 2713-2737, 2016.
[6] D. He, S. Chan, and M. Guizani, "Securing software defined wireless networks," IEEE Communications Magazine, vol. 54, pp. 20-25, 2016.
[7] M. Karakus and A. Durresi, "Quality of Service (QoS) in Software Defined Networking (SDN)," J. Netw. Comput. Appl., vol. 80, pp. 200-218, 2017.
[8] Z.-j. Han and W. Ren, "A Novel Wireless Sensor Networks Structure Based on the SDN," International Journal of Distributed Sensor Networks, vol. 10, p. 874047, 2014/03/01 2014.
[9] T. Kgogo, B. Isong, and A. M. Abu-Mahfouz, "Software defined wireless sensor networks security challenges," in 2017 IEEE AFRICON, 2017, pp. 1508-1513.
[10] F. Olivier, G. Carlos, and N. Florent, "SDN Based Architecture for Clustered WSN," in 2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2015, pp. 342-347.
[11] C. Ioannou, V. Vassiliou, and C. Sergiou, "An Intrusion Detection System for Wireless Sensor Networks," in 2017 24th International Conference on Telecommunications (ICT), 2017, pp. 1-5.
[12] A. D. Gante, M. Aslan, and A. Matrawy, "Smart wireless sensor network management based on software-defined networking," in 2014 27th Biennial Symposium on Communications (QBSC), 2014, pp. 71-75.
[13] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A Survey on Software-Defined Wireless Sensor Networks:

Challenges and Design Requirements," IEEE Access, vol. 5, pp. 1872-1899, 2017.
[14] S. W. Pritchard, G. P. Hancke, and A. M. Abu-Mahfouz, "Security in software-defined wireless sensor networks: Threats, challenges and potential solutions," in 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), 2017, pp. 168-173.
[15] K. S. Sahoo, M. Tiwary, and B. Sahoo, "Detection of high rate DDoS attack from flash events using information metrics in software defined networks," in 2018 10th International Conference on Communication Systems & Networks (COMSNETS), 2018, pp. 421-424.
[16] S. Shin, L. Xu, S. Hong, and G. Gu, "Enhancing Network Security through Software Defined Networking (SDN)," in 2016 25th International Conference on Computer Communication and Networks (ICCCN), 2016, pp. 1-9.
[17] J. Wu, K. Ota, M. Dong, and C. Li, "A Hierarchical Security Framework for Defending Against Sophisticated Attacks on Wireless Sensor Networks in Smart Cities," IEEE Access, vol. 4, pp. 416-424, 2016.
[18] P. Zhang, H. Wang, C. Hu, and C. Lin, "On denial of service attacks in software defined networks," IEEE Network, vol. 30, pp. 28-33, 2016.
[19] D. E. P. Alina Madalina Lonea, Huaglory Tianfield, "Detecting DDoS Attacks in Cloud Computing Environment," International Journal of Computers Communications & Control, vol. 8, 2013.
[20] Y. Ashok Khimabhai and V. Rohokale, SDN Control Plane Security in Cloud Computing Against DDoS Attack, 2016.
[21] S. K. Fayaz, Y. Tobioka, V. Sekar, and M. Bailey, "Bohatei: Flexible and elastic ddos defense," in 24th {USENIX} Security Symposium ({USENIX} Security 15), 2015, pp. 817-832.
[22] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," in 2015 International Conference on Computing, Networking and Communications (ICNC), 2015, pp. 77-81.
[23] A. Navaz, V. Sangeetha, and C. Prabhadevi, "Entropy based anomaly detection system to prevent DDoS attacks in cloud," arXiv preprint arXiv:1308.6745, 2013.
[24] R. Vadehra, M. Singh, B. Singh, and N. Chowdhary, "Evaluation of Flow and Average Entropy Based Detection Mecha-nism for DDoS Attacks using NS-2," International Journal of Security and Its Applications, vol. 10, pp. 139-146, 2016.
[25] S. Yu, W. Zhou, R. Doss, and W. Jia, "Traceback of DDoS attacks using entropy variations," IEEE Transactions on Parallel and Distributed Systems, vol. 22, pp. 412-425, 2011.
[26] B. Rashidi, C. Fung, and E. Bertino, "A collaborative DDoS defence framework using network function virtualization," IEEE Transactions on Information Forensics and Security, vol. 12, pp. 2483-2497, 2017.
[27] G. A. N. Segura, S. Skaperas, A. Chorti, L. Mamatas, and C. B. Margi, "Denial of Service Attacks Detection in Software-Defined Wireless Sensor Networks," in 2020 IEEE International Conference on Communications Workshops (ICC Workshops), 2020, pp. 1-7.
[28] A. Wani and S. Revathi, "DDoS Detection and Alleviation in IoT using SDN (SDIoT-DDoS-DA)," Journal of The Institution of Engineers (India): Series B, vol. 101, pp. 117-128, 2020/04/01 2020.
[29] G. A. Nunez Segura, A. Chorti, and C. Borges Margi, "Centralized and Distributed Intrusion Detection for Resource Constrained Wireless SDN Networks," arXiv e-prints, p. arXiv: 2103.01262, 2021.
[30] Á. MacDermott, Q. Shi, and K. Kifayat, "Distributed Attack Prevention Using Dempster-Shafer Theory of Evidence," in Intelligent Computing Methodologies, Cham, 2017, pp. 203-212.
[31] h. tan, M. Ma, H. Labiod, and P. H. J. Chong, "TEDS: A Trusted Entropy and Dempster Shafer Mechanism for Routing in Wireless Mesh Networks," presented at the MOBILITY 2014 The Fourth International Conference on Mobile Services, Resources, and Users, Paris, France, 2014.
[32] M. Ahmed, X. Huang, and D. Sharma, "Dempster-Shafer Theory to Identify Insider Attacker in Wireless Sensor Network," in Network and Parallel Computing, Berlin, Heidelberg, 2012, pp. 94-100.
[33] A. Vassilev and T. A. Hall, "The Importance of Entropy to Information Security," Computer, vol. 47, pp. 78-81, 2014.
[34] R. R. Y. Liu, Classic Works of the Dempster-Shafer Theory of Belief Functions: Springer, Berlin, Heidelberg, 2008.
[35] J. H. Ying-Jin Lu, "Dempster-Shafer Evidence Theory and Study of Some Key Problems," Journal of Electronic Science and vol. 15, pp. 106-112, 2017.

Reyhane Hoseini was born in 1993. She received the BSc degree from Payame Noor University, IRAN, in 2016 and a Master's degree from Imam Reza International University in 2018. Hers research topics include Security, trust management, wireless sensor network security, Software defined Network

Nazbanoo Farzaneh received her B.S degree in 2002 and her M.S degree in computer engineering from the Ferdowsi University, Mashhad, Iran in 2006. She received his Ph.D. degree in computer engineering from Ferdowsi University in 2014. She is currently an assistant of computer engineering department of Imam Reza International University, Mashhad, Iran. Her main research interests include Wireless Sensor Network, Next Generation Network (NGN), Vehicular Ad hoc Network, Congestion Control, Quality of Services, Fuzzy Logic Control, Reinforcement Learning, Game theory and Queuing Theory

Denoising and Enhancement Speech Signal Using Wavelet

Meriane Brahim*
University of Batna 2 (UB2), Advanced Electronics Laboratory
Higher School of Technological Education Skikda ENSET , Algeria
tlcom_brahim@yahoo.fr

Received: 28/Oct/2020 Revised: 09/Feb/2021 Accepted: 31/Mar/2021

Abstract

Speech enhancement aims to improve the quality and intelligibility of speech using various techniques and algorithms. The speech signal is always accompanied by background noise. The speech and communication processing systems must apply effective noise reduction techniques in order to extract the desired speech signal from its corrupted speech signal. In this project we study wavelet and wavelet transform, and the possibility of its employment in the processing and analysis of the speech signal in order to enhance the signal and remove noise of it. We will present different algorithms that depend on the wavelet transform and the mechanism to apply them in order to get rid of noise in the speech, and compare the results of the application of these algorithms with some traditional algorithms that are used to enhance the speech. The basic principles of the wavelike transform are presented as an alternative to the Fourier transform. Or immediate switching of the window The practical results obtained are based on processing a large database dedicated to speech bookmarks polluted with various noises in many SNRs. This article tends to be an extension of practical research to improve speech signal for hearing aid purposes. Also learn about the main frequency of letters and their uses in intelligent systems, such as voice control systems.

Keywords: Wavelet Transform; Speech Enhancement; Denoising, Discrete Wavelet Ttransforms (DWT); Noise Reduction in Speech Signals.

1- Introduction

With the advent of wavelet analysis, it became popular to address unstable physical quantities, such as speech analysis, voice signature detection, and speech recognition. Wavelets have proven successful in front-end speech recognition processors that are an alternative to instant switching using time-wave resolution. For speech recognition, or voice enhancement, wave shapes are based on Henning's window [1-3] The performance of the recognition depends on the frequency domain coverage. The goal of good speech recognition is to increase the wavelength bandwidth without significantly affecting time accuracy. This can be done by collecting the hard-to-detect white noise of the wave and removing it by conventional methods.

Speech components will have large values compared to noise because it is considered an extraneous signal. Transactions are calculated using a multi-precision wave filter bank.[4-13] The filter selection depends on the noise level and other parameters. To obtain a good noise reduction result, a good threshold level must be estimated [2-12].

The wavelet function and the level of decay also play an important and pioneering role in de-noise and noise-canceling signal quality. Recently, various wavelet-based methods have been proposed for the purpose of reducing noise in speech. The wavelength division modulus method is a noise reduction procedure to remove noise by reducing the wavelet coefficients in the wavelet field. The method depends on the threshold in which the signal is each wavelet.

2- Speech Signals

Speech is a natural and basic way for humans to convey message and thoughts. Speech frequency normally ranges between 3 Hz to 4 KHz depending upon the character. However the human beings have an audible frequency range of 20 Hz to 20 KHz. The most common problem in speech processing is the effect of meddling of noise in the speech signals. The noise masks the speech signal reduces the quality and the speech is greatly affected by presence of backdrop noise [1-11], Noise shrinking or speech enrichment algorithm is to improve the performance of communication systems when their input or output signals are corrupted by noise signal[14].

Non-stationary signal analysis methods are focused to model the inherent time-varying characteristics of the analyzed signals recorded in several areas namely, communications, speech analysis and synthesis, radar, biomedical, and mechanical engineering [1-5], The conventional methods based on the Fourier transform are not well suited for spectroscopy of these signals. Moreover, complex biological signals such as a brain signal or a speech signal [1-5-17].

3- Speech Acquisition With Cool Edit

The Cool Edit program for recording, auditory and visual analysis of sounds and their spectra

They will be recorded in monophony at the sampling frequency of 10 kHz with a converter 16 so called the sampling period.

$$T_e = \frac{1}{F_e} = \frac{1}{10 \times 10^3} = 0.1 \times 10^{-3} s \qquad (1)$$

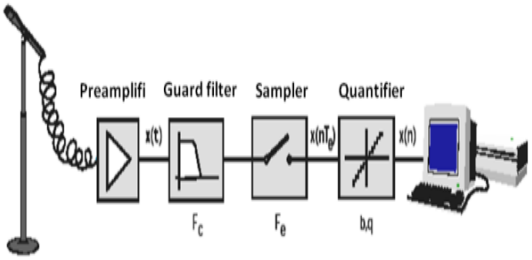


Fig.1 Block of Recording a speech signal

4- Wavelet Transform

The wavelet transform (WT) uses wavelet function and varied scales to decompose signals in the T-F domain, and it can guarantee the temporal and spectral resolutions in the entire frequency range. Since introduced in the 1970s, the WT has been used in varied applications, such as signal detection, imaging processing, de-noising of signal, speed improvement, audio classification, etc [2-8]. Zhu and Kim applied the Morlet wavelet transform to analyze the impact noise.

The wavelets have two important properties: first, the scaling factor, and secondly, the transformation and the relationship between them roughly correspond to the measurement process. Compressed waves are used. When the high-bandwidth waves span, they correspond to the low-frequency signals [14-17], at lower bands, it corresponds to rapidly changing signals that consist of high frequencies. Unlike other transmission tools (Fourier transforms, etc.) used in signal processing, waves allow analysis of signals in both frequency and time domains. There are two types of wavelet transfers: continuous and discrete wavelet transfers. Both transformations are continuous in time (analog), and with their help analog signals can be represented. [5-6-16].

4-1- General theory of CWT

In this work, we stated only some keys equations and concepts of wavelet transform, more rigorous mathematical treatment of this subject can be found in [3-6]. A continuous-time wavelet transform of f(t) is defined as:

$$CWT_{\psi} f(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t).\overset{*}{\psi}\left(\frac{t-b}{a}\right).dt \qquad (2)$$

Here $a,b \in R,\ a \neq 0$ and they are dilating and translating coefficients, respectively. This multiplication of $|a|^{-1/2}$ is for energy normalization purposes so that the transformed signal will have the same energy at every scale. The analysis function Ψ(t), the so-called mother wavelet has to satisfy that it has a zero net area, whichsuggest that the transformation kernel of the wavelet transform is a compactly support function.**[9]**

A disadvantage of CWT is that the signal representation is often redundant, because a and b are continuous over R (the real number).As the original signal can be completely reconstructed by a model copy of W_f (b, a). Usually, we try W_f (b, a) in a binary network i.e., $a = 2^{-m}$and $b = n2^{-m}m,n \in Z+$. Substituting the last one into

$$DWT_{\psi} f(a,b) = \int_{-\infty}^{+\infty} f(t).\overset{*}{\psi}(t).dt \qquad (3)$$

where *Ψm,n(t) = 2-mΨ(2mt-n)* is the dilated and translated version of the mother wavelet *Ψ(t)*.[9,14]

SNR is the power of the useful signal to the power of the noise ratio in which meaningful information characterizing the ratio of these power. SNR depended on an additive noise where the undistorted unquantized signal *s* [*i*] and an additional quantization error *e* [*i*] are superposition generated the quantized signal *sq* [*i*]. SNR is usually specified in the logarithmic measure in decibels (dB) in order to cover a wide range of possible SNR values:[10]

$$SNR_{dB} = 10\log\left(\frac{Px}{Pe}\right) = 20\log\left(\frac{Ax}{Ae}\right) \qquad (4)$$

Where *Px*, *Pe*are the average powers of the corresponding signals, and *Ax*, *Ae*is the average value of the amplitude. SNR is often called SQNR

4-2- Temporal and Spectral Resolutionsin the CWT

Resolutions in the time and frequency domains are critical for evaluation of performance of different wavelets. The temporal resolution in the time domain σ_t and the spectral resolution in the frequency domain $\sigma_w$of the CWT can be defined as[5.6]:

$$\sigma_t^2(\gamma) = \int_{-\infty}^{+\infty} (t-u\gamma)^2 \left|\phi_\gamma(t)\right|^2 dt \qquad (5)$$

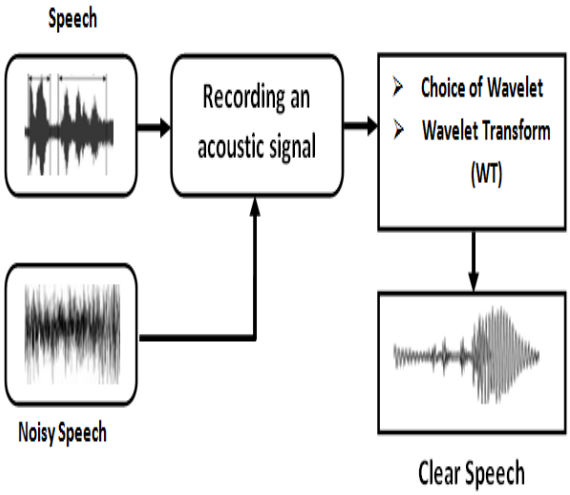
DWT is mathematical tool for decomposing data in a top down fashion. DWT represent a function in terms of a rough overall form, and a wide range of details. Despite of the requirement and type of function i.e., signals images etc. DWT offers sublime methodology and technique for representing the amount of detail present.

Wavelets perform and offer scale based analysis for a given data. A wide range of applications and usage has been found for these wavelets including signal processing, mathematics and numerical analysis and, for its better performance in signals and image processing it is considered an alternative to Fast Fourier Transform as DWT provide time frequency representation When there is a need for processing and analyzing non stationary tool, DWT can be used .Study shows that discrete wavelets transform have high performance in speech signal processing so far. [18-19].

5- Speech enhancement methods

There are various speech enhancement methods proposed for noise reduction and to improve the speech quality and clearness. Only one algorithm.
is not enough for all the types of noise present in the surrounding. Hence speech enhancement algorithms are created based on the application she block diagram of speech enhancement is show in figure (2)

Fig. 2 Speech Enhancement Method



The functional diagram of the proposed method has been shown in Fig.3

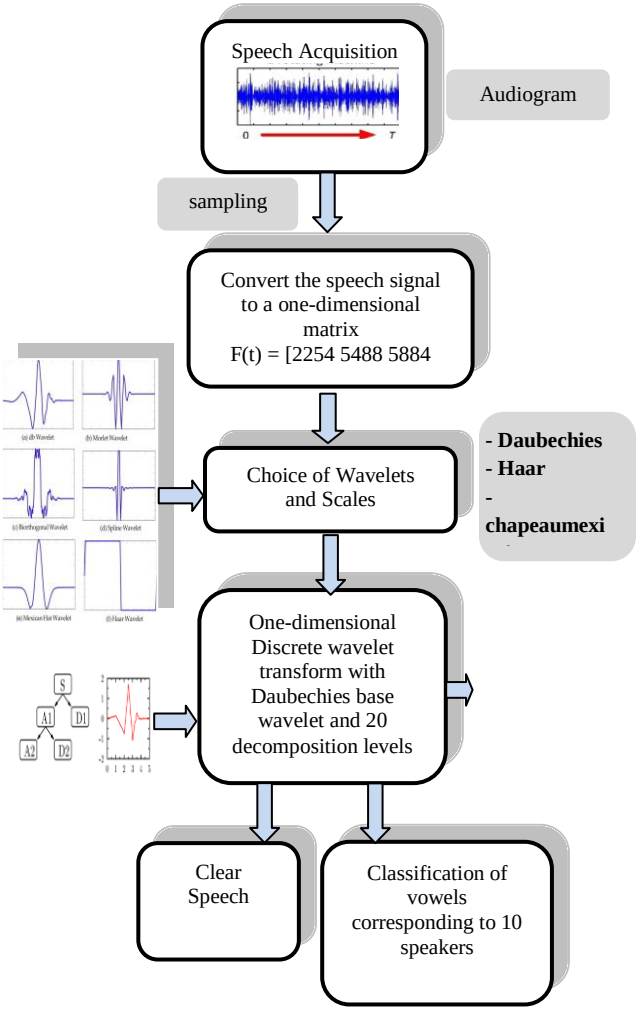


Fig. 3 Diagram of the proposed method

In this method, we rely on the processing of the audio signal stored in the database, where noise can be removed and the main frequencies of each letter can be identified.
This method can be used in real time after developing the algorithm in embedded system that manages the voice signal and identifies certain commands that control an automatic system.

5-1- Detection of Singularity of the impulse noise signal in CWT

For noise evaluation, the oscillation of the acoustic signals is regarded as a considerable important metric. The CWT is often applied to detect the singularities of a transient signal.

On every stage of numerical simulations standard procedures for calculations of wavelet coefficients were used (in the case of the continuous as well as discrete wavelet transforms), which are integral parts of MATLAB software. For these calculations we employed MATLAB cwt function.

Fig. 4 vowel A time domain

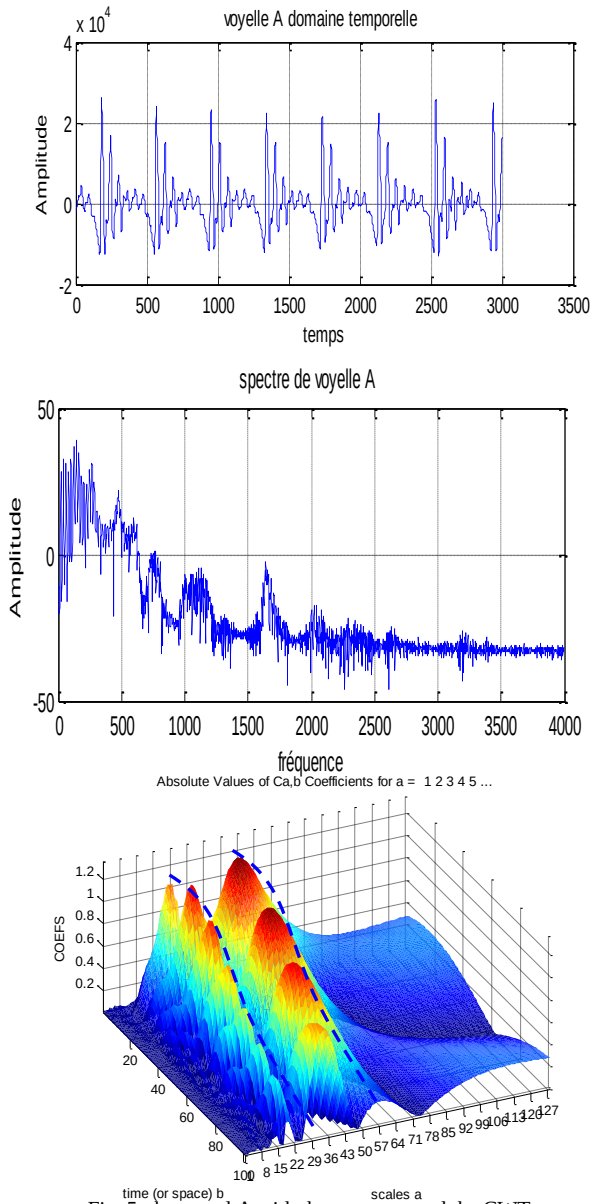


Fig. 5 the vowel A with the spectrum and the CWT

Table 1: shows the different frequencies of the vowel " A, E, I, O, U " respectively

Frequency Hz / Vowel	Frequency F1	Frequency F2	Frequency F3
A	625	1491	2356
E	387	1985	2875
I	246	2018	3196
O	313	756	2271
U	312	750	2079

We usually focus on three main frequencies for letter and word identification, after removing noise and applying wavelet transform.

In practice, the third frequency can be neglected because it may be close in several letters, and we are satisfied with the first and second frequency only, especially if the noise is removed at a satisfactory rate.

The figure opposite shows the letter E after noise cancellation

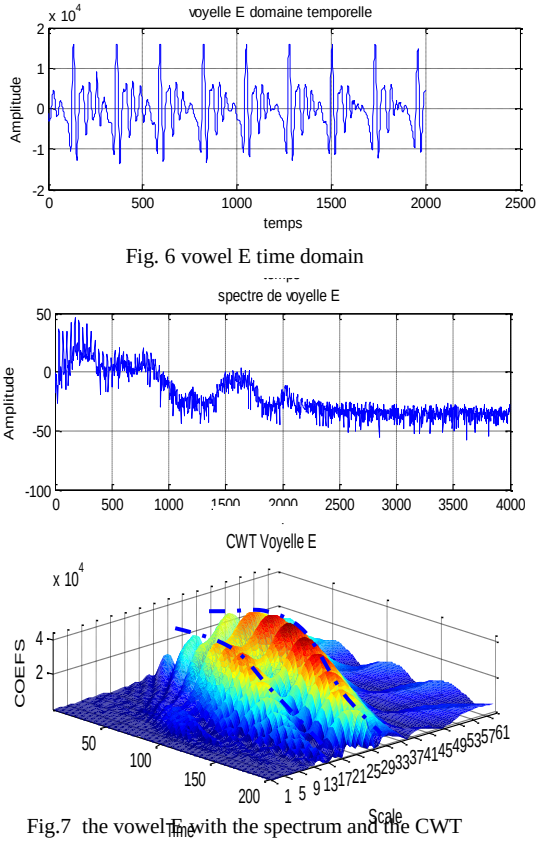


Fig. 6 vowel E time domain

Fig.7 the vowel E with the spectrum and the CWT

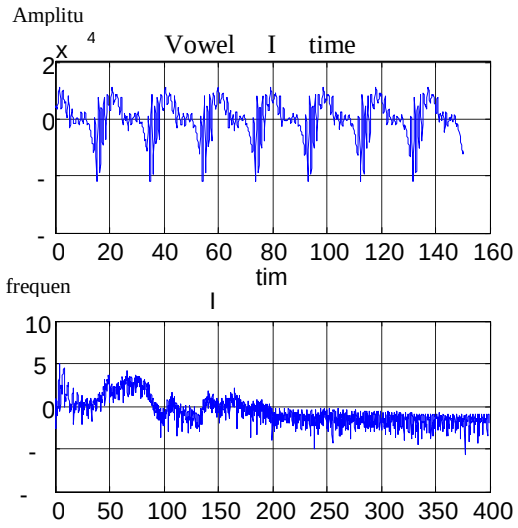


Fig.8 the vowel E with the spectrum and the CWT

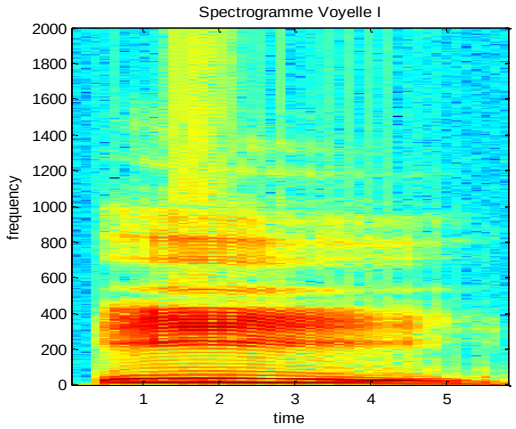


Fig.9 Spectrogram vowel i

Table 2: Presents the results obtained for various simulation cases following the order of Daubechies wavelets, applied to the vowel 'a'

Vowel a		Baub.2	Baub.4	Baub.8	Baub.16	Baub.20
Appro N°=1	A1	7268.9	7269.8	7268.4	7265.6	7264.3
Decom N°=1	D1	209.2332	146.5979	144.4224	144.1895	137.3471
Appro N°=2	A2	5139.5	5139.6	5137.7	5133.7	5131.8
Decom N°=1	D2	147.9397	103.6432	102.0855	101.8817	97.0284
Appro N°=3	A3	5139.0	5138.1	5134.2	5126.3	5122.5
Decom N°=1	D3	0.2686	0.0442	0.0555	0.0760	0.0567
Appro N°=4	A4	5138.5	5136.7	5130.8	5119.0	5113.2
Decom N°=1	D4	0.0529	0.0104	0.0174	0.0269	0.0285
Appro N°=5	A5	5138.5	5135.2	5127.3	5111.7	5104.0
Decom N°=1	D5	0.0529	0.0092	0.0068	0.0208	0.0192

5-2- Enhancement vowel "A", by the Wavelets

The following figure shows the resulting multi-resolution vowel "A", by the Wavelets

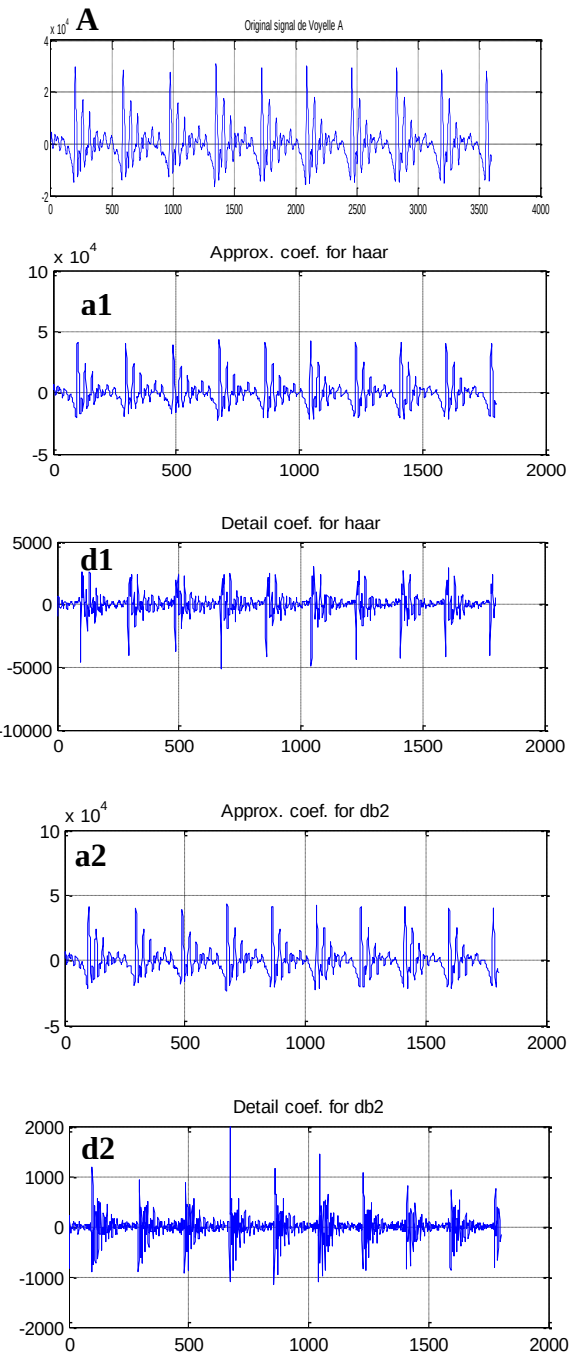


Fig.10 vowel "A", Levels of Decomposition by the Wavelets A: original speech signal, a1: appro 1 , d1: levele 1 , a2: appro 2, d2:level 2, A=a1 + d1 + a2 + d2

The method is based on thresholding in the signal that each wavelet coefficient of the signal is compared to a given threshold. Using wavelets to remove noise from a signal requires identifying which components contain the noise, and then reconstructing the signal without those components.

Unlike STFT which has constant resolution at all times and at all frequencies, WT has good temporal resolution and low frequency resolution at high frequencies, and good frequency resolution and low temporal resolution at low frequencies.

In The figure 7, the red represent the maximum spectral intensity, while the blue represents the minimum frequency.

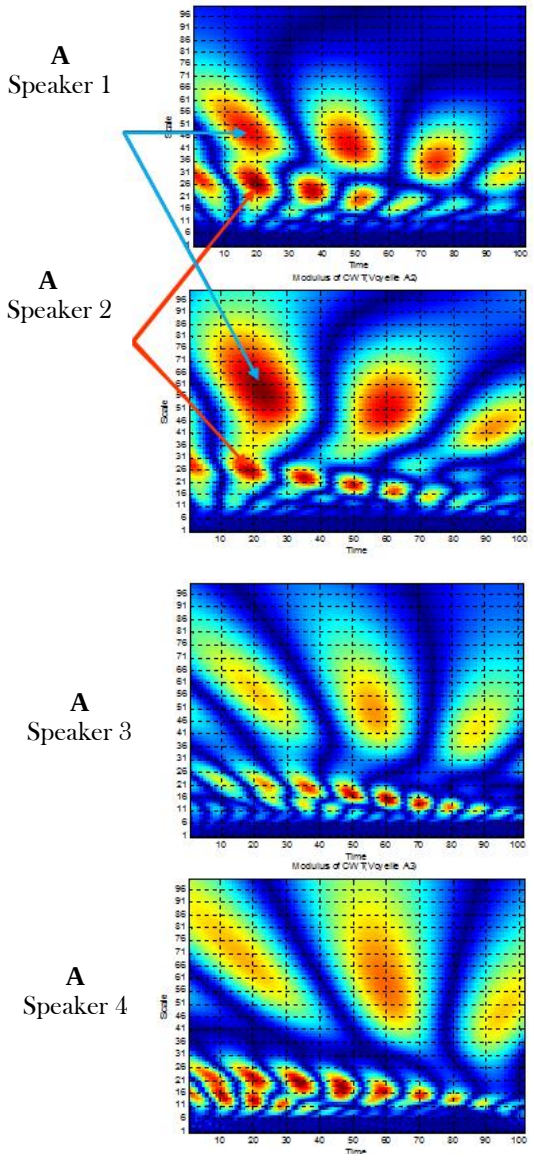


Fig.11 the CWT, Daubechies db12

Table 3 : shows the wavelet transform on the letter a after the noise is removed

	Speaker 1 /A/	Speaker 2 /A1/	Speaker 3 /A2/	Speaker 4 /A3/
The first Scale	26	25	26	24
Second Scale	51	49	50	53

Table 4 : In the same method we find the following results for the Vowel U

	Speaker 1 /U/	Speaker 2 /U1/	Speaker 3 /U2/	Speaker 4 /U3/
The first Scale	41	42	44	41
Second Scale	89	82	80	85

Does the system recognize single words or continuous speech? Obviously, it is easier to recognize isolated words well separated by periods of silence than to recognize the sequence of words constituting a sentence. Indeed, in the latter case, not only is the border between words no longer known but, moreover, the words become strongly articulated (i.e. the pronunciation of each word is affected by the word preceding as well as by the one that follows - a simple and well-known example being the links of French).

The following figure shows the Slice of a vowel A zone of the word / slap / with CWT from Daubechies.

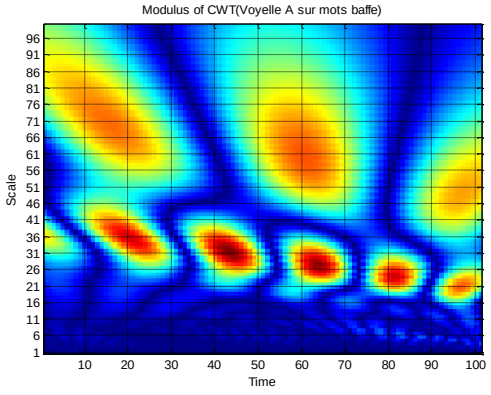


Fig.12 .vowel A the word / slap / with CWT from Daubechies

Note that the formants presented on the same scale, the following.

Table 5: shows the comparison between Vowel / A / from word slap and vowel / A / separated

	Vowel /A/ separated	Vowel /A/ from word slap
The first Scale	26	28
Second Scale	51	56

The following figure shows the Slice of a vowel / U / of the word / Une / with CWT from Daubechies

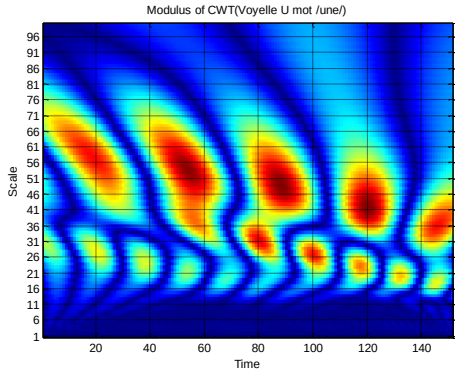


Fig.13 vowel U the word / Une / with CWT from Daubechies

Table 6 : shows the comparison between the Vowel / U / of word Une and vowel / U / separated,

	Vowel /U/ separated	Vowel /U/ from word Une
The first Scale	42	39
Second Scale	82	79

From the previous results, it can be said that it can be difficult to identify the frequencies of words or letters without removing noise.
The results also indicate that the choice of wavelet type affects the degree of filtering and noise cancellation.
The figure follows presented the word Une with CWT from Daubechies.

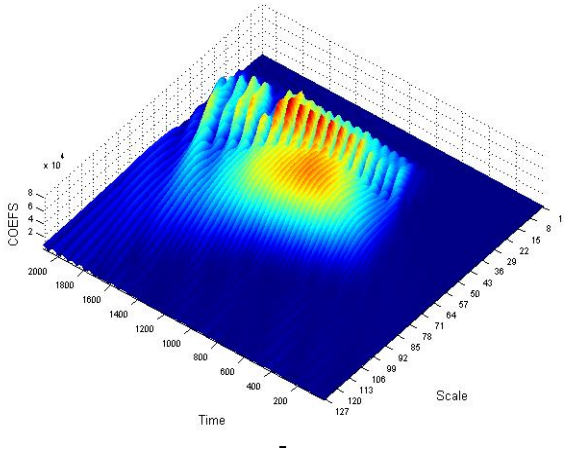


Fig.14 the word one with CWT from Daubechies

The figure (15), presented the classification of the vowels forming according to the scales corresponding with 10 speakers.

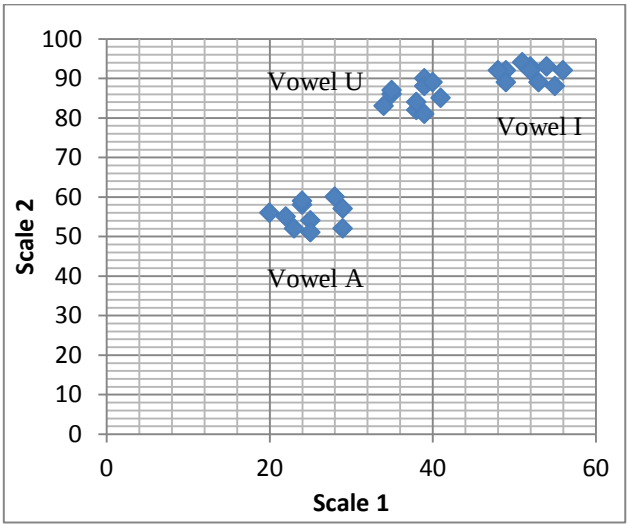


Fig.15 the classification of the vowels

After recording the speech signal for several people, and when applying the proposed method, the vowels were divided into three groups, so we can say that there is a convergence of the first and second scale in some values for each group.

6- Conclusion

The analog acquisition of the speech signal is the first thing to consider for spectral analysis of vowels or isolated words. Without it, it would be impossible to decompose this signal correctly and accurately in order to study it. This representation is not always the best for most signal processing applications. In many cases, the most relevant information is hidden in the frequency component of the Signal. The frequency SPECTRUM of a signal is constituted by the frequency components of this signal. The frequency spectrum of a signal indicates which frequencies exist in the signal.

The wavelet decomposition is similar to the Gabor decomposition: a speech signal is written in the form of a superposition of such offset and dilated wavelets.

If we get rid of the noise in the audio signal, the main frequencies are recognized in a short time and with great precision, especially in word recognition programs, and this is what we have discussed in this article where the wavelet conversion can be used to reduce noise and gain an understanding of sound.

Denoising of speech signals has been achieved successfully using wavelets. This paper provides a practical approach on how noisy audio (in wavelet form) incorporated with white gaussian noise can be denoised by using the coiflet wavelet.

References

[1] Rupali V. Mane, and Dr.M.T.Kolte, " Implementation of Adaptive Filtering Algorithm for Speech Signal on FPGA " , International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, Vol. 2, Issue3, March 2014.

[2] J. Kim, " Time-frequency characterization of hand-transmitted, impulsive vibrations using analytic wavelet transform", Journal of Sound and Vibration, vol.308, pp. 98-111, Nov 2007.

[3] M. HemaLatha and Dr.S, " Resolution enhancement of low resolution satellite images using Dual tree complex wavelet transform", International Journal of Scientific & Engineering Research, Volume 8, Issue 5, May-2017.

[4] I. Daubechies, Ten Lectures on Wavelets. Philadelphia: SIAM, 1992.

[5] U. Jayakrishnan, G. Dhavale and P. Khandelwal, Wavelet Denoising Of Discrete-Time Signals, EE678 Wavelets Application Assignment, 2005.

[6] Y. Hoshino, Wavelet Transform Analysis the Recognizing Brain Activities for Development the Palm-Size and Simplification Near-Infrared Spectroscopy Prototype System by Using Arduino,2018.

[7] K. Borna, and S. Palizdar, "Short Time Price Forecasting for Electricity Market Based on Hybrid Fuzzy Wavelet Transform and Bacteria Foraging Algorithm" Journal of Information Systems and Telecommunication, Vol. 4, No. 4, October-December 2016.

[8] S. Rani, and R. Kaur, "review: audio noise reduction using filters and discrete wavelet transformation", Journal of The International Association of Advanced Technology and Science (JIAATS), ISSN-5563 1682, Vol. 16 , June 2015.

[9] C. Gargour, M. Gabrea, V. Ramachandran, and J. Lina, "A Short Introduction to Wavelets and Their Applications", IEEE Circuits and Systems Magazine, ISSN: 1531-636X, vol. 2, pp. 57-67, 2009.

[10] R. Torkamani, and Sadeghzadeh, "Wavelet-based Bayesian Algorithm for Distributed Compressed Sensing", Journal of Information Systems and Telecommunication, Vol. 7, No. 2, April-June 2019.

[11] N. Kaladharan, "Speech Enhancement by Spectral Subtraction Method", International Journal of Computer Applications (0975 – 8887) Volume 96– No.13, June 2014.

[12] P. Goli, "Speech Intelligibility Improvement in Noisy Environments for Near-End Listening Enhancement", Journal of Information Systems and Telecommunication, Vol. 4, No. 1, January-March 2016.

[13] M. R. Kahrizi, "Long-Term Spectral Pseudo-Entropy (LTSPE): A New Robust Feature For Speech Activity Detection", Journal of Information Systems and Telecommunication, Vol. 6, No. 4, October-December 2018.

[14] K. Wang, "Wavelet packet analysis for speaker-independent emotion recognition", Neurocomputing 398 (2020) 257–264.

[15] A. Bhattacharyya, " Fourier–Bessel series expansion based empirical wavelet transform for analysis of non-stationary signals ", m5G; v1.232; Prn:7/03/2018; 12:30] P.1 (1-12).

[16] A. Upadhyay, and R.B. Pachori, "Speech enhancement based on mEMD–VMD method, Electron", ELECTRONICS LETTERS 30th March 2017 Vol. 53 No. 7 pp. 502–504.

[17] P. Singh, S.D. Joshi, R.K. Patney, and K. Saha, "The Fourier decomposition method for nonlinear and non stationary time series analysis", Proc. R. Soc. A 473 (2017).

[18] Yaseen, G. Young Son, and Soonil Kwon," Classification of Heart Sound Signal UsingMultiple Features", Appl. Sci. 2018, 8, 2344; doi:10.3390/app8122344.

[19]M, Pourseiedrezaei, "Prediction of Psychoacoustic Metrics Using Combination of Wavelet Packet Transform and an Optimized Artificial Neural Network ", Archives of Acoustics – Volume 44, Number 3, 2019.

Meriane brahim was born in Skikda, Algeria, in April 1982. He received the Dipl.-Ing. degree in electronics in 2006, the M.Sc. degree in electronics in 2009, It is currently working toward the Ph.D. degree in the faculty of Electrical from Batna University, Algeria. Currently, he is an Assistant Professor in the Department of Technology at Higher School of Technological Education Skikda ENSET , Algeria. His research interests include data compression, and Astronomical Image and speech noise cancelation and signal processing in the field of digital audio, member of the advanced electronics laboratory team, He is supervised in the doctoral thesis Pr. Benatia Djamel, Mr. Meriane was an IEEE member in 2015.

Human Activity Recognition based on Deep Belief Network Classifier and Combination of Local and Global Features

Azar Mahmoodzadeh
Department of Electrical Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran.
mahmoodzadeh@iaushiraz.ac.ir

Received: 11/Feb/2020 Revised: 18/Aug/2020 Accepted: 18/Jan/2021

Abstract
During the past decades, recognition of human activities has attracted the attention of numerous researches due to its outstanding applications including smart houses, health-care and monitoring the private and public places. Applying to the video frames, this paper proposes a hybrid method which combines the features extracted from the images using the 'scale-invariant features transform' (SIFT), 'histogram of oriented gradient' (HOG) and 'global invariant features transform' (GIST) descriptors and classifies the activities by means of the deep belief network (DBN). First, in order to avoid ineffective features, a pre-processing course is performed on any image in the dataset. Then, the mentioned descriptors extract several features from the image. Due to the problems of working with a large number of features, a small and distinguishing feature set is produced using the bag of words (BoW) technique. Finally, these reduced features are given to a deep belief network in order to recognize the human activities. Comparing the simulation results of the proposed approach with some other existing methods applied to the standard PASCAL VOC Challenge 2010 database with nine different activities demonstrates an improvement in the accuracy, precision and recall measures (reaching 96.39%, 85.77% and 86.72% respectively) for the approach of this work with respect to the other compared ones in the human activity recognition.

Keywords*:* BoW; DBN; GIST; HOG; Human Activity Recognition; SIFT.

1- Introduction

As the diversity of the applications of supervisory and security systems grows, the need for smart algorithms which are able to detect activities and behaviors of the people is intensified. Progresses in data collecting and analysis technologies have led to wide usage of the human activity recognition (HAR) systems in the daily life. Applications such as security and surveillance, crowd management, content-based image retrieval, action retrieval in images, user interface design, human-computer interaction, robot learning, sport images analysis and eHealth have raised the attention of the researchers to propose various methods for recognizing the human activities [1, 2].

Based on the design methodology for data acquisitioning process, the HAR systems are mainly categorized into visual, non-visual, and multimodal sensor technologies. One of the most popular approaches for identifying the human activities is utilizing the visual sensors (cameras) and applying the pattern recognition techniques to the images. The most important difference between the cameras and other sensors is the method of perceiving information from the environment. While most sensors generate information as one-dimensional signals, in cameras the information is received as a set of two-dimensional signals (i.e., visible images). The applications of HAR systems based on visual approach are categorized to the following groups: (1) daily life and smart houses, (2) healthcare monitoring systems, (3) surveillance and security in public environments, and (4) sports and public outdoor applications [1]. Despite the great progresses in pattern classification, the human activity recognition in static images is still considered a big challenge. In this regard, several issues such as images with different and complicated backgrounds, high volume of data, images from different views, low intra-class similarity (doing a single action in different ways by different people), low inter-class variability (e.g., the similarity between drinking and eating) and lack of temporal information have led to difficulties in the human activity recognition using static images [1]. Several publications in this regard are reported in Section 2.

In general, the human activity recognition in static images includes four steps, as follow: (1) pre-processing: applying a set of operations to the images with the aim of image enhancement and reduction of noise and redundancy; (2) feature extraction: computing and finding effective and distinctive features; (3) feature reduction: decreasing the number of features by keeping or producing the most

*Corresponding Author

discriminative ones. This step moderates the computational load; (4) classification: this is the most important step of every machine learning system which identifies the human activities [1]. For this purpose, we extract some stable and useful features of the images using a combination of GIST, HOG and SIFT descriptors. A descriptor is a representation of an image that simplifies it by extracting useful information and throwing away unimportant information. Typically, a feature descriptor converts an image to a feature vector. Although the feature vector is not useful for the purpose of viewing an image, it is appropriate for tasks like image recognition and object detection. GIST is a global feature extraction algorithm which is used to detect the scenes and provide precise prediction of the activities in scenes. HOG algorithm works based on the image gradient. This algorithm is able to accurately detect the image edges and extract the features. The SIFT algorithm extracts the features from the image which are robust against image scale changes, rotation and lightening. The two latter algorithms are classified as the local feature extractors. Using the combination of global and local approaches enhances the recognition rate for classifying the human activities, as the results of this paper demonstrates.

Since the lengths of the feature vectors of all the methods are high, the bag of word (BoW) technique is used to map the high-dimension feature space to a lower-dimension one. Recently, the application of deep learning techniques in pattern recognition problems has been widely increased. Besides several advantages of deep learning methods, they suffer from two major drawbacks: (i) overfitting and (ii), much time-consumption. One of the robust and fast deep learning techniques is the deep belief network (DBN). This network consists of the Restricted Boltzmann Machines (RBMs) for training and a back-propagation neural network for tuning the network weights. Therefore, DBN is a good classifier for the problem of the human activity recognition with several classes. In this paper, all the extracted features are concatenated and a deep belief network (DBN) is applied to 1classify and detect the activity type.

The paper is organized as follows. Section 2 reviews the related literature and research methods. In section 3, the framework of the proposed algorithm is described in details. Particularly, the pre-processing, the feature extraction and reduction and classification by means of the deep belief network is presented. In section 4, the simulation results of applying the proposed approach and those of two other well-known methods are reported and compared. Finally, the conclusions are provided in section 5.

2- Related Work

The method proposed by Wang et al. in 2006 was one basic work in the static image-based HAR field [3]. In [4], a real-time algorithm was used for the human activity identification. In the feature extraction step, the algorithms of ‘scale-invariant features transform’ (SIFT) and ‘histogram of oriented gradient’ (HOG) were used. Indeed, the human skeleton was divided into five parts and the geometric configuration of these parts are determined. Finally, the Markov hidden model and the support vector machine (SVM) were used to classify the activities. In [5], HAR was investigated using the ‘global invariant features transform’ (GIST) algorithm. In that paper, the geometrical relations between the human body parts were used for the recognition. Finally, they addressed the images classification using the SVM algorithm. In [6], the human activities were identified using the convolutional neural network. The proposed method was appropriate for streaming video images, since the extracted features were taken from the images based on individual motions.

In [7] two algorithms of ‘speeded up robust features’ (SURF) and HOG were used to extract the images features. Then the authors made use of the SVM for the classification. That paper considered only five human activities. In [8], the recognition and classification of the activities using several body-worn sensory methods were proposed. In that work, the recognition system operates based on which sensor is activated. In 2008, Ikizler et al. [9] addressed the HAR in static images by presenting a rectangular area with oriented spatial histogram. They used linear discriminant analysis and a binary SVM to categorize the activities. In that paper, the human state was extracted using the SIFT algorithm and a SVM classifier. Li and Fei [10], classified the activities in the static images by combining the scenes and objects. They realized that combining the high-level signs may improve the recognition accuracy. The results of Thurau and Hlavac [11] showed that by combining the characteristics of objects and people states, the recognition rate increases. In 2011, Li et al. [12] studied the activities and behaviors in static images in the web. In [13], the human state was estimated using the HOG algorithm and the image scene model and features were obtained using the SIFT algorithm based on the bag of features method. Delaitre et al., [14] studied the activities recognition of the new dataset using bag of features method and combined them with SVM in static images.

Zheng et al., [15] addressed the human action recognition by extracting the features using the gradient-oriented histogram descriptor and two classifiers (one based on the Poselet and the other one based on the content-based learning). Sner et al., [16] proposed a multi signs-based method to recognize human activities in static images. Sharma et al., [17] proposed an expanded part model

which is a strong distinctive descriptor of human detection. In [18] a poselet-based method was presented where poselet activation vectors obtain the pose of a person. In [19] proposed a method which learns a set of sparse features and part bases for HAR in still images. A human-centric method that identifies the location of humans and objects associated with an action is proposed in [20]. Khan et al. [21] evaluated the color descriptors and color-shape fusion methods for HAR. Moreover, in [22] they proposed some pose-normalized semantic pyramids employing body part detectors. In [23], the authors encoded multi-scale information during the image encoding stage.

3- Proposed Approach

The block diagram of the proposed approach for the human activity recognition based on the static images is shown in Fig. 1. First, a set of pre-processing operations are carried out to improve the quality of the image and prepare them for the next steps. Then some features are extracted from the image using three local and global descriptors. Following this, the bag of words technique is applied to decrease the dimension of each feature vector. In the next step, the reduced feature vectors are concatenated to form a single vector. Finally, these features are given to the classifier to predict the human activity type. In order to make comparison, the SVM and the artificial neural network (ANN) are also considered in the classification step besides the DBN.

3-1- Pre-processing

In the first step, sizes of all training and testing images are equalized to 64×64. Also, since none of the feature extraction algorithms use the color features, the color images are converted to grey ones. This conversion leads to a reduction in the number of computation operations for each image. Then, the images are passed through a low-pass filter to remove the noise. This task improves the feature extraction algorithms of the next step by avoiding the production of artifacts and wrong keypoints. Finally, edge sharpening filter is applied on the image to enhance the contrast of the edges. The enhanced image is sent to the next step for extracting features.

3-2- Feature Extraction

In this step, the features required for the classification step are computed using the HOG, GIST, and SIFT algorithms. In the following, a brief description of each algorithm is addressed.

HOG descriptor: Histogram of oriented gradient (HOG) is a local feature descriptor which is used in this paper to extract useful features from images containing the human activities. The HOG descriptor uses the distribution of directions of gradients as features. The reason for applying this descriptor is that local information of the image components can be represented by the intensity gradients or the path of the edges. This algorithm computes the gradients in local regions of an image. The general representation for computing the HOG descriptor is shown in Fig. 2. First, the image is divided into a grid of 8×8 cells. Then using the Sobel operator, the gradient magnitude and direction are calculated for every pixel in each cell. Following this, the histogram of the gradient orientations in each cell is computed with 8 bins. In fact, bin of histogram is defined based on the quantized directions and the votes (the values that go into the bins) are selected based on the magnitudes. The votes in each bin are added up to produce the 8-bin histogram for every cell. Since the gradients of an image are sensitive to overall lighting, they are normalized to become robust against lighting variations. For this purpose, a 16-cell window is used to form big-size 2×2 blocks. Notice that sliding this window by one cell constructs the neighbor block with two overlapping cells with the current block. Thus, each block has 4 histograms with 8 bins which can be concatenated to form a vector of length 4*8=32 and then it is normalized using the L2 norm. The final feature vector of HOG descriptor for the entire image is produced by concatenating all the 32×1 vectors into one giant vector. The size of this vector is (8-1)*(8-1) *32=1568 [24]. HOG has two important advantages in the application of human activity recognition. The first advantage is robustness against the lightning variations thanks to computing the gradient directions from the difference of the local intensities. The second advantage is robustness against deformation which is due to the shift and partial affine deformation. This property leads to ignorable changes in the histogram values.

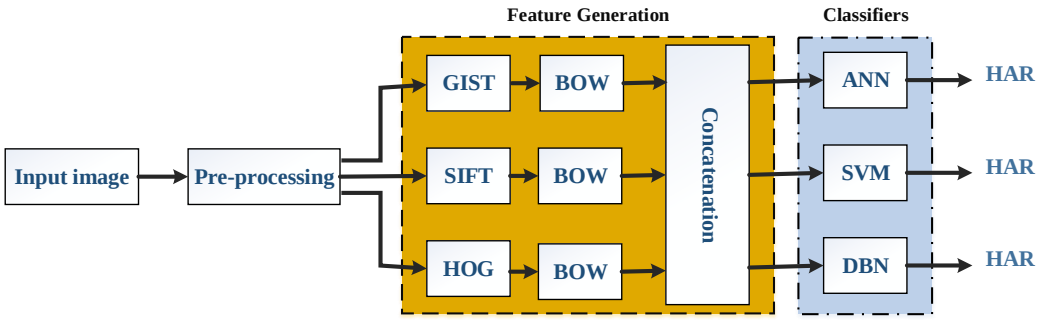


Fig. 1 The block diagram of the proposed algorithm for human activity recognition.

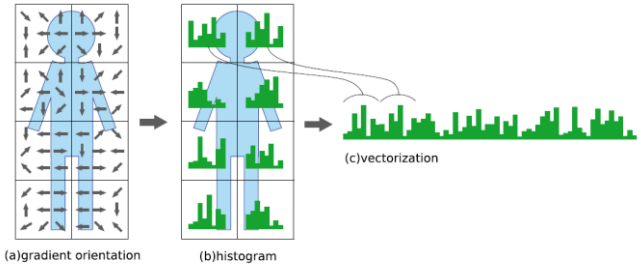


Fig .2 The general diagram of computing the HOG feature vector [24].

GIST descriptor: Global invariant features transform (GIST) descriptor, proposed by Oliva and Torralba in 2001 [25, 26], provides a small-size representation that contains sufficient information for recognizing the human activity in an image. This global descriptor represents the dominant spatial structure of an activity by analyzing the spatial frequency and orientation. The GIST algorithm was developed based on a phenomenon called spatial envelope which enables the algorithm to precisely predict the image scenes. In fact, the GIST is produced by combining the outputs of a number of Gabor filters at different scales and directions. Gabor filter is a linear filter which is a strong tool for analyzing the texture due to its multi-resolution property in the frequency and special domains. Given an image, the GIST descriptor is computed by applying 32 Gabor filters at four scales and eight directions to the image. Accordingly, 4*8=32 feature maps with the resolution same as the original image are constructed. Then, using an 8×8 grid, each feature map is divided into 64 cells and the average of values within each block is computed. Following this, for each feature map, a vector of length 64 containing this gradient information (averages) is generated. By concatenating the feature vectors of all feature maps, the GIST vector with length of 32*64=2048 is achieved [5].

SIFT descriptor: Scale-invariant feature transform (SIFT), introduced by Lowe in 2004 [27], is one the feature-based adaptive algorithms for pattern recognition in images. Extracting the features in the SIFT algorithm is done by an identifier. The feature extraction phase includes three steps: (*i*) extracting the scale space extremums, (*ii*) improving the location accuracy and removing unstable extremums, and (*iii*) allocating a direction to each feature. In the first step, to extract the scale space extremums, the stable features in different scales are extracted using a *scale space*. The scale space presents the image structures in different scales. This space is composed of a set of Gaussian images and difference of Gaussian (DoG) images in different scales which are sorted in different layers called *Octaves*. The scale space Gaussian images for the image $I(x,y)$ using the Gaussian kernel function $G(x,y,\sigma) = \exp(-(x^2+y^2)/2\sigma^2)/2\pi\sigma^2$ is calculated as follows [28]:

$$L(x,y,\sigma) = G(x,y,\sigma) \otimes I(x,y) \quad (1)$$

Where σ indicates the scale of each image and its initial value is $\sigma_0 = 1/6$. This scale value is increased using a constant multiplier parameter k in different Octave levels. DoG images are computed as follows using the difference between two adjacent Gaussian images [28]:

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma) \quad (2)$$

The smaller scale is considered as the scale of the DoG image. After generating each Octave, the Gaussian image is scaled down to half using the resampling approach. Then, the resulted image is considered as the initial image of the next Octave; and this process is repeated. The aim of constructing the scale space is extracting features which are independent of the scale. Therefore, to extract the stable situations in DoG images, the intensity of each pixel in every Octave is compared with its eight neighboring pixels in a 3×3 window and nine pixels in neighboring upper and lower DOG images. If it is an extremum (maximum and minimum) comparing to those 26 pixels, it is stored as a candidate feature. Then, for each extracted feature based on the scale of the DOG image, the scale parameter is chosen. In the second step, to eliminate the unstable extremums, features with low contrast and those on the edges are removed. Also by interpolating the adjacent points, the exact location of each keypoint is determined. In the last step of feature extraction, a direction is dedicated to each stable keypoint [28].

Once the keypoints are extracted, the next phase is to generate the features descriptor. To make them robust against the scale and rotation variations, descriptors are made according to the scale and direction of each feature. In addition, the descriptor is designed so that it is robust against lightning variations and those caused by imaging viewpoint. To do this in the standard SIFT algorithm, first a square block around every feature in the related Gaussian image is considered. The dimensions of this block are selected according to the scale of the feature so that every bin is a square with a side equal to three times of the scale. Then the coordination of the block is rotated to get aligned the main direction of the feature. Following this, the values and directions of the gradients of the pixels within the rotated region are calculated and the gradients directions are also rotated with the main direction of that feature. Then, a Gaussian function with the scale equal to the half width of the block is used to weighting the values of the gradients. Next for each bin in the block, a histogram of weighed gradient directions of the pixels within the bin is constructed. To prevent the effects of boundaries between the bins, a tri-liner interpolation is performed to distribute the gradient values in the histogram. Finally, a SIFT descriptor is produced as a vector with 128 components. In this descriptor, the amplitudes of the components are normalized in order to reduce the lightening variation effects. After this step, a threshold value of 0.2 is considered for the values of the descriptor to reduce the effects of angle variations of the imaging. Then, the normalizing process of the descriptor is repeated.

3-3- Feature Reduction and Final Concatenation

After applying each feature extraction algorithm, a long-length feature vector is produced for the image. Finding the useful features is considered an important topic in the human activity recognition; since with smaller number of features, the computational load is decreased. To reduce the dimension of the feature space, in this paper the BoW technique is applied to the features extracted by the descriptors. The recent studies showed that the BoW method presents a set of discriminative and robust features comparing other methods which use the texture or intensity [30, 31].

To generate a BoW model, all features of different images in different classes are collected in a set. These features are clustered using the *k*-Mean algorithm. The centers of the clusters represent the code-words and their union produces a code-book. For every input image, each feature vector is dedicated to one of the centers of the clusters using the nearest neighbor method. Then, a histogram is made for the image wherein the horizontal axis is the centers of the clusters and the vertical axis is the number of the features which are dedicated to each of the centers. Finally, this histogram is considered as the *new feature vector* generated for that feature extraction algorithm [30, 31]. Once the new feature vectors of the three descriptors are found, they are concatenated to generate the final feature vector, to be given to the classifier.

3-4- Deep Belief Network

In this paper the deep belief network (DBN), as one of the most important deep learning models, is used to model the human states. This network is a fast learning algorithm which can find the optimal responses with high speed. The learning model is composed of two steps: (*i*) pre-training and (*ii*) fine-tuning; see Fig. 3. The pre-training step is done using the restricted Boltzmann machine (RBM) which is a generative model. The RBM is a type of the Boltzmann machine in which the connections between the visible and hidden units are disconnected. The unsupervised pre-training system works effectively in solving classification problems with numerous data and high diversity [32]. In the second step, the network weights are precisely tuned using a supervised algorithm. For this step, a back-propagation neural network is used. A DBN can be trained by repeatedly maximizing the conditional probability of input vectors or observable vectors. By doing this, the hidden vectors and a specific set of layer weights are obtained. As the RBM is considered the heart of the deep belief network, this machine is briefly introduced.

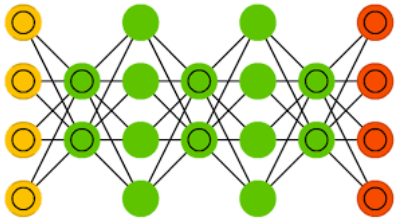


Fig. 3 Representation of the graph of a DBN.

Boltzmann machine is a special type of the Markov random field (MRF), which is represented by a symmetrical network with binary random units. This model has a set of D visible units $\boldsymbol{v} = \{0,1\}^D$ and a set of F hidden units $\boldsymbol{h} = \{0,1\}^F$. The common energy of these units in the Boltzmann machine is defined as follows [31]:

$$E(\boldsymbol{v},\boldsymbol{h}) = -\boldsymbol{b}^T\boldsymbol{v} - \boldsymbol{a}^T\boldsymbol{h} - \boldsymbol{v}^T W\boldsymbol{h} \quad (3)$$

Where the components a_i and b_i - which from the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ - are the bias terms for the hidden and visible units, respectively. These vectors are updated in each iteration through some recursive formulas. The parameters of these formulas are the main parameters of the RBM in a DBN. Moreover, each weighting element W_{ij} in the matrix W indicates the symmetrical transactional term between the visible unit i and the hidden unit j. These weights are the parameters of the

model. This network devotes a probability value to every possible pair of hidden and visible vectors in the energy function. The resulted probability distribution is defined as follows [31]:

$$P(\boldsymbol{v},\boldsymbol{h})=\frac{1}{Z}\exp\left(-E(\boldsymbol{v},\boldsymbol{h})\right),\ Z=\sum_{\boldsymbol{v}}\sum_{\boldsymbol{h}}E(\boldsymbol{v},\boldsymbol{h}) \quad (4)$$

The value of Z is recognized as a normalizing constant. The probability which a network dedicates to a training data can be increased by tuning the weights and bias in order to reach the lower energy. By defining $g(x)$ as a logistic sigmoid function, the conditional probability of the visible vector $\boldsymbol{v}$ and the hidden vector $\boldsymbol{h}$ can be obtained as follows [31].

$$P(v_i=1|\boldsymbol{h})=g\left(\sum_{j=1}^{F}W_{ij}h_j+b_i\right) \quad (5)$$

$$P(h_j=1|\boldsymbol{v})=g\left(\sum_{i=1}^{D}W_{ij}v_i+a_j\right) \quad (6)$$

$$g(x)=\frac{1}{1+\exp(-x)} \quad (7)$$

When the modes of the hidden units are selected, the input data can be reproduced by putting every v_i equal to '1' according to (6). Then the modes of the hidden units are updated so that they show the reproduced features. To find the optimal weights in the matrix W, the contrastive divergence (CD) learning method [32] is applied. The variations of the weights are defined according to the following relation [32].

$$\Delta w_{ij}=\varepsilon\left(v_i h_{j\ \text{data}}-v_i h_{j\ \text{reconstruction}}\right) \quad (8)$$

Where ε is the learning rate. The strong point of the restricted Boltzmann machines is learning with the aim of reconstruction. During the reconstruction process, this machine only uses the information of the hidden units as the feature of the learned input. If the model is able to retrieve the main input well, this means that the weights and bias are trained correctly. Because of the advantages of the RBM, in the recent years this model has been widely used in constructing DBNs. Also numerous papers are presented with the aim of improving this model and increasing its efficiency.

A RBM with a simple hidden layer is not adequate to find the features of a data. Greedy layer-wise training methodology is an efficient tool to improve the system accuracy. In this method, the first machine mapped the input data from the zero-layer to the first-layer. After training the machine, the trained features (output of the first-machine) are used as the inputs for the second RBM. The features of the last machine are considered as the learned features of the whole training process. This type of the layered learning system can be used to construct the DBNs [34, 35]. Then, the network is able to discover the deep features of the data. In fact, this network learns the deep features of the input by a pre-training process in a hierarchical process.

Logistic regression (LR) layer is added to the end of the learning system as the second stage of the DBN. This classifier is used to tune the previously trained network so that the classification is performed using the learned features. The accurate tuning process is implemented using a back-propagation algorithm. The target of this algorithm is to find the minimum in the peripheral area of parameters which have already been determined by the DBN [35, 36].

4- Simulation Results

To evaluate the proposed algorithm, the PASCAL VOC Challenge 2010 database was used which includes 225 images and nine human activities [37]. The selected images have different sizes and they are color or gray-scale. For every activity, eighteen images are randomly selected for the training phase (about 70% of all available images) and the seven remaining ones are kept for the testing phase. Therefore, the set of the training and testing images include 162 and 63 images, respectively. Fig. 4 shows some of the sample images.

The proposed algorithm is run under the MATLAB R2014a programming environment on a PC equipped with 3.2 GHZ CPU and 8 GB RAM memory. To evaluate the performance of the proposed algorithm, three measures of precision (Pre.), recall (Rec.) and accuracy (Acc.) were used which are formulated in (9)-(11), respectively:

$$\text{Pre.}=\frac{TP}{TP+FP} \quad (9)$$

Fig. 4 Some sample images of the dataset.

$$\text{Rec.}=\frac{TP}{TP+FN} \quad (10)$$

$$Acc.=\frac{TP+TN}{TP+FN+TN+FP} \quad (11)$$

Consider a two-class problem, called 'P' and 'N'. Then, TP is the number of truly detecting the class P; also FP is the number of falsely detecting the class N as P. Similarly, TN is the number of truly detecting the class N; also FN is the number of falsely detecting the class P as N. Table 1 shows the recall, precision and accuracy of the proposed algorithm for combining different features for nine human activities. In the first and second cases, the

features extracted by SIFT only and HOG only were used. In the third case, the features extracted by combination of SIFT and HOG algorithms were used and in the fourth case the features extracted by all three algorithms of SIFT, HOG and GIST were applied. Notice that before concatenating the feature vectors, the BoW algorithm is applied to each descriptor using the *k*-Mean algorithm with k=20. Moreover, the parameters of the DBN are the learning rate ε=0.5, number of hidden layers=1, number of the hidden units n=10, momentum ϕ=0.006 and the weight decay λ=0.4.

According to the results obtained, the first case had the lowest precision compared to the others. Also, the performance of the HOG features was better than that of the SIFT. By applying these two feature extraction methods in the third case, the precision values in the activities such as 'Playing instrument', 'Riding horse' and 'Using computer' were higher than the other cases. In the fourth case, the precisions of the HAR system for activities such as 'Phoning', 'Reading', 'walking' and 'Taking photo' were increased compared with the third case. The average of the total precision in nine activities for the fourth case compared with the second and the third cases increased 6% and 2.5%, respectively. Because of the inherit complexity and the nonlinear behavior of the images and the features extraction methods, increasing the number of the features or combining the diverse features not only necessarily improve the system performance but also, in some cases, increase the redundancy. Therefore, in some cases of the Table, the non-homogeneous behaviors (improvement and reduction of the efficiency index) were reported.

In addition to the precision, the accuracy of the proposed algorithm was investigated which indicated the correct detection of the algorithm. While the precision indicates the proximity of the repeated measurements to each other, accuracy is the proximity of a measurement to the actual value. The latter measure indicates that in the worst case, a measuring set to what extent is near to the real value. A correct system is not necessarily precise and vice versa. A system has appropriate performance if both the precision and accuracy are simultaneously high. From Table 1 it is inferred that the proposed algorithm generally has higher accuracy in case four than the other cases. Additionally, adding HOG to the SIFT improved the efficiency of the SIFT feature solely. Generally, it can be seen that all four cases have high accuracies. Nevertheless, for 'Phoning' the accuracy in the third case (HOG & SIFT) was lower than that of the second case (HOG alone).

By comparing the recall measure in Table 1 for the fourth case, it can be seen that for some activities such as 'running', the value of this measure is high and for some other activities such as 'Reading' and 'Taking photo' this value is low. Furthermore, using the HOG feature in the 'Riding horse' activity leads to a higher recall value compared to the SIFT and 'SIFT & HOG'. Therefore, the recall value depends on the activity type. Generally, given the results shown in Table 1 it can be seen that although the efficiency of the SIFT alone is lower than the other features in terms of the recall index, using this feature along with the HOG and GIST can improve the efficiency. The average recall value when all the features are used is enhanced 21%, 5.5% and 2% compared with (*i*) SIFT alone, (*ii*) HOG alone and (*iii*) HOG and SIFT, respectively.

To evaluate the proposed algorithm, the performance results of this method is compared with the results obtained from two well-known methods, i.e., artificial neural network (ANN) and multi-class support vector machine (SVM). The ANN is a multi-layer perceptron (MLP) with 20 neurons in one hidden layer. Also for the optimized SVM, the penalty parameter (C) is 3.5384 and the kernel parameter (σ) is 0.5147. The average accuracy results and the training and testing times for each of the four feature extraction techniques given to the SVM, ANN and DBN classifiers are reported in Table 2. Comparing according to the accuracy measure, Table 2 validates that the proposed method outperforms the SVM and ANN, in all the feature extraction methods. Nevertheless, using deep learning method in DBN leads to higher computational complexities and larger run times. Also divided results for each activity are shown in Fig. 5. According to this figure, the accuracy of the HAR based on the ANN and the SVM has reached the 74.42% and 91.80%, respectively. Meanwhile, the proposed method (based on the DBN) has achieved the accuracy of 96.39% which is higher than two other methods. It is worth nothing that all three above-mentioned classifiers were applied to the combined features set (SIFT& HOG & GIST).

Table 3 compares the proposed approach with some state-of-the-art methods for human action recognition. The approach of [21] attains a precision of 62.4%, while that of [22] achieves 63.5%. The method of [19] based on learning a sparse basis of features and parts obtains a precision of 65.1%. The approach of this work yields consistent improvement over the state-of-the-art methods with a precision of 85.8%.

5- Conclusion

The main goal of this paper is to develop a robust human activity recognizer based on the images' data. Using images for the application of HAR is feasible thanks to high-quality yet not-much expensive cameras. Easy installation and standard communication protocols make these cameras suitable for a variety of daily life uses. Thus, a novel approach was proposed in this work for the HAR application. First some pre-processing operations

were performed on the images in order to enhance their quality and make them prepared for the next steps. Then, multiple robust features including SIFT, HOG and GIST were extracted followed by the BoW technique for feature reduction. Finally, the robust features were entered to a DBN for activity training and recognition on query images. The proposed method was compared with traditional multiclass SVM and ANN approaches where it showed the superiority of our technique. The HAR system is evaluated for nine different physical activities where it achieved a mean recognition rate of 96.39%. On the contrary, the SVM and ANN approaches obtained mean recognition rates of 91.80% and 74.42%, respectively. For the next work, we plan to use other robust features and learning approaches to achieve more efficient activity recognition systems for real-time applications in complex environments.

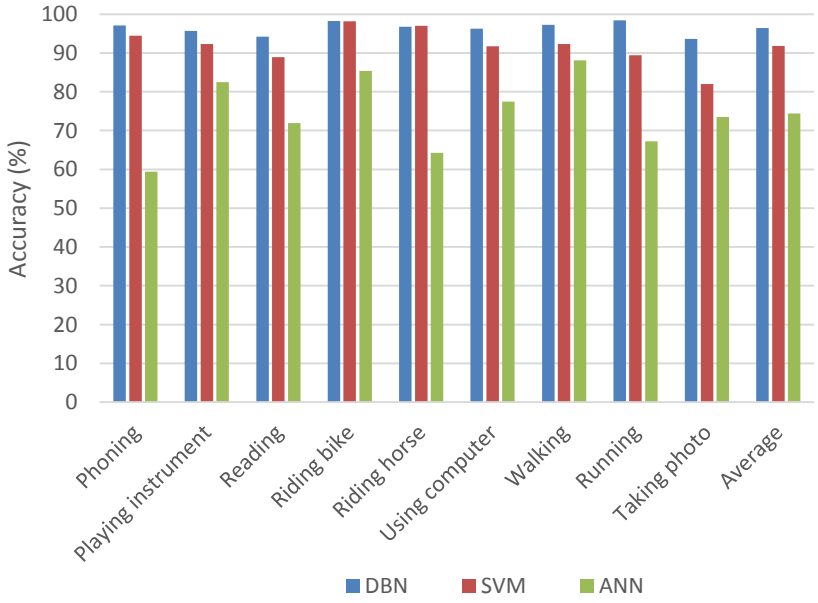


Fig. 5 The performance results of DBN, SVM and ANN methods in terms of the recognition accuracy.

Table 1: Comparing the performance of different feature extraction methods for nine human activities. The following abbreviations are used: 'Pre.': Precision, 'Rec.': Recall, 'Acc.': Accuracy, 'Ave.': Average.

	SIFT			**HOG**			**SIFT & HOG**			**SIFT & HOG & GIST**		
Activity	**Pre.**	**Rec.**	**Acc.**	**Pre.**	**Rec.**	**Acc.**	**Pre.**	**Rec.**	**Acc.**	**Pre.**	**Rec.**	**Acc.**
Phoning	52.14	80.93	92.88	73.85	92.85	96.22	72.00	93.49	96.09	80.28	95.03	97.11
Playing instrument	79.85	62.43	91.69	76.71	72.23	93.79	86.71	78.21	95.50	82.42	81.21	95.67
Reading	65.42	53.79	89.36	69.71	65.90	92.23	74.57	68.24	93.00	88.00	71.37	94.23
Riding bike	75.71	94.24	96.50	89.57	85.80	96.50	93.00	93.36	98.15	92.28	94.56	98.22
Riding horse	68.85	55.20	90.07	89.71	86.33	96.85	91.57	89.04	97.42	89.71	85.63	96.77
Using computer	58.85	59.22	90.44	71.71	89.76	95.60	75.85	92.81	96.30	70.85	97.34	96.27
Walking	56.14	57.11	90.19	83.28	85.22	96.14	86.42	88.30	96.87	92.00	87.33	97.22
Running	57.71	73.12	92.38	91.14	97.88	98.50	90.00	97.73	98.39	87.85	99.98	98.43
Taking photo	56.14	52.91	89.30	79.14	59.07	91.20	84.28	61.18	91.88	88.57	68.02	93.66
Ave.	**63.42**	**65.47**	**91.42**	**80.53**	**81.59**	**95.23**	**83.28**	**84.70**	**95.96**	**85.77**	**86.72**	**96.39**

Table 2: Comparing the average accuracy results and training and testing times for four feature extraction techniques given to the SVM, ANN and DBN.

Classifier	Measure	SIFT	HOG	SIFT & HOG	SIFT & HOG & GIST
SVM	Ave. ACC.	88.73	90.54	90.88	92.23
	Training Time (s)	0.287	0.523	0.774	1.188
	Testing Time (s)	0.035	0.071	0.101	0.151
ANN	Ave. ACC.	72.94	75.26	75.59	76.38
	Training Time (s)	0.136	0.172	0.592	0.981
	Testing Time (s)	0.029	0.047	0.089	0.134
DBN	Ave. ACC.	91.42	95.23	95.96	96.39
	Training Time (s)	2.835	3.418	6.172	8.529
	Testing Time (s)	0.081	0.094	1.053	1.105

Table 3: Comparison with the state-of-the-art results according to the precision measure. 'PrM.' stands for the proposed method.

	Phoning	*Playing instrument*	*Reading*	*Riding bike*	*Riding horse*	*Using computer*	*Walking*	*Running*	*Taking photo*	***Ave.***
Poselet[18]	49.6	43.2	27.7	83.7	89.4	59.1	67.9	85.6	31.0	*59.7*
IaC [13]	45.5	54.5	31.7	75.2	88.1	64.1	62.0	76.9	32.9	*59.0*
POI [14]	48.6	53.1	28.6	80.1	90.7	56.1	69.6	85.8	33.5	*60.7*
LAP [19]	42.8	60.8	41.5	80.2	90.6	66.1	74.4	87.8	41.4	*65.1*
WPOI [20]	55.0	81.0	69.0	71.0	90.0	50.0	44.0	59.0	36.0	*62.0*
CF [21]	52.1	52.0	34.1	81.5	90.3	59.9	66.5	88.1	37.3	*62.4*
SM-SP[22]	52.2	55.3	35.4	81.4	91.2	59.6	68.7	89.3	38.6	*63.5*
BDF [23]	64.3	94.5	65.1	96.9	96.8	87.7	78.9	93.4	77.1	*83.7*
PrM.	79.2	82.4	88.0	92.2	89.7	70.8	92.0	87.8	88.5	*85.7*

References

[1] S. Ranasinghe, F. Al Machot, and H.C. Mayr, "A review on applications of activity recognition systems with regard to performance and evaluation," International Journal of Distributed Sensor Networks, vol. 12, no. 8, p. 1550147716665520, 2016.

[2] S.S. Agaian, J. Tang, and J. Tan, "Electronic imaging applications in mobile healthcare," 2019.

[3] Y. Wang, H. Jiang, M.S. Drew, Z.N. Li, and G. Mori, "Unsupervised discovery of action classes," in Proceedings of CVPR, pp. 17-22.

[4] S. Yan, J.S. Smith, W. Lu, and B. Zhang, "Multibranch Attention Networks for Action Recognition in Still Images," IEEE Transactions on Cognitive and Developmental Systems, vol. 10, no. 4, pp. 1116-1125, 2017.

[5] Y. Wang, Y. Li, X. Ji, "Human action recognition based on global gist feature and local patch coding," International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 8, no. 2, pp. 235-246, 2015.

[6] E. Park, X. Han, T.L. Berg, and A.C. Berg, "Combining multiple sources of knowledge in deep cnns for action recognition," in 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1-8, 2016.

[7] H.A. Qazi, U. Jahangir, B.M. Yousuf, and A. Noor, "Human action recognition using SIFT and HOG method," in 2017 International Conference on Information and Communication Technologies (ICICT), pp. 6-10, 2017.

[8] H.F. Nweke, Y.W. Teh, G. Mujtaba, and M. Al-Garadi, "Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions," Information Fusion, vol. 46, pp. 147-170, 2019.

[9] N. Ikizler, R.G. Cinbis, S. Pehlivan, and P. Duygulu, "Recognizing actions from still images," in 2008 19th International Conference on Pattern Recognition, pp. 1-4, 2008.

[10] L.J. Li, and L. Fei-Fei, "What, where and who? classifying events by scene and object recognition," In 2007 IEEE 11th international conference on computer vision, pp. 1-8, 2007.

[11] C .Thurau and V. Hlavác, "Pose primitive based human action recognition in videos or still images," in 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8, 2008.

[12] P. Li, J. Ma, and S. Gao, "Actions in still web images: visualization ,detection and retrieval," in International Conference on Web-Age Information Management, pp. 302-313, 2011.

[13] N. Shapovalova, W. Gong, M. Pedersoli, F.X. Roca, and J. Gonzalez, "On importance of interactions and context in human action recognition ",in Iberian conference on pattern recognition and image analysis, pp. 58-66, 2011.

[14] V. Delaitre, J. Sivic, and I. Laptev, "Learning person-object interactions for action recognition in still images," in Advances in neural information processing system, pp. 1503-1511, 2011.

[15] Y. Zheng, Y.J. Zhang, X. Li, and B.D. Liu, "Action recognition in still images using a combination of human pose and context information," in 2012 19th IEEE International Conference on Image Processing, pp. 785-788, 2012.

[16] F. Sener, C. Bas, and N. Ikizler-Cinbis, "On recognizing actions in still images via multiple features," in European Conference on Computer Vision, 2012, pp. 263-272.
[17] G. Sharma, F. Jurie, and C. Schmid, "Discriminative spatial saliency for image classification," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3506-3513, 2012.
[18] S. Maji, L. Bourdev, and J. Malik, "Action recognition from a distributed representation of pose and appearance," in CVPR 2011, pp. 3177-3184, 2011.
[19] B. Yao, X. Jiang, A. Khosla, A.L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in 2011 International Conference on Computer Vision, pp. 1331-1338, 2011.
[20] A. Prest, C. Schmid, and V. Ferrari, "Weakly supervised learning of interactions between humans and objects," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 3, pp. 601-614, 2011.
[21] F.S. Khan, R.M. Anwer, J. Van De Weijer, A.D. Bagdanov, and M. Felsberg, "Coloring action recognition in still images," International journal of computer vision, vol. 105, no. 3, pp. 205-221, 2013.
[22] F.S. Khan, J. Van De Weijer, R.M. Anwer, M. Felsberg, and C. Gatta, "Semantic pyramids for gender and action recognition," IEEE Transactions on Image Processing, vol. 23, no. 8, pp. 3633-3645, 2014.
[23] F.S. Khan, J. Van De Weijer, R.M. Anwer, A.D. Bagdanov, M. Felsberg, and J. Laaksonen, "Scale coding bag of deep features for human attribute and action recognition," Machine Vision and Applications, vol. 29, no. 1, pp. 55-71, 2018.
[24] T. Watanabe, S. Ito, and K. Yokoi, "Co-occurrence histograms of oriented gradients for pedestrian detection," in Pacific-Rim Symposium on Image and Video Technology, pp. 37-47, 2009.
[25] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," International journal of computer vision, vol. 42, no. 3, pp. 145-175, 2001.
[26] A. Oliva and A. Torralba, "Building the gist of a scene: The role of global image features in recognition," Progress in brain research, vol. 155, pp. 23-36, 2006.
[27] G. Lowe, "SIFT-The Scale Invariant Feature Transform," Int. J, vol. 2, pp. 91-110, 2004.
[28] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, pp. 91-110, 2004.
[29] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in null, p. 1470, 2003.
[30] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), pp. 524-531, 2005.
[31] M.A. Carreira-Perpinan and G.E. Hinton, "On contrastive divergence learning," in Aistats, pp. 33-40, 2005.
[32] G.E. Hinton, "Training products of experts by minimizing contrastive divergence," Neural computation, vol. 14, no.8, pp. 1771-1800, 2002.
[33] N. Le Roux, and Y. Bengio, "Deep belief networks are compact universal approximators," Neural computation, vol. 22, no. 8, pp. 2192-2207, 2010.
[34] R. Salakhutdinov and G. Hinton, "Deep boltzmann machines," in Artificial Intelligence and Statistics, pp. 448-455, 2009.
[35] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in Neural Networks for Perception, ed: Elsevier, pp. 65-93, 1992.
[36] I. Sutskever and G.E. Hinton, "Deep, narrow sigmoid belief networks are universal approximators," Neural computation, vol. 20, no. 11, pp. 2629-2636, 2008.
[37] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," International journal of computer vision, vol. 88, no. 2, pp. 303-338, 2010.

Azar Mahmoodzadeh received B.Sc., M.Sc. and Ph.D. degrees in Electrical Engineering from University of Shiraz, University of Shahed and University of Yazd, Iran, in 2005, 2008 and 2013, respectively. From 2009, she was with the Islamic Azad University, Shiraz Branch, Shiraz, Iran. Her research interests include pattern recognition and image and signal processing.

Energy Efficient Routing-Based Clustering Protocol Using Computational Intelligence Algorithms in Sensor-Based IoT

Mohammad Sedighimanesh
Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran
mohammad.sedighimanesh@gmail.com
Hessam Zandhessami*
Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran
Zandhessami@srbiau.ac.ir
Mahmood Alborzi
Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran
Mahmood_alborzi@yahoo.com
mohammadsadegh Khayyatian
Institute for science and technology studies, shahid Beheshti university, Tehran, Iran.
M_khayatian@sbu.ac.ir

Received: 28/Nov/2020 Revised: 28/Jan/2021 Accepted: 03/Apr/2021

Abstract

Background: The main limitation of wireless IoT sensor-based networks is their energy resource, which cannot be charged or replaced because, in most applications, these sensors are usually applied in places where they are not accessible or rechargeable. Objective: The present article's main objective is to assist in improving energy consumption in the sensor-based IoT network and thus increase the network's lifetime. Cluster heads are used to send data to the base station. Methods: In the present paper, the type-1 fuzzy algorithm is employed to select cluster heads, and the type-2 fuzzy algorithm is used for routing between cluster heads to the base station. After selecting the cluster head using the type-1 fuzzy algorithm, the normal nodes become the members of the cluster heads and send their data to the cluster head, and then the cluster heads transfer the collected data to the main station through the path which has been determined by the type-2 fuzzy algorithm. Results: The proposed algorithm was implemented using MATLAB simulator and compared with LEACH, DEC, and DEEC protocols. The simulation results suggest that the proposed protocol among the mentioned algorithms increases the network's lifetime in homogeneous and heterogeneous environments.
Conclusion: Due to the energy limitation in sensor-based IoT networks and the impossibility of recharging the sensors in most applications, the use of computational intelligence techniques in the design and implementation of these algorithms considerably contributes to the reduction of energy consumption and ultimately the increase in network's lifetime.

Keywords: Sensor-based IoT; Clustering and Routing; Type-1 and type-2 Fuzzy Algorithms; Computational Intelligence Techniques.

1- Introduction

The Internet of Things (IoT) is one of the technologies of the present era that bridges the gap between the physical and virtual worlds. In the wireless sensor-based IoT network, numerous large-scale sensor nodes are deployed, which leads to an increase in complexity [1]. These networks have an extensive range of applications such as disaster management, environmental monitoring, health care, identification and investigation of the subject of defense, etc. In these networks, after placing the sensors in the environment, all sensors collect data from the environment and then process and transfer the data to the base station [2]. Normally, the energy of the power supply of sensors is limited, irreplaceable, and rechargeable; for this reason, energy has become one of the most important factors in these networks. Hence, the reduction of energy consumption in sensors has increased the network's lifetime, which has caused these networks to be one of the interesting research subjects among researchers. There are many approaches and techniques to reduce energy consumption in these networks, one of which is the topology control that increases the system's efficiency and reduces energy consumption [3][4]. Clustering and routing are of the most effective techniques in controlling the topology control. Clustering in sensor-based IoT networks has advantages, including scalability, energy consumption, and reduction of data transmission latency in routing, Moreover, in clustering, the energy consumption is balanced between the sensors, which will increase the network's life.

* Corresponding Author

In clustering, first numerous sensors are selected as cluster heads, which is usually performed randomly for the first time, and then normal sensors (nodes) become the members of these cluster heads, and the sensors send their collected data from the environment to their cluster head, and the cluster head sends the collected and aggregated data in a single-hop manner (each cluster head separately) to the base station. However, in more optimal routing methods, the shortest path from the cluster heads to the base station is created, which reduces the energy loss of the cluster heads. See Figure (1) [5][5].

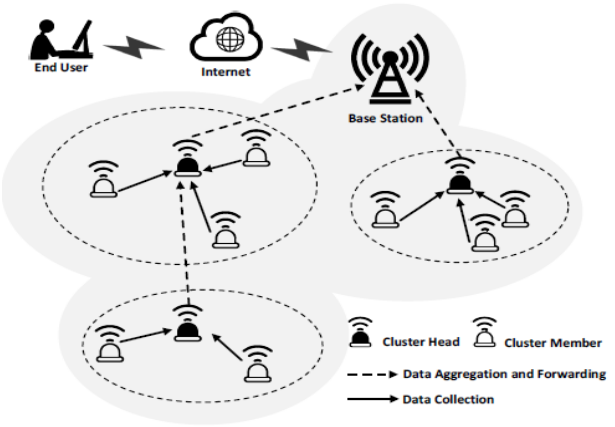


Fig.1. A clustered wireless sensor-based IoT network.[5]

Clustering leads to the local formation of the path within the cluster; hence, the size of the routing table of each node is reduced, and the scalability in the network is enhanced. Clustering saves the communication bandwidth and intra-cluster communications and reduces the redundancy resulted from message exchange between sensors. Moreover, the sensors are only involved in connection with their cluster heads and are not affected by changes in the levels between cluster heads. As a result, the maintenance overload of the network topology is reduced [7][8].

In addition to clustering algorithms, proper routing algorithms play an effective role in reducing energy consumption and, consequently, increasing the network's lifetime. Designing routing algorithms in sensor-based IoT networks is challenging since the network energy limitations must also be considered during design. In general, routing is the way of sending the data packet from source to destination. Network routing protocols are divided into two types: 1) Flat routing method, 2) Hierarchical, or classified routing method [4],[7],[9].

In flat routing, all nodes (sensors) have similar roles, and features, and data transmission is outspread in the network similar to the flood flow. The flat routing method is somewhat appropriate in relatively small networks, but in large networks, data processing, and high bandwidth are needed due to a large amount of collected data in the sensors, limiting the application of flat routing. In hierarchical routing, the data collected in each cluster head is transferred to the upper cluster head, and this process continues until data to be sent to the base station, which results in scalability, shorter data transmission distance, lower energy consumption, and also lower load.

In the present article, a new routing and clustering energy-aware algorithm using computational intelligence techniques called RCECI was presented for wireless sensor-based IoT networks. In the proposed method, the cluster heads at the base station and the path from the cluster heads to the base station are selected using the computational intelligence technique, and this data is sent to all nodes in the network. All normal nodes become the members of the existing cluster heads and are then organized, and after the data is collected by the sensors and sent to the cluster heads, the collected data is transferred to the base station through the path specified by the computational intelligence algorithms. Finally, some experiments are performed on the proposed algorithm (RCECI), and the results are compared with LEACH, DCE, DEEC, and SEP algorithms. The results indicate the efficiency of the proposed algorithm in energy consumption, network coverage, and the number of data packets sent to the base station.

The structure of the article is as follows. A review of some related works is discussed in Section 2. The proposed algorithm is described in Section 3. The experimental results are presented and compared with other algorithms in Section 4, and the conclusions are given in Section 5.

2- Previous Works

Numerous routing and clustering algorithms have been developed for wireless sensor-based IoT networks; the LEACH algorithm is one of the most important and basic [10][11]. The clustering operation in LEACH is performed randomly. Cluster heads are replaced in each round. Every single round is divided into two phases: the setup phase to select the cluster head and the steady-state phase to send data. The setup phase includes selecting the cluster head node, declaring the cluster head to the entire network, and joining normal nodes to the cluster head. The cluster heads are randomly selected at the start of the LEACH algorithm. The selection rule is that each sensor randomly generates a number 0 or 1. If these numbers are less than the threshold, the sensor is considered as a cluster head. A formula for the threshold limit T(n) is indicated in Equation (1).

$$T(n) = \begin{cases} \dfrac{p}{1 - p\left[r \bmod \left(\frac{1}{p}\right)\right]} & n \in G \\ 0 & otherwise \end{cases} \quad (1)$$

Where, p implies the expected percentage of cluster head nodes in the sensor population. R denotes the current

phase number. G is referred to as a group of nodes that have not been converted to cluster head nodes in the last 1/p step. LEACH's main disadvantage is that the sensor node with considerably low energy is probable to be selected as the cluster head, and the cluster heads send the data directly to the base station in a single-hop communication. Hence, this method increases the energy consumption of the cluster head [12][13].

The Stable Election Protocol (SEP) [14][5] and Distributed Energy-Efficient Clustering (DEEC) [15][5] algorithms have been specifically designed for heterogeneous networks. SEP is aimed to prolong the stability period of two-level heterogeneous networks and includes two types of nodes based on initial energy: normal nodes and advanced nodes. Also, SEP operates similarly to LEACH; however, the rotation round of the cluster heads and the probability of their selection as the cluster head is directly related to the initial energy of the nodes. Nevertheless, SEP does not consider the remaining energy of the sensors for a multilevel heterogeneous network.

Contrary to SEP, DEEC considers the sensor's initial energy and remaining energy in selecting the cluster head, which improves the network's lifetime. DEEC shows better performance than LEACH and SEP in multilevel heterogeneous networks. In DEEC, the central station is assumed to be at the center of the network; therefore, this method cannot be applied if it is extremely far from the sensor nodes. Besides, although cluster head selection is improved by changing the probability function and can guarantee that sensors with relatively high remaining energy and initial energy are more probable to become cluster heads, low-energy sensors still have a good chance of becoming cluster heads [5].

The DCE algorithm for heterogeneous networks is on the basis of two-phase cluster head selection. In DCE [5], the cluster head is selected in two phases. In the first phase, the test cluster heads are picked according to the initial energy and the remaining energy of the sensors. In the second phase, if the test cluster heads have lower energy than one of the cluster members, they are replaced with them to determine the final cluster head. The use of two phases in cluster head selection ensures that nodes with higher energy have a better chance of becoming cluster heads. Moreover, this algorithm does not consider some parameters such as density and centrality in the selection of cluster heads and select the cluster heads based on the initial energy and remaining energy.

The EECA[1] algorithm [2]: The node's residual energy, distance and data overhead have been used in selecting the cluster head in the EECA method. Clustering in this method is done in two steps; in the first stage, clustering is done according to energy and distance, and in the second stage, clusters are optimized using the K-means clustering algorithm. The information is sent to the base station, after collecting the data and aggregating it by the headers.

1. Energy Efficient Clustering Algorithm for Wireless Sensor Networks

3- Proposed Method

The assumptions considered in the proposed method are as follows:

- The simulation is performed in several scenarios; the sensors' energy is either homogeneous or heterogeneous depending on the scenario. In the homogeneous scenario, all nodes' energy is equal to 0.5 J, and when the environment is heterogeneous, the energy of half of the sensors equals 1 J.
- All sensor nodes are placed randomly and uniformly, and the sensors are aware of neighboring nodes and their positions, as well as the base station's position.
- The base station can be in different positions and also movable or fixed depending on various scenarios.
- According to the scenario, the sensors may not be fixed, and their position may change. Nodes' instability does not mean relocating the sensors by remote control but only involves ground displacements including displacements or erosion caused by external objects that lead to in-place changes.
- Since the authors have assumed that mobility is done by external factors, there is no energy consumption in nodes.
- Sensors can change their signal strength for transmission based on nodes' distance.
- Cluster heads can aggregate the collected data to remove additional data.

3-1- Energy Consumption Model

Energy consumption in the sensor-based IoT network consists of three sections: data transmission, data reception, and data processing. Equation (2) presents the energy model as follows [16][6]:

$$\begin{cases} P_T(K) = E_{elec} * K + E_{amp} * d^{\gamma} * K \\ P_R(K) = E_{elec} * k \\ P_{cpu}(K) = E_{cpu} * k \end{cases} \quad (2)$$

Where, P_T, P_R, and P_{cpu} denote the energy consumption of transmission, reception, and processing of k-bits data, respectively. E_{elec}, E_{amp}, and E_{cpu} are the energy consumption (nJ/bit) for transmitting per bit in the radio radius, the energy required for transmitting with a radius higher than E_{elec}, and the energy needed for processing per bit, respectively. The total energy consumption of k bits, according to Equation (3), is as follows [16][6]:

$$P_{Total} = P_T + P_R + P_{cpu} = k(2E_{elec} + E_{cpu} + E_{amp} \times d^{\gamma}) \quad (3)$$

Equation (3) indicates that energy consumption is directly related to the data length. In the case that the data sent first is less, less energy will be consumed. If the transmission

distance is less than the threshold, the energy consumption would be in relation to d^2, and if the transmission distance is greater than the threshold, it will be in relation to d^4. Thus, the shorter the transmission distance, the lower the energy consumption.

3-2- Description of Proposed Method

The most important objective of the present article is to use computational intelligence algorithms to improve the sensor-based IoT network's lifetime. Therefore, the authors have presented a method based on computational intelligence algorithms to minimize the energy consumption of sensors in the network. In this section, the proposed algorithm is discussed.

Proposed Algorithm

1. For k=1: number of clusters
2. Calculate remain energy, density and centrality of nodes;
3. Calculate fuzzy a mount of nodes with Relay Fuzzy Logic Type1 ();
4. Sort nodes according to fuzzy amount;
5. Select 10 percent of node with maximum fuzzy amount as cluster heads;
6. End_For
7. For k=1: number of clusters
8. For i=1: number of nodes
9. If node_i is normal node
10. Node_i joins to nearest cluster head_k;
11. End_IF
12. End_For
13. End_For
14. Calculate fuzzy amount of each cluster head with Route Fuzzy Logic Type 2();
15. For i = cluster heads from closest to farest to sink
16. While cluster head_i does not reach to sink
17. If reach to node which has a route to sink
18. Break;
19. End_If
20. Neighbor_i=half of closest cluster heads to cluster head_i;
21. If Neighbor_i is empty
22. Cluster head_i connects to sink;
23. End_If
24. Sort neighbor_i according to fuzzy amount decently;
25. For j=sorted neighbor_i
26. If cluster head_j near than cluster head_i to sink
27. Cluster head_i connects to cluster head_j;
28. Break;
29. End_If
30. End_If
31. If cluster head_i is nearest to sink
32. Cluster head _i connects to sink directly;
33. End_If
34. End_While
35. End_For

The suggested algorithm (RCECI) in the proposed method is implemented after the sensor nodes are distributed in the desired area. In this method, the cluster heads are performed by the type-1 fuzzy algorithm, and also the path between the cluster heads to the base station is done by the type-2 fuzzy algorithm; the operations are performed accurately at the base station. The proposed algorithm operates based on round, and each round includes two phases: setup and steady-state. The network initializes all sensor nodes before the first round and notifies them by sending a message containing information about the total network energy, synchronization time, and start order.

In each round, after selecting the cluster heads by the type-1 fuzzy algorithm at the base station and also specifying the path between the cluster heads to the base station using the type-2 fuzzy algorithm, a message is sent from the base station to the identified cluster heads to inform the related sensor about its role as a cluster head and the path of data transfer to the base station in the current round. Then, the sensors selected as cluster heads inform the other sensors of their role as cluster heads in the current round. For this purpose, each header broadcasts an announcement message throughout the network. Normal sensors then become the members of cluster heads, which require the least energy and the shortest distance to communicate with them. After each normal sensor makes a decision about the cluster, it wants to join in the current round and notifies the cluster head of this decision with a Join-REQ message.

The cluster heads, depending on the number of cluster members, create a Time-Division Multiple Access (TDMA) schedule and notify their members to prevent data collisions during the transfer. Consequently, there would be no collision between the data within a cluster. After the TDMA schedule is identified by all member nodes of the clusters, the setup phase is completed, and the Steady-state (data transfer phase) starts.

At this stage, the network performance is divided into a number of time slots; each sensor transfers data to the related cluster header at a specified period. In the proposed algorithm, the base station transmits simultaneous pulses to the network sensors so that all the sensors begin the starting phase together. After identifying the cluster heads, forming the clusters, and determining the TDMA schedule, each sensor transmits its data to the cluster at a specific time using the Carrier-Sense Multiple Access (CSMA) method and a unique distribution code. Cluster heads also apply the same distribution code to send their data. When the cluster head has data to send, it should listen to the channel to ensure that no other sensor has data to send at that time. If the channel is empty, it should transfer its data to another cluster head or base station, and if the channel is busy, it waits for a random period and sends its data after the channel is empty.

3-3- Selection of Cluster Heads and Routing

The optimal selection of cluster heads is performed using the type-1 fuzzy algorithm. Fuzzy logic is a theory for acting in conditions of uncertainty; the theory is capable of mathematically formulating many inaccurate and ambiguous concepts, variables, and systems and provides a basis for reasoning, deduction, control, and decision-making in conditions of uncertainty. Most of our decisions and measures are in conditions of uncertainty, and clear and unambiguous states are extremely rare. Figure (2) demonstrates the basic structure of the fuzzy system.

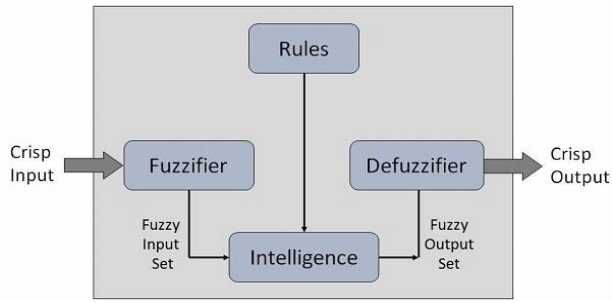


Fig. 2. Structure of the fuzzy algorithm [17]

After placing the nodes for clustering in the proposed algorithm, type-1 fuzzy logic is applied to select the cluster head. In this algorithm, the triangular membership function with three parameters has been used, the first parameter of which is the remaining energy and is obtained by the equation as follows:

$$E = E_R \quad (4)$$

Density is equal to the ratio of neighboring nodes to total nodes; the higher it is, the node is more suitable for becoming a cluster head.

$$D = \frac{N_n}{N_T} \quad (5)$$

N_n is referred to as the neighboring nodes, and N_T denotes the total nodes.

Centrality was applied, which is referred to as the centrality of the node relative to its neighbor nodes and is the sum of the total distance of the node from its neighbors. A lower value indicates that the node needs less energy as a cluster head and is more suitable for becoming a cluster head.

$$C = \sum d_i \quad (6)$$

d_i is the distance to the neighboring node.

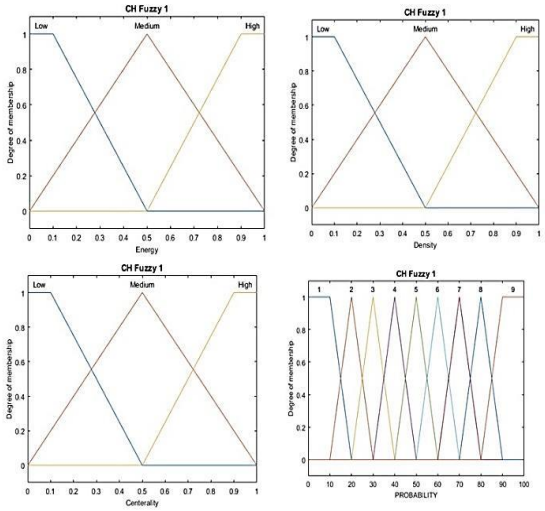


Fig. 3. Membership functions of remaining energy, density, and centrality, and the output membership function

After the membership functions of all three parameters, as well as the output membership function, are formed, the fuzzy rules are developed according to Table (1) and given to the fuzzy network.

Table 1. Fuzzy rules

Row	Energy	Density	Centrality	Probability
1	Low	Low	Low	3
2	Low	Low	Medium	2
3	Low	Low	High	1
4	Low	Medium	Low	4
5	Low	Medium	Medium	3
6	Low	Medium	High	2
7	Low	High	Low	5
8	Low	High	Medium	4
9	Low	High	High	3
10	Medium	Low	Low	4
11	Medium	Low	Medium	3
12	Medium	Low	High	2
13	Medium	Medium	Low	6
14	Medium	Medium	Medium	5
15	Medium	Medium	High	4
16	Medium	High	Low	7
17	Medium	High	Medium	6
18	Medium	High	High	5
19	High	Low	Low	7
20	High	Low	Medium	6
21	High	Low	High	5
22	High	Medium	Low	9
23	High	Medium	Medium	8
24	High	Medium	High	7
25	High	High	Low	9
26	High	High	Medium	9
27	High	High	High	8

Then, each node's fuzzy value is obtained, and after implementing the fuzzy algorithm, 10% of the nodes with the best values are specified as cluster heads. Also, the base station informs the cluster heads of their role, and then the cluster heads send a message to the network announcing that they are cluster heads. Afterward, each normal node that receives this message joins the cluster head based on its energy and distance to the cluster head node and announces its membership to the desired cluster head.

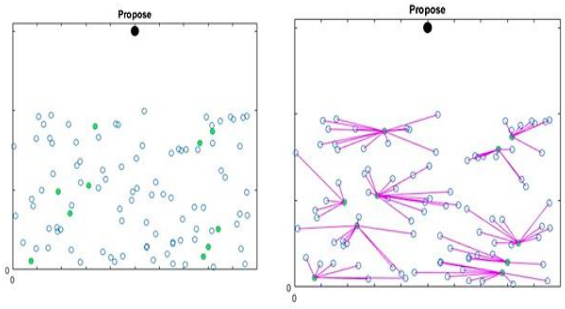


Fig. 4. Selection of cluster head by type-1 fuzzy logic and joining the normal nodes to the cluster head

After selecting the cluster head and carrying out the membership operations for cluster heads, normal sensors collect the sensed data from the network and send it to their cluster head; then, the cluster heads must transfer data to the base station. For this purpose, the type-2 fuzzy algorithm has been applied to obtain the best nodes for creating a minimum tree from the cluster heads toward the sink. Figure (5) indicates the type-2 fuzzy logic diagram block.

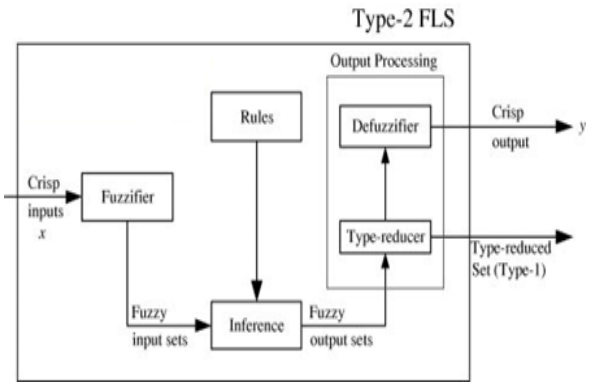


Fig. 5. Type-2 Fuzzy Logic Diagram[18]

In the setup phase, after determining which sensors have been selected for becoming the cluster heads, the type-2 fuzzy algorithm is applied for path selection. First, the cluster heads are arranged based on their distance to the base station, and then the routing algorithm is implemented for all cluster heads so that the neighbor of each cluster head is equal to half of the cluster heads closer to it; if there is no neighbor with these conditions, the cluster head is connected directly to the base station. In the case that the cluster heads have neighbors, the type-2 fuzzy algorithm is implemented for the neighbor cluster heads, and they are arranged in descending order based on their fuzzy value. Then, the neighbor with the maximum fuzzy value is picked, provided that the neighbor cluster head is closer to the base station compared to the cluster head itself, and if such conditions do not exist, the cluster head connects directly to the base station. In order to implement the type-2 fuzzy algorithm in routing, first, the type-2 fuzzy triangular membership function for all cluster heads is specified based on the parameters remaining energy, the distance from the cluster head to the base station, and the distance to the current cluster head.

$$E = E_r \tag{7}$$

E_r denotes the remaining energy of the sensors.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{8}$$

(X_3, y_3) is the space coordinate of the cluster head, and (x_4, y_4) is the space coordinate of the base station.

$$d_{ch} = \sqrt{(x_3 - x_4)^2 + (y_3 - y_4)^2} \tag{9}$$

(x_3, y_3) is the space coordinate of the node, and (x_4, y_4) is the space coordinate of the current cluster head.

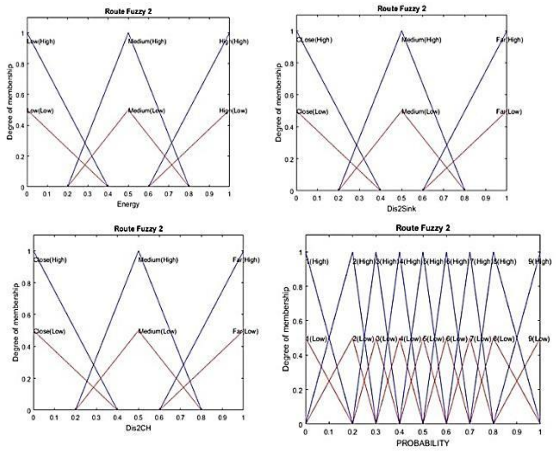


Fig. 6. Membership functions of remaining energy, distance to sink, and distance to cluster head node, and output membership function

After the membership functions are specified, the fuzzy rules are defined and given to the fuzzy network. The routing algorithm is implemented to create a path between the cluster heads and the base station. Then the data is transferred from the cluster heads toward the base station according to the created path.

Table 2. Type-2 fuzzy rules

Row	Energy	Dis2Sink	Dis2CH	Probability
1	Low	Low	Low	5
2	Low	Low	Medium	4
3	Low	Low	High	3
4	Low	Medium	Low	4
5	Low	Medium	Medium	3
6	Low	Medium	High	2
7	Low	High	Low	3
8	Low	High	Medium	2
9	Low	High	High	1
10	Medium	Low	Low	7
11	Medium	Low	Medium	6
12	Medium	Low	High	5
13	Medium	Medium	Low	6
14	Medium	Medium	Medium	5
15	Medium	Medium	High	4
16	Medium	High	Low	4
17	Medium	High	Medium	3
18	Medium	High	High	2
19	High	Low	Low	9
20	High	Low	Medium	9
21	High	Low	High	8
22	High	Medium	Low	9
23	High	Medium	Medium	8
24	High	Medium	High	7
25	High	High	Low	7
26	High	High	Medium	6
27	High	High	High	5

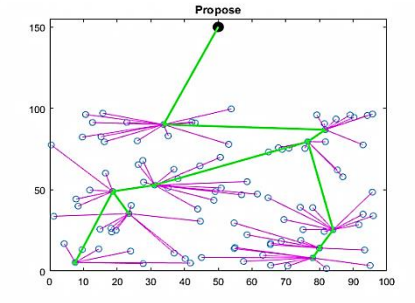


Fig.7. Path formation and data transfer to the base station

The pseudo-code related to clustering and routing in the proposed algorithm is presented as follows.

Clustering Fuzzy Type1 Algorithm

1. Calculate Remaining Energy of nodes;
2. Calculate Density of nodes;
3. Calculate Centrality of nodes;
4. Create Membership Function of Remaining energy;
5. Create Membership Function of Density;
6. Create Membership Function of Centrality;
7. Do Fuzzification;
8. Do Inference according to Rule Base;
9. Create Membership Function of Output;
10. Do Defuzzification;
11. Calculate fuzzy a mount of nodes;

Routing Fuzzy Type2 Algorithm

1. Calculate Remaining Energy of cluster heads;
2. Calculate Distance to sink of cluster heads;
3. Calculate Distance to neighbor cluster heads;
4. Create Membership Function of Remaining energy;
5. Create Membership Function of Distance to sink;
6. Create Membership Function of Distance to neighbor cluster heads;
7. Do Fuzzification;
8. Do Inference according to Rule Base;
9. Create Membership Function of Output;
10. Do Type Reduction;
11. Do Defuzzification;
12. Calculate fuzzy a mount of nodes;

4- Results

The proposed protocol in the present article is implemented and tested with MATLAB software in two scenarios: 1) Constant energy of all sensors (homogeneous environment) and the fixed base station 2) Not equal energy for all sensors (heterogeneous environment), movable sensors, and the clockwise movement of the base station at a speed of 10 degrees per round.

Table 3. General parameters for simulations

Parameter	Value	Parameter	Value
E_0	0.5J	ε_{fs}	10 pJ/bit/m^2
E_{elect}	5 nJ/bit	ε_{mp}	0.0013 pJ/bit/m^4
E_{DA}	5 nJ/bit/message	L_D	4000 bits
d_{break}	87.7 m	L_c	16 bits

4-1- Scenario I

In this scenario, 150 sensor nodes were randomly distributed over an area of 200*200 m^2. In this test, the initial energy of all sensors is identical (homogeneous environment) and equals 0.5 J. The base station is also fixed in the position (100*350).

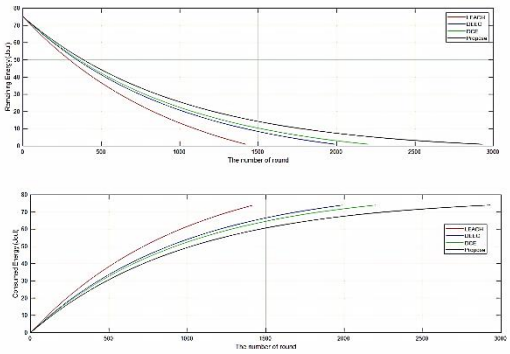


Fig. 8. Comparison of energy consumption and remaining energy in the network

As shown in Figure (8), the proposed method (RCECI) has improved the network's energy consumption compared to previous algorithms. The proposed algorithm increases networks' lifetime by approximately 34%, 45%, and 56% compared to the DCE, DEEC, and LEACH algorithms, respectively.

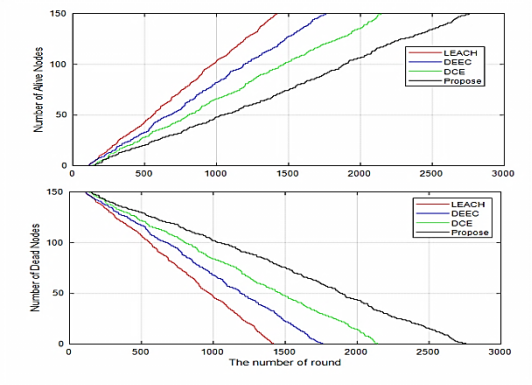


Fig. 9. Comparison of the number of live and dead nodes in the network

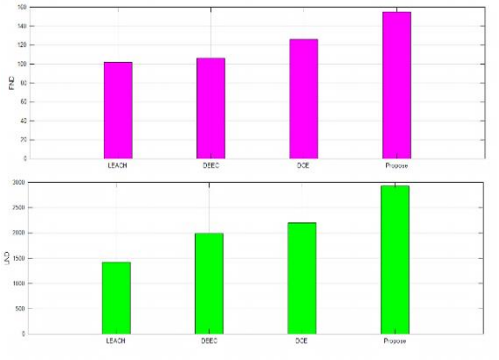


Fig. 10. Comparison of the first node die and last node die

According to Figures (9) and (10), it can be concluded that the proposed algorithm is superior than other algorithms in both the first node die (FND) and the last node die (LND) because sensors survive in most rounds. As shown in Figures (9) and (10), in the proposed algorithm, FND occurs in the 155th round, and the LND is taken place in the 2930s round. Also, in the DCE algorithm, the FND occurs in the 126th round, and the LND is taken place in the 2201st round, which shows the performance of the proposed algorithm is superior than other algorithms.
In the following, we examine the proposed scenario in environments of different sizes.

Table 4. the proposed scenario 1 in environments of different sizes

Environment size										
600*600 m^2			400*400m^2			300*300m^2				
100% node	50% node	First node	100% node	50% node	First node	100% node	50% node	First node	**lifetime / method**	**Node number**
530	268	38	1065	532	75	1335	668	94	LEACH	**150** nodes
692	330	40	1380	668	83	1720	837	104	DEEC	
788	503	47	1570	1000	92	1950	1256	117	DCE	
1020	512	56	2090	1305	113	2685	1670	144	Propose	

As shown in Table 4, the larger the environment, the sooner the sensors die, and this is because the distance between the sensors increases, and they spend more energy on communicating with each other. Besides, the proposed method performs better in different interval environments than the previous algorithms, and the sensors survive longer rounds, which shows that the proposed algorithm does not lose its working efficiency as the environment grows.

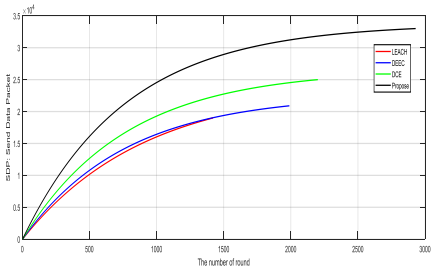


Fig.11. The number of packets sent to the base station

As demonstrated in Figure (11), the numbers of packets sent to the base station in the proposed algorithm, DCE, DEEC, and LEACH are approximately equal to $3.3*10^4$, $2.5*10^4$, $2.09*10^4$, and $1.9*10^4$, respectively, which shows that the number of packets sent to the base station in the proposed method is more than DCE, DEEC, and LEACH algorithms by about 32%, 50%, and 62%, respectively.

4-2- Scenario II

In this scenario, 100 sensor nodes were randomly distributed in an area of 100*100 m^2. The sensors are movable, and their mobility is due to external factors, and the sensor's energy is not consumed, and only its position changes. The sensors' initial energy is varied (heterogeneous environment, half of the sensors have twice the energy of the others), and the base station in the position 50*150 moves around the environment in a clockwise manner at a speed of 10 degrees per round.

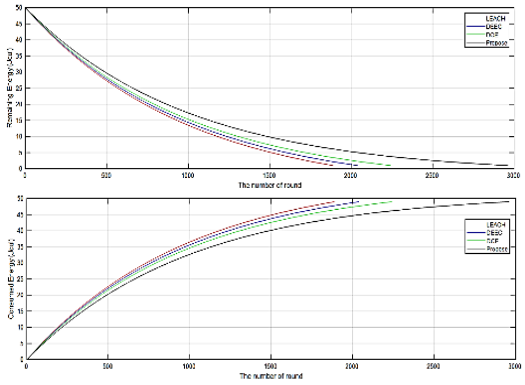


Fig. 12. Comparison of energy consumption and remaining energy in the network

As shown in Figure (12), the proposed method improved the network's energy consumption compared to previous algorithms. The proposed algorithm increases the network's lifetime compared to the DCE, DEEC, and LEACH algorithms by approximately 32%, 41%, and 54%, respectively.

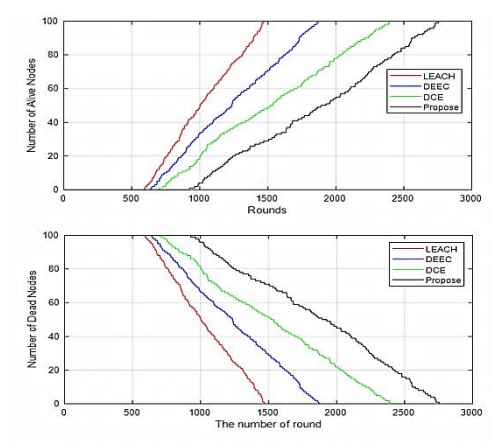


Fig.13 .Comparison of the number of live and dead nodes in the network

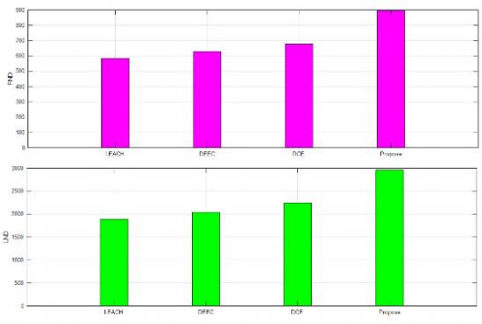


Fig.14 .Comparison of the first node die and last node die

According to Figures (13) and (14), it can be concluded that the proposed algorithm is superior than other algorithms regarding both FND and LND, and sensors survive in most rounds. In the proposed algorithm, FND occurs in the 896th round, and LND is taken place 2960s round. Also, in the DCE algorithm, FND occurs in the 629th round, and LND is taken place in the 2242nd round, which indicates the better performance of the proposed method compared to other algorithms.

In the following, we examine the proposed scenario in environments of different sizes.

Table 5. the proposed scenario 2 in environments of different sizes

Environment size										
600*600 m^2			400*400m^2			200*200m^2				
100% node	50% node	First node	100% node	50% node	First node	100% node	50% node	First node	**lifetime / method**	**Node number**
323	137	98	918	390	280	1530	645	467	LEACH	**150** nodes
362	216	104	1020	612	299	1685	1012	499	DEEC	
383	222	116	1095	728	334	1812	1208	558	DCE	
461	302	143	1362	885	414	2290	1490	692	Propose	

The results obtained in Table 5 show that the proposed method is not affected by environmental change and is still better than previous methods in which the sensors survive further rounds. As it is clear, with increasing the length of the environment, using the energy by sensors will be increased due to increment of the distance of communication and leads to a reduction in the total lifetime of the network.

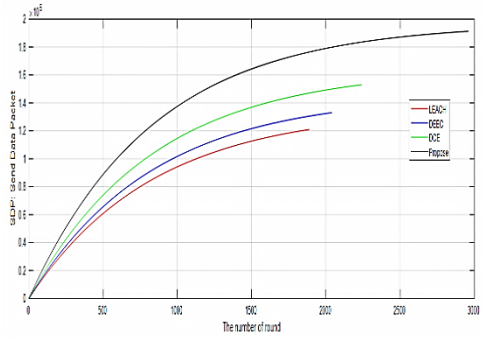


Fig. 15. The number of packets sent to the base station

As indicated in Figure (15), the numbers of packets sent to the base station are approximately equal to $19.13*10^4$, $15.3*10^4$, $13.31*10^4$, and $12.1*10^4$ in the proposed method, DCE,

DEEC, and LEACH, respectively, which shows that the number of packets sent to the base station in the proposed method is more than DCE, DEEC, and LEACH algorithms by about 25%, 36%, and 51%, respectively.

5- Discussion and Conclusion

In the present research, a clustering-based routing protocol using computational intelligence algorithms was proposed for wireless sensor-based IoT networks. In this algorithm, first, the appropriate cluster heads are specified from the existing sensor nodes using

the type-1 fuzzy algorithm, and then the sensors join these cluster heads and send the collected data to their cluster heads. Then, the cluster heads perform hierarchically to send their collected data to the sink, so that select the best cluster head to send the data from the cluster heads to the base station in a multi-hob manner using the type-2 fuzzy algorithm. According to the simulation, it can be concluded that the network's lifetime (based on the first sensor die and the last node die) has been improved by approximately 33%, 40%, and 52% compared to DCE, DEEC, and LEACH algorithms, respectively in the case of changing the number of sensors, altering the environment, homogeneity and heterogeneity of sensors and movability of the base station. Furthermore, the number of packets sent to the base station has been evaluated in different scenarios, which indicates that the proposed algorithm led to a significant improvement in the number of packets sent to the base station.

The following suggestions may be made for future research:

- The use of other computational intelligence algorithms and integrating them to improve routing between cluster heads. For example, the use of machine learning algorithms
- The use of various criteria for creating paths and selecting cluster heads. For instance, the use of the distance between the Cluster head or the number of members of each cluster

References

[1] F. Gregorio, G. González, C. Schmidt, and J. Cousseau, "Internet of Things," in Signals and Communication Technology, 2020.

[2] M. Bavaghar, A. Mohajer, and S. T. Motlagh, "Energy Efficient Clustering Algorithm for Wireless Sensor Networks." Journal of Information Systems and Telecommunication (JIST), pp. 238–247, doi: 10.7508/jist.2019.04.001.

[3] S. Rani, R. Talwar, J. Malhotra, S. H. Ahmed, M. Sarkar, and H. Song, "A novel scheme for an energy efficient internet of things based on wireless sensor networks," Sensors (Switzerland), 2015, doi: 10.3390/s151128603.

[4] J. V. V. Sobral, J. J. P. C. Rodrigues, R. A. L. Rabêlo, K. Saleem, and V. Furtado, "LOADng-IoT: An enhanced routing protocol for internet of things applications over low power networks," Sensors (Switzerland), 2019, doi: 10.3390/s19010150.

[5] R. Han, W. Yang, Y. Wang, and K. You, "DCE: A distributed energy-efficient clustering protocol for wireless sensor network based on double-phase cluster-head election," Sensors (Switzerland), 2017, doi: 10.3390/s17050998.

[6] M. Sedighimanesh* and H. Z. H. and A. Sedighimanesh, "Routing Algorithm based on Clustering for Increasing the Lifetime of Sensor Networks by Using Meta-Heuristic Bee Algorithms," International Journal of Sensors, Wireless Communications and Control, vol. 10, no. 1. pp. 25–36, 2020, doi: http://dx.doi.org/10.2174/2210327909666190129154802.

[7] K. Thangaramya, K. Kulothungan, R. Logambigai, M. Selvi, S. Ganapathy, and A. Kannan, "Energy aware cluster and neuro-fuzzy based routing algorithm for wireless sensor networks in IoT," Comput. Networks, 2019, doi: 10.1016/j.comnet.2019.01.024

[8] F. Fanian and M. Kuchaki Rafsanjani, "Cluster-based routing protocols in wireless sensor networks: A survey based on methodology," J. Netw. Comput. Appl., vol. 142, pp. 111–142, Sep. 2019, doi: 10.1016/J.JNCA.2019.04.021.

[9] B. Bhushan and G. Sahoo, "Routing protocols in wireless sensor networks," in Studies in Computational Intelligence, 2019.

[10] J. Bhola, S. Soni, and G. K. Cheema, "Genetic algorithm based optimized leach protocol for energy efficient wireless sensor networks," J. Ambient Intell. Humaniz. Comput., vol. 11, no. 3, pp. 1281–1288, 2020.

[11] J. H. Lee, "Energy-efficient clustering scheme in wireless sensor network," Int. J. Grid Distrib. Comput., 2018, doi: 10.14257/ijgdc.2018.11.10.09.

[12] A. Kochhar, P. Kaur, P. Singh, and S. Sharma, "Protocols for wireless sensor networks: A survey," Journal of Telecommunications and Information Technology. 2018, doi: 10.26636/jtit.2018.117417.

[13] B. Bhushan and G. Sahoo, "Routing protocols in wireless sensor networks," in Studies in Computational Intelligence, 2019.

[14] G. Smaragdakis, I. Matta, and A. Bestavros, "SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks *," 2nd Int. Work. Sens. Actor Netw. Protoc. Appl., 2004, doi: 10.3923/jmcomm.2010.38.42.

[15] L. Qing, Q. Zhu, and M. Wang, "Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks," Comput. Commun., 2006, doi: 10.1016/j.comcom.2006.02.017.

[16] M. Sedighimanesh, H. Zandhesami, and A. Sedighimanesh, "Presenting the Hybrid Algorithm of Honeybee - Harmony in Clustering and Routing of Wireless Sensor Networks," Int. J. Sensors, Wirel. Commun. Control, 2018, doi: 10.2174/2210327908666181029094346.

[17] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," Fuzzy Sets Syst., 1988, doi: 10.1016/0165-0114(88)90113-3.

[18] A. Kousar, N. Mittal, and P. Singh, "An Improved Hierarchical Clustering Method for Mobile Wireless Sensor Network Using Type-2 Fuzzy Logic," in Lecture Notes in Electrical Engineering, 2020, doi: 10.1007/978-3-030-30577-2_11.

Mohammad Sedighimanesh received the B.S. degree in Information technology from Sufi Razi University, Zanjan Branch, Iran in 2009, and M.S. degree in Information technology from Azad University, Qazvin, Iran, in 2013. Currently she is Ph.D. Candidate in Azad University, Tehran Branch, Iran. Her research interests include Wireless sensor network, IOT (internet of things), Blockchain, Web mining and Data mining.

Hessam Zandhessami is Assistant-Professor of management in the Faculty of management and economic at Azad University, Science and research Branch, Iran. He received the B.S. degree in industrial management from Shiraz University, Iran in 1998, and M.S. degree in industry management from Shahid Beheshti University, Iran, in 2001. and received his Ph. D degree in Industrial management from SRIBIA university, Iran,2008. He is conducting research activities in the areas of techno-Entrepreneurship and information technology.

Mahmood Alborzi is a full-time Assistant-Professor of Information technology at Azad University, Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran. He received his Ph.D. degree PhD in Artificial Neural Networks Brunel University, UK, England in 1996. He is conducting research activities in the areas of information technology, Artificial Neural Networks, data mining.

Mohammadsadegh Khayatian is an Assistant-Professor of Technology Management in the Institute for Science and Technology Studies at Shahid Beheshti University, Tehran, Iran. He received his Ph.D. degree in Technology Management from Allameh Tabataba'i University, Iran in 2015. His research interests include Science and Technology Policies, Commercialization of research and technology, Supporting knowledge-intensive firms, Funding technology and innovation development, innovation policy, Innovation intermediaries.

Secured Access Control in Security Information and Event Management Systems

Leila Rikhtechi
Department of Computer Engineering, Faculty of Engineering, Arak University, Arak 38156-8-8349, Iran
l-rikhtechi@phd.araku.ac.ir
Vahid Rafe*
Department of Computer Engineering, Faculty of Engineering, Arak University, Arak 38156-8-8349, Iran
v-rafe@araku.ac.ir
Afshin Rezakhani
Department of Computer Engineering, Faculty of Engineering, Ayatollah Boroujerdi University, Boroujerd, Iran
rezakhani@abru.ac.ir

Received: 15/Jul/2020 Revised: 25/Aug/2020 Accepted: 01/Jan/2021

Abstract
Nowadays, Security Information and Event Management (SIEM) is very important in software. SIEM stores and monitors events in software and unauthorized access to logs can prompt different security threats such as information leakage and violation of confidentiality. In this paper, a novel method is suggested for secured and integrated access control in the SIEM. First, the key points where the SIEM accesses the information within the software is specified and integrated policies for access control are developed in them. Accordingly, the threats entered into the access control module embedded in this system are carefully detected. By applying the proposed method, it is possible to provide the secured and integrated access control module for SIEM as well as the security of the access control module significantly increases in these systems. The method is implemented in the three stages of the requirements analysis for the establishment of a secure SIEM system, secure architectural design, and secure coding. The access control module is designed to create a secured SIEM and the test tool module is designed for evaluating the access control module vulnerabilities. Also, to evaluate the proposed method, the dataset is considered with ten thousand records, and the accuracy is calculated. The outcomes show the accuracy of the proposed method is significantly improved. The results of this paper can be used for designing an integrated and secured access control system in SIEM systems.

Keywords: Software; Logs; Security Information and Event Management; Integrated Access Control.

1- Introduction

The ever-increasing expansion of software as a major element in everyday activities and the high cost of program failure has led to the emergence of tools for evaluating software. The software has been made for many years and its security has been taken into account more or less. Moreover, as threats become smarter, software security has become more and more important. Security is a requirement that should be considered in software [1]. On the other hand, today one of the most important categories is to monitor users' behaviors in access to available resources [2]. The precise choice of access control model and its security on SIEM systems play a key role in the security of these systems [3]. SIEM systems are located alongside the software and monitor all the events happening in them [4, 5]. These systems have access to all of the information in the programs, and in fact, they are a complete repository of all the events that occur in software. Suitable security measures are taken on software [6], but neglecting the security of the SIEM system overwhelms all the security measures of software; this is due to the access of SIEM systems to all events within the software. If the security of these systems is not properly considered, in addition to threatening the event management system, the software and all its information are also threatened. Failure to pay attention to the access control module's threats can cause malicious and irreparable damages to software and SIEM systems [7]. This study proposes an approach not also for creating the SIEM system for software, but also for applying a proper and integrated access control module in these systems based on new standards and access control models [8, 9, 10]. All key points in SIEM that require access to information for generating, storing, analyzing, and monitoring security events are specified and access control is carefully done at all points. All threats to the access control module are identified and solutions are suggested to reduce these threats.

2- Related Works

Here are some recent works on the subject of this paper. Nazir et al. (2016) conducted a study aimed at proposing

high-level language for managing information and security events [11]. In the paper, a Data Specification Language is introduced that simplifies the generation of law for information management systems and security events. Di Sarno et al. (2016) studied the information management systems and security events that solve disparities in security policies [12] and discover the unauthorized network data paths and appropriate configurations for network tools. Granadillo et al. (2016) proposed two new approaches to correlation alert [13]; the previous depends on strategy requirement and safeguard ability models, and the last depends on data security markers. Grambow et al. (2016) [14] provided a background on the existing technical challenges and a practical approach to the Context-mindful Software Engineering Environment Event-driven system (CoSEEEK). This research shows how to use automatic knowledge in creating processes, process compatibility, and environmental process support. In a previous study by the authors of this paper, it was noted that to have a comprehensive AAA model, AAA requirements should be carefully considered through multilayer security policies [15]. For the doctoral dissertation, Grispos (2016) conducted an exploratory case study in an organization to practically address the security events in organizations [16]; therefore, several evaluations of SIEM have been investigated in a case study of an organization. Betz (2016) examined whether information technology by an organization's financial services company can be used to reduce intrusion and security events [17]. B. Mahesh Babu et al. (2015) proposed an advantage the board component that deals with the clients by joining hazard, trust into an entrance control system to build up a more versatile and adaptable avoidance instrument against insider assaults [18]. The paper [26] proposed an autonomous log stockpiling the board convention with a blockchain technique and access control for the IoT climate. The independent model permitted sensors to scramble the logs before sending them to the passage and worker with the goal that the logs were not uncovered to people in general during the correspondence cycle. As per one exemplification, when a SIEM gadget acquired a security occasion, a danger level of the security occasion was determined dependent on at any rate a connection of the security occasion with at least one resource ascribes of an organization that was overseen by the SIEM gadget [27]. The utility of a convenient purchaser gadget was stretched out by permitting account holders the capacity to acquire section into access-controlled scenes utilizing a compact customer gadget that was related with a record that was utilized to buy the affirmation or passes to the occasion at the entrance controlled setting [28]. A summary of the above-discussed related studies is presented in the Table1.

Table 1. Comparison some of the related works

Author/s	**Description**	Advantages	**Disadvantages**
Nazir et al [11]	An undeniable level area explicit language for SIEM (plan, development, and formal check)	Providing a language for managing information and security events	Lack of standard rules to be used in other security systems.
Di Sarno et al [12]	An epic security data and occasion the board framework for improving online protection in a hydroelectric dam	Explaining information management systems and security events where differences in security policies are resolved	Lack of broad test examinations to investigate the adequacy of the SIEM framework in basic foundation applications.
Granadillo et al [13]	Novel Types of Correlation between Alert for Event Management Systems	Proposing correlation between alert approaches. The former is based on models of defense and compulsory policy, and the latter depends on data security markers	PIP and PDP are access control blocks and in this paper, the exact relationship of access control with proposed SIEM is not specified.
Grambow et al [14]	Context-Aware and Process-Centric Knowledge Provisioning: An Example from the Software Development Domain	Providing background on the existing technical challenges and a practical approach to Context-aware Software Engineering Environment Event-driven framework (CoSEEEK)	Lack of software security approach in secure software development
Rezakhani et al [15]	A novel multilayer AAA model for integrated software	Providing a comprehensive approach that defines AAA design for both the operational and executive level and the organizational level	Ignore system logs for more accurate access control
Grispos et al [16]	On the upgrade of information quality in security occurrence reaction examinations	Examining learning security event in organizations	This thesis does not use integrated access control to obtain accurate data.
Betz et al [17]	an investigation of the connection between Security Information innovation improvements and Computer security penetrates and occurrences	Examining whether information technology by an organization's financial services company can be used to reduce intrusion and security events	Low attention to security information and event management in promoting IT security
Hsu et al [26]	An independent log stockpiling the board convention with Blockchain instrument and Access control for the Internet of Things	propose a free log stockpiling the executive's convention with a blockchain technique and access control for the IoT climate	Inability to check log age and log investigation the board Protocol with blockchain mechanism and Access Control for the Internet of Things
Liang [27]	Security information and event management	Figuring a risk level of the security occasion dependent on at any rate a relationship of the security occasion	Lack of integration between SIEM and software and lack of security between them
De Oliveira et al [28]	Occasion access with information field encryption for approval and access control	Strategies permit cardholder verification in a non-installment setting that empowers cardholders to get to an area or a particular occasion.	Encryption is used for confidentiality but it is not used for digital signatures and access control.

3- Proposed Approach

The focus of this research is on creating an integrated and secured access control in security information and event management systems. The proposed module is provided beside the security information and management system along with the software.

The SIEM must have full access to all information and events within the software and analysis event correlation [19]. All access to generated logs must be carefully monitored to prevent information leakage and other security threats. Therefore, integrated access control has a key role in establishing the authority level in the proposed conceptual model [20]. The conceptual model is presented in Figure 1.

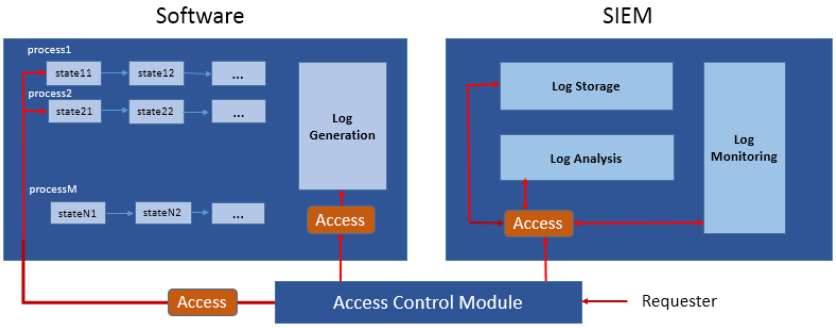


Fig. 1. General Conceptual Model

As shown in the above figure, various components are used to manage the software logs. These components include log generation in the internal structure of software, as well as log storage, log analysis, and log monitor in the SIEM structure. All sections require to store or retrieve logs. Therefore, the access control module is defined as the central point to manage access of all components requesting logs in SIEM. The proposed levels for access management in SIEM is shown in Figure 2.

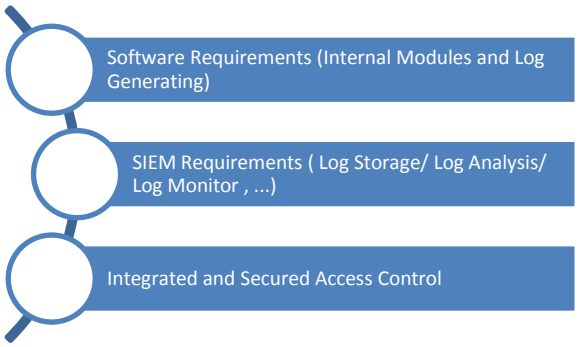


Fig. 2. The proposed levels for access control in SIEM

3-1- Software Requirements

The first step in creating the access control module in SIEM is to make the requirements in software. The software must be created based on software development standards; hence, ISO/IEC 12207 [8, 9, 21]. It is necessary to designing the components for log generation, designing components for securing logs, and also designing the components to logs access. The effective factors in software for creating the SIEM system and access control module are as follows:

I. **Using ISO/IEC 12207:2017 as a deployment guideline:** the ISO/IEC 12207:2017 standard generates software from the perspective of functionality and non-functionality requirements. It even considers the organizational perspective and defines the life cycle for software development. Therefore, this standard is used as a guideline for determining the requirements and the necessary processes and designing a secure architecture for the creation of software.

II. **Designing components for log generation:** After defining the software requirements, the processes to meet the logs generation need to be identified. Accordingly, based on the XES template following the IEEE 1849-2016 standard, the logs' format is determined [22, 23]. The emphasis of this paper is on the generation of those logs that record activities within the business processes. Therefore, logs contain items such as process numbers and activity numbers within processes. After generating the logs, in addition to storing the generated logs in SIEM, they are also stored in software repositories.

III. **Designing components for securing logs:** It is necessary to identify all the general and specific requirements that are needed to secure the logs. One of these requirements is to create and secure log files as well as securing communications to access these files and also syncing logs within the software and SIEMs.

IV. **Designing the components to logs access:** Another requirement to be considered within the software is to

design access control capabilities that will be required during production or use of logs. The reason for designing access control in software is that all events within the software are accessible in the log generation process, and failure to consider the security will result in information leakage. Therefore, it is necessary to create an access control capability for accessing the logs being generated and already generated.

3-2- SIEM Requirements

By doing the above steps in software, the conditions are provided to create the initial requirement in the SIEM system which is capable of storing, analyzing, and monitoring logs. The SIEM system can receive the required logs in the XES format in the IEEE 1849-2016 (IEEE Standard for eXtensible Event Stream (XES) for Achieving correlation of events) [22, 23] and take the appropriate action. The following main components should be considered in designing the SIEM system:

I. **Log Storage:** This block is used to receive logs from the log generation block within the software in the XES format and store them in SIEM.
II. **Log Analysis:** This block is used to analyze the logs stored in SIEM. Determining the correlation between events is one of the most important tasks of this block, which analyzes the logs and extracts the correlations between the events. Event analysis can be used to detect users' suspicious behaviors.
III. **Log Monitoring:** After storing and analyzing events, logs and correlations should be monitored.

The next step for deploying access control module in the SIEM is to provide key points for giving users access permission. In other words, all the key points in which the authority level of users in the SIEM should be investigated are determined. Some of the key points for access control are as follows [24]:

I. **Key points for access control in the log generation block:** Access control in the log generation block is placed on the software side. There are several components in this block, including log generating formatting and log generating parsing. Among these components, the two log-generated storage and log retrieving components are suggested for the users' access control. The access control of SIEM users is performed not only on servers where SIEM is based but also on access to information in software (before sending log information to SIEM) at log generated storing and log retrieving from the log block key points.
II. **Key points for access control in the log storage block:** The log storage block is located on the SIEM system side. This block contains components such as log conversation and log rotation. Among the components of this block, the log storage, log rotation, and log archival key points have access to logs that need to be monitored carefully.
III. **Key points for access control in the log analysis block:** This block is used for event correlation in logs, and contains components such as log retrieving, Rule/policy Definition, Rule/policy editing, Rule/policy deleting, and Reactive Affairs.
IV. **Key points for access control in the log monitoring block:** This block also includes various components in which key access control points include log viewing and alert reporting.

3-3- Integrated and Secured Access Control

The integrated access control module is responsible for issuing access permissions on all software and the SIEM components. The access control module is suggested to use the attribute-based model and architecture described in ISO/IEC 10181. The XACML language is a standard OSSIS language based on the XML and describes security policies. According to this standard, to deal with the requester's request, it is expected to send the solicitation to Policy Enforcement Points (PEP). This unit is given dependents on the reaction to the Policy Decision Point (PDP), as the user interface gives a coherent reaction to the requester. PDP chooses by two fundamental units of Policy Administration Point (PAP) and Policy Information Point (PIP). In PAP every one of the policies is composed by the proprietors and this unit assumes the part of client arrangements knowledgebase. Another unit, Policy Information Point, should exist next to PDP for a dynamic that decides the credits of the requester. PDP gives the authorization of availability dependents on the arrangements in PAP and the requester data in PIP [15]. The access management is to check if the SIEM components are allowed to access the requested information (for storing, retrieving, analyzing, etc.). In an integrated access control module, the applicant within SIEM sends the request for access to the required information to the policy enforcement point. The policy enforcement point inquiries from the policy decision point and decides to issue permission to access or deny access based on the received response.

On the other hand, threat modeling is a way to deal with security examination. This sort of modeling is an organized methodology that empowers planners to recognize, measure, and right the risks. It is suggested to examine the access control module designed for the SIEM system in terms of different threats. The main feature in creating threat modeling will be to reduce potential insider/outsider threats to the integrated access control module. During the explanation of any internal/outsider threat, suggested solutions to decrease them are provided as well.

3-3-1- Insider Threats (Abuses) & Proposed Safeguards

Insider threats (abuses) are the potential threats to the access control module that issue unauthorized access permissions to well-known users. These threats can occur intentionally or unintentionally, but they do have a devastating impact on the accuracy of the access control module. Insider threats (abuses) can have different effects on the SIEM system.

By ISO/IEC 12207:2017 a secure architecture should be created for the access control module in the SIEM system. The proposed architectural design is used to describe the functional requirements as well as non-functionality requirements. The architecture of the secured access control module is presented in Figure 3.

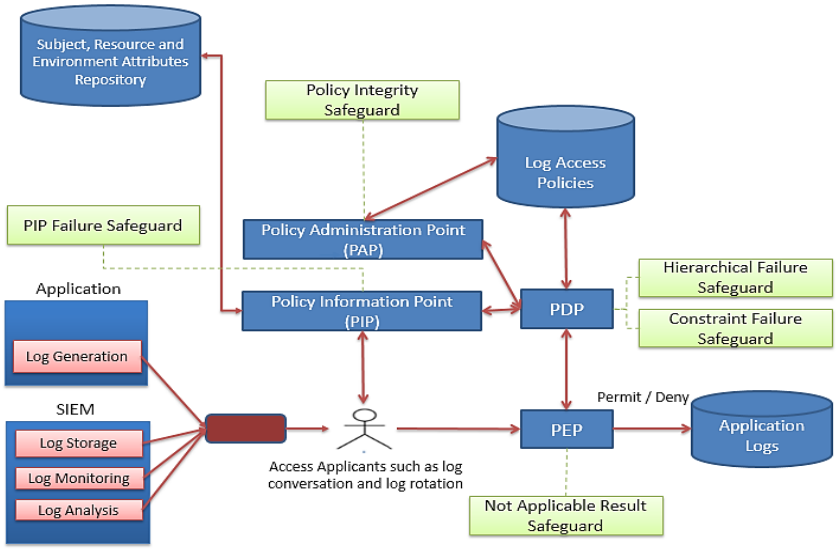


Fig. 3. The Secured architecture of the access control module of SIEM

After examining different insider threats of the access control module, the discovered abuses and proposed safeguards can be accurately categorized as follows:

I. Insider PIP Failure & Safeguards

In process of issuing access permissions, the attributes of the requesting user are compared with the policies in the policyset, and the access permissions are issued to the users at the above listed key points. However, the characteristics of the requester may not be recognized for any reason. In this case, the access control module cannot map the policy to the requestor and therefore the module may make a wrong decision. Among the internal factors that can cause this threat are the tools and technologies used to protect privacy.

The solution to this vulnerability is to use appropriate privacy engineering methods. Given that the proposed access control module is defined in the form of an integrated manner, access control policies and privacy policies are defined accordingly. This will always allow the access control module access to the requesting attributes, and the PIP failure problem will not occur.

II. Policy Integrity Failure & Safeguards

One of the potential threats to the access control module is the lack of a mechanism for policy integrity. Different parts of the organization may use different repositories for access control policies. As a result, policies may interfere by mistake. Internal users just need to exploit this vulnerability in the access control module and create conditions in which their access policies are selected and reviewed by them from their predetermined repository, and therefore, they access unauthorized resources.

In the proposed approach, to fix the problem, policyset, policy, and rule structure are used to define access control policies. The structure of policies is in this way that there are several policysets. Each policyset contains several policies and each policy contains several rules. This kind of definition allows policies to fit into this structure according to different conditions. Applying this structure to define access control policies in a unified and managed way can accommodate all conditions and control accesses.

III. Not Applicable Result & Safeguards

This vulnerability occurs when there is no policy defined in the policy repository for the user's attributes. In this case, the access control module cannot make the right decision. Therefore, it is enough that the internal user attacker provides situations where there will be no policies on his or her attributes, and by most of the access control systems, the access permission will be issued to him.

The suggested solution is to use deny-override and permit-override policies according to the level of trust of users. The deny-override policy means that the access control module will not allow access if no policy is found for the requestor. Also, the permit-override policy means that if the access policy is not found for the requestor, it is granted access. It is recommended that the access control module use the permit-override policy if the requestor trust level is above a certain level and the deny-override policy if the requestor trust level is lower.

IV. Hierarchical Failure & Safeguards

Occasionally due to mistakes made by users in the access control module, restrictions may be occurred and lead to unauthorized permission. Wrong restrictions are an insider threat and abuse can even deny authorized users. For example, the hierarchical failure threat leads to define a series of constraints for an existing user (due to a role inherited from another role) intentionally or unintentionally. In this case, although the user should have access to specific information due to inheritance, the definition of inappropriate constraints will affect his access. The proposed approach to prevent this vulnerability is to develop policies to control roles when defining new constraints. Thus, if inheritance exists between roles, it will not be possible to define any constraint, and only constraints can be defined that do not impair inheritance.

V. Constraint Failure & Safeguards

Different constraints can be defined for access by users of the SIEM system in the access control module. For example, suppose that two roles are inconsistent pair-wisely. This means that these two roles are not attributable to a user. However, the user acquires permissions via a role inconsistent with their current role in any way. In this case, access to the SIEM is unauthorized and causes information leakage. One of the examples leading to unauthorized permission is the inheritance from the roles.

In other words, due to user mistakes in the access control module, the inheritance of the roles may be mistaken and cause an invalid authorization to a role. This off-base inheritance is an insider threat and abuse. For instance, assume an authorized user in the access control module has a user permission definition. Therefore, while defining a new user, the new user inherits an existing role intentionally or unintentionally. In this case, the newly created user will have all of the parent user permissions, and if it is done incorrectly, it can cause information leakage.

To prevent this vulnerability, the definition of new inheritances is controlled over roles. In a way that, if there are constraints between roles, it will not be possible to define inheritances that violate those constraints. In this case, the only inheritances can be defined that do not impair the existing constraints. Because otherwise it may be denied access through the extinct of the constraints, but because of the inherently wrong definition, this allowance is done by error.

3-3-2- Outsider Threats (Misuses) & Proposed Safeguards

Outsider threats (misuse) include those threats that are exerted by outsourced and unauthorized users on the access control module. These types of threats are usually imposed on the system due to vulnerabilities in the access control module. In this section, the outsider threats that enter the access control module of the SIEM system are identified and categorized. These types of threats have a very destructive role in reducing SIEM security.

I. Misuses

Sometimes the failure to identify the characteristics of the requester isn't because of inside reasons however relies upon untouchable causes. One of the elements that can cause untouchable dangers can be DoS attacks on servers that cause their servers to break. The breakage of servers that hold the user characteristics will cause the PDP failure to receive the necessary information to make a proper decision. The XACML language does not have any solution to maintain confidentiality in remote users' communications with the SIEM server. This threat is a misuse and can disrupt SIEM work or disclose information among users. Another type of outsider threat that can be created by individuals outside the organization is the message replay. The attacker person can save the solicitations and replay them at the time of the attacks. In another sort of threat, an outcast attacker sends messages among the legitimate messages that the SIEM clients are sending and causes various problems. In another kind of threat, the attacker modifies messages between SIEM users. This kind of threat is very dangerous because the attacker can change the unauthorized decision into an authorized one. Another kind of outsider threat that can enter the SIEM system is to attack the PDP. In this threat, the attacker tries to send a large number of messages to the PDP. This threat causes a failure in PDP.

II. Safeguards

Some of the problems mentioned above, such as the DoS attacks, are not addressed in this article. However, other problems such as breach of confidentiality, Message replay, Message insertion, and Message modification can be remedied by using software certificates within SIEM and requiring the use of these certificates in communicating SIEM with the access control module. The suggested solution is to use a valid certificate authority to create digital certificates and use them within SIEM. The algorithm of the proposed secured access control module in terms of outsider attacks is as Figure 4.

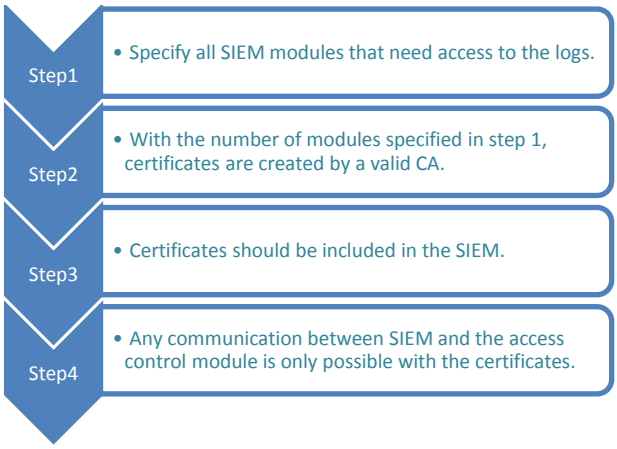


Fig. 4. The algorithm of secured access control in term of outsider attacks

4- Implementation and Evaluation

In this section, the proposed model will be implemented and evaluated. At first, the case study is discussed in detail and the proposed method will be implemented. Then it will be evaluated in terms of insider and outsider threats.

4-1- Implementation

Along with the implementation of the SIEM in this study (including requirements analysis, architectural design, and coding), the specific access control module is designed and produced, which is responsible to control access to receiving and analyzing software events. The SIEM system has various capabilities through which the users defined can access all the events in software and saves, analyzes, and monitors the events. Before implementing the access control module, the implementation conditions must first be specified. In this case, the blocks defined in specific SIEM in the system are presented in Table 2. Also, the number of roles and users applied to use SIEM in the presented software are listed in Table 3.

Table 2. Defining the blocks in SIEM

	SIEM block	Abbreviation
1	Block of Log Generation	LG
2	Block of Log Storage	LS
3	Block of log analysis	LA
4	Block of log monitoring	LM

Table 3. The number of roles and users in SIEM

	Name	Numbers	Example
1	Users	20	User1
2	Roles	9	LA manager, LA user
3	Permissions	21	Read, Write

Table 2 shows the blocks requested by the subjects. For example, log generation blocks are used to create logs in the software. The log storage block is used to store logs and the log analysis block is also used to analyze logs. Table 3 defines the users, roles, and permissions in the implemented access control module. For example, 20 users are defined in the system, and also the roles of the LA manager and LA manager in the access control module are defined.

Step 1: Requirement analysis

By Standard ISO/IEC 12207:2017 (Systems and software engineering – Software life cycle processes), security requirement analysis should be done to deploy an access control module. These requirements are expressed in terms of use cases. In the use case diagram, in addition to the main cases to create the access control module, some issues will reduce the insider threats of the module. The issues raised as the functionality requirements of the access control module in the SIEM system are as follows:

- Policy Enforcement point
- Policy administration point
- Policy Information point
- Policy Administration point

As shown in the previous section, insider threats were suggested during requirement analysis. Now the non-functional requirements of the access control module to reduce the insider threats should be determined as follows:

- PIP Failed safeguard
- Policy administration point Integrity
- Not applicable result safeguard
- Hierarchical Safeguard
- Constraint Safeguard

Step 2: Secured Coding

By performing the above steps, all conditions for the implementation of a secured access control module for the SIEM system are provided. At this point, the access control module is built according to the security requirements expressed in the first step. Insider threats that are entered by the internal authorized users of the access control module into the SIEM system are controlled for the prepared verification checklist.

In addition to the above-mentioned insider threats, the module also controls the outsider threats that come from outside of this module. To reduce outsider threats we are deployed certificates within the SIEM. First, Root CA is installed on the Windows server and we use it as a certificate issuer. Next, the certificates are issued for the components of log generation, log storage, log analysis, and log monitoring. Finally, we embed these certificates into the above components. Therefore, each component associates with the certificate created on a secure platform with the access control module.

4-2- Evaluation

In this section, the proposed method will be evaluated and compared with the recent methods. One of the

problems in evaluating and checking the accuracy of the proposed method is the lack of any standard and dataset in this context. So first, a dataset is created randomly with 10000 records to observing the effective features for evaluation, and then the accuracy parameter will be examined carefully. The created dataset features that we call TMDS are provided in Figure 5.

The TMDS dataset contains several features that play an essential role in granting access. The first features are the requester's name and part of the software from which the access request was issued. Requester location and request time are other fields. The role and level of trust of the requester are other features of the dataset. The business process and the state from which the access is requested are also other features of this dataset. In addition to resource, action, and grant, the hierarchical roles and constraints between roles as well as the policies repository are also the features of the TMDS dataset. We have completely randomized this dataset with 10,000 records. One of the data records shown in Figure 5 means that user1 in the log-storing process, within the organization at 8 am and with the role of LS_LS if the trust level of the user is 0.6, the permission is permitted to write on the log. This is the case if it conflicts with the role of LS_LR and Repository1 is also the repository of policies.

FEATURE NAME	Requester_Name	Requester_Position	Requester_Location	Request_Time	Requester_Role	Requester_Privacy	Process	State	Resource	Action	Grant	Hierarchical_Roles	Constraints_Roles	Policies_Repository
Record0	User1	Log Storing	Enterprise	8:00	LS_LS	0.6	Log Store	Log Storing	Logs	Write	Permit	---	LS_LR	Repository1
	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Fig. 5. The provided dataset features

4-2-1- Insider Threats Reduction

To evaluate the proposed access control module, the accuracy parameter in association with insider threats is validated. Verification and validation checklists have been used to consider some of the insider threats related to the proposed module. This checklist is used to validate the access control module against the PIP failure attack. The verification checklist for this threat is in Table 4.

Table 4. Verification checklist for PIP failure threat

	Verification List	Explanation	Classification
1	ISO/IEC 12207 applies to software production	Application of ISO 12207 standard in software development phases	Discretionary
2	Use privacy engineering policies	Using privacy engineering to respect the privacy	Discretionary
3	Implement ISO 10181 correctly	Use ISO 10181 to grant access permissions	Discretionary
4	Identify access decision-making components	The decision-making components include PDP, PIP, PEP.	Mandatory
5	Communication between software modules is not limited	No restriction on sending information between decision components	Mandatory
6	The PIP module does not restrict the sending of user information to the PDP	Unlimited sending of information from PIP to PDP upon request	Mandatory

We also designed a tool called *Test Tool* to check the accuracy of the access control module against other insider threats. With the designed tool, for each threat, we check the access control module before and after applying the proposed method. In Figures 6 and 7 two parts of this tool are shown to determine the accuracy of the module in Hierarchical Failure and Constraint Failure threats.

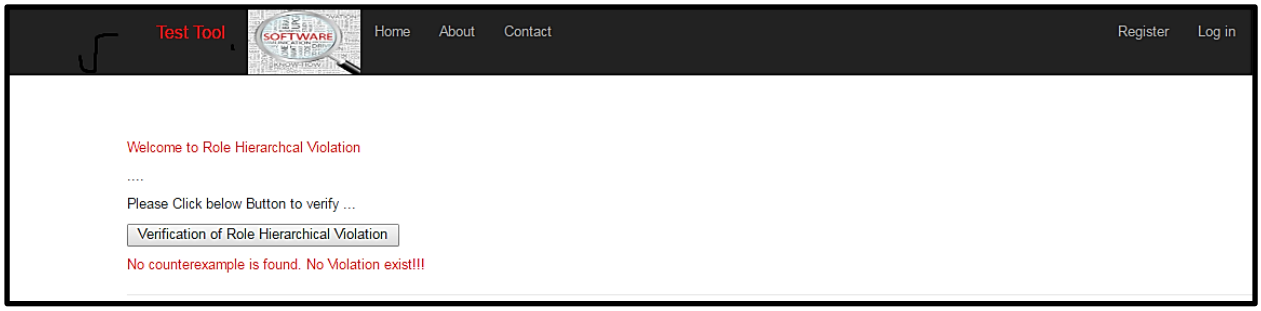


Fig. 6. Role hierarchical verification in the Test tool

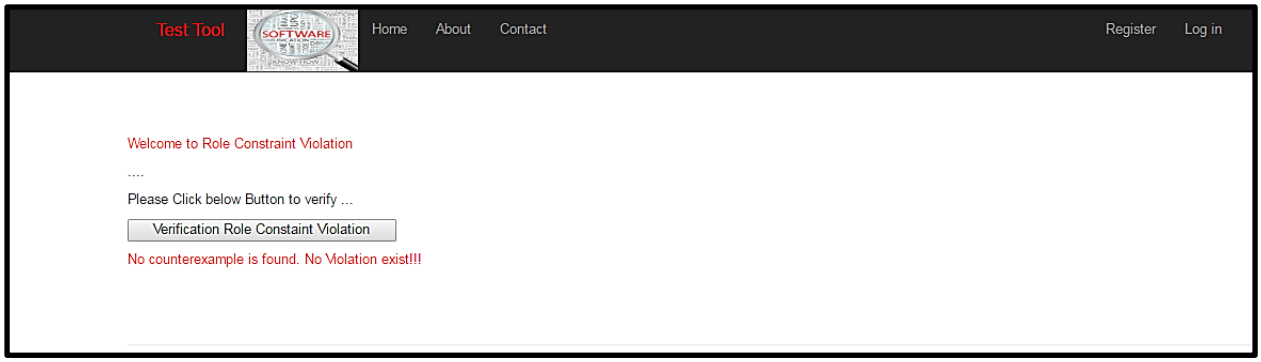


Fig. 7. Role constraint verification in the Test tool

To evaluate the access control module, it is necessary to calculate the confusion matrix parameters. For this purpose, the following values are defined:

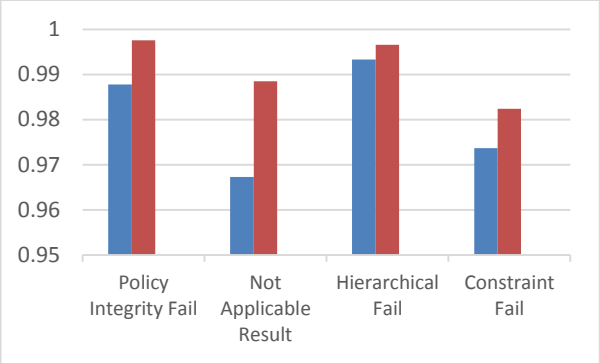
- ✓ TP: The number of records of the TMDS dataset that issue access permissions correctly.
- ✓ TN: The number of records of the TMDS dataset that do not issue access permissions correctly.
- ✓ FP: The number of records of the TMDS dataset that issue access permissions incorrectly.
- ✓ FN: The number of records of the TMDS dataset that do not issue access permissions incorrectly.

According to these values, the parameter of accuracy is calculated in the formula $\text{Accuracy}=\frac{TN+TP}{TN+FN+TP+FP}$. The accuracy parameter is calculated for PIP failed Results, Policy Integrity Failed, Not applicable result, Hierarchical Fail and Constraint Fail threats. According to the information in our TMDS dataset, the TP, TN, FP, and FN parameters are calculated after utilizing a secured access control module for each of the insider threats discussed above, along with the parameters to be evaluated in Table 5.

Table 5. The accuracy calculated for each insider threat

	Policy Integrity Failure	Not Applicable Result	Hierarchical Failure	Constraint Failure
True Positive	5022	5034	5015	4783
True Negative	4954	4966	4951	5041
False Positive	12	116	0	94
False Negative	12	0	34	83
Accuracy	0.9976	0.9885	0.9966	0.9824

As can be seen, accuracy is calculated for each of the insider threats after using the proposed access control module. For each threat, the true positive, true negative, false positive, and false negative are calculated, and then the accuracy is obtained using the defined formula. For example, for the Policy Integrity Failure threat, the accuracy parameter is 0.9976. The accuracy is calculated for the rest of the insider threats as shown in the table above. Also, the accuracy parameter is compared before and after using the proposed method and is shown in Figure 8.



Blue bar: the accuracy before any insider threat reduction
Red bar: the accuracy after any insider threat reduction

Fig. 8. Accuracy before and after of any insider threat reduction

4-2-2- Outsider Threats Reduction

In the previous sections, the access control module is evaluated in terms of insider threats. In this section, the access control module embedded in SIEM is evaluated in terms of outsider threats. The acunetix tool is one of the tools for analyzing and detecting software vulnerabilities [25]. This tool provides alerts based on malicious targets' access paths at four levels:

1. *High:* The highest level of vulnerability, which is very dangerous and allows the hacker to control the software;
2. *Medium:* The medium vulnerability level is less than the previous one, but there is still the ability to control the software;
3. *Low:* It has the lowest risk and ordinary hackers cannot access the software;
4. *Information:* A warning to prevent theft of information

At first, by using the acunetix tool, the vulnerabilities in the access control module are examined. At this step, no action has been taken to utilize the certificates. Then, based on the description of the previous section, the SIEM components communicate via software-defined certificates with the proposed access control module. Finally, the vulnerabilities in the access control module are re-examined by the acunetix tool. The results of the evaluation of threats are provided in Table 6.

Table 6. The number of security alerts for outsider threats

	Informational Risk	Low Risk	Medium Risk	High Risk
Before Proposed Method	5	4	2	5
After Proposed Method	1	1	1	2

5- Conclusion

In this paper, a new method was developed to enhance the security of the access control module in the SIEM system. First, all key points that are accessed by the SIEM within the software are identified and then policies are developed to control precise access. Also, the threats entered into the access control module are carefully detected and then decreased. To assess the proposed method from the perspective of insider threats, the parameter of accuracy was calculated. Also, the number of vulnerabilities was calculated to evaluate the proposed method based on outsider threats. By applying the method proposed in this study, it is possible to enhance the security of the access control module in SIEM systems.

References

[1] D. Godoy and A. Corbellini, "Folksonomy-Based Recommender Systems: A State-of-the-Art Review," Int. J. Intell. Syst., vol. 31, no. 4, pp. 314-346, 2016.
[2] Mohammed, N. M., Niazi, M., Alshayeb, M., & Mahmood, S. (2017). Exploring software security approaches in software development lifecycle: A systematic mapping study. Computer Standards & Interfaces, 50, 107-115.
[3] DURAIRAJ, S. K. J., & Singla, A. (2017). U.S. Patent Software No. 15/303,771.
[4] Detken, K. O., Jahnke, M., Kleiner, C., & Rohde, M. (2017, September). Combining Network Access Control (NAC) and SIEM functionality based on open source. In Proceedings of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Software (IDAACS), Bucharest, September 20th till September 23rd.
[5] Miller, D. R., Harris, S., Harper, A., VanDyke, S., & Blask, C. (2010). Security Information and Event Management (SIEM) Implementation (Network Pro Library). McGraw Hill.
[6] Layton, T. P. (2016). Information Security: Design, implementation, measurement, and compliance. Auerbach Publications.
[7] Piessens, F., & Verbauwhede, I. (2016, March). Software security: Vulnerabilities and countermeasures for two attacker models. In Proceedings of the 2016 Conference on Design, Automation & Test in Europe (pp. 990-999). EDA Consortium.
[8] Shostack, A. (2014). Threat modeling: Designing for security. John Wiley & Sons.
[9] Aydan, U., Yilmaz, M., Clarke, P. M., & O'Connor, R. V. (2017). Teaching ISO/IEC 12207 software lifecycle processes: a serious game approach. Computer Standards & Interfaces, 54, 129-138.
[10] López-Lira Hinojo, F. J. (2014). Agile, CMMI®, RUP®, ISO/ IEC 12207...: is there a method in this madness? ACM SIGSOFT Software Engineering Notes, 39(2), 1-5.
[11] Hu, V. C., Kuhn, D. R., & Ferraiolo, D. F. (2015). Attribute-based access control. Computer, 48(2), 85-88.
[12] Nazir, A., Alam, M., Malik, S. U., Akhunzada, A., Cheema, M. N., Khan, M. K., ... & Khan, A (October 2016). A high-level

domain- specific language for SIEM (design, development, and formal verification). Cluster Computing, 1-15.
[13] Di Sarno, C., Garofalo, A., Matteucci, I., & Vallini, M. (2016). A novel security information and event management system for enhancing cybersecurity in a hydroelectric dam. International Journal of Critical Infrastructure Protection, 13, 39-51.
[14] Granadillo, G. G., El-Barbori, M., & Debar, H. (2016, November). New Types of Alert Correlation for Security Information and Event Management Systems. In New Technologies, Mobility and Security (NTMS), 2016 8th IFIP International Conference on (pp. 1-7). IEEE.
[15] Grambow, G., Oberhauser, R., & Reichert, M. (2016). Context-Aware and Process- Centric Knowledge Provisioning: An Example from the Software Development Domain. Innovations in Knowledge Management (pp. 179-209). Springer Berlin Heidelberg.
[16] Rezakhani, A., Shirazi, H., & Modiri, N. (2018). A novel multilayer AAA model for integrated software. Neural Computing and Software, 29(10), 887-901.
[17] Grispos, G. (2016). On the enhancement of data quality in security incident response investigations (Doctoral dissertation, University of Glasgow).
[18] Betz, L. (2016). An Analysis of the Relationship between Security Information Technology Enhancements and Computer Security Breaches and Incidents. (Doctoral dissertation, Nova Southeastern University).
[19] Babu, B. M., & Bhanu, M. S. (2015). Prevention of insider attacks by integrating behavior analysis with risk-based access control model to protect the cloud. Procedia Computer Science, 54, 157-166.
[20] Bhatt, S., Manadhata, P. K., & Zomlot, L. (2014). The operational role of security information and event management systems. IEEE Security & Privacy, (5), 35-41.
[21] Boucher, P., Wright, M., Cranny, T., Nault, G., & Smith, M. (2015). U.S. Patent No. 9, 197, 668. Washington, DC: U.S. Patent and Trademark Office.
[22] ISO, I. IEC 12207: 2017 Systems and software Engineering-Software life cycle processes., (2017). International Organization for Standardization.
[23] Verbeek, H. M. W., Buijs, J. C., Van Dongen, B. F., & Van Der Aalst, W. M. (2010, June). Xes, xesame, and prom 6. In Forum at the Conference on Advanced Information Systems Engineering (CAiSE) (pp. 60-75). Springer, Berlin, Heidelberg.
[24] IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams, (2016), IEEE Std, pp. 1849-2016.
[25] Kent, K., & Souppaya, M. (2006). Guide to computer security log management: recommendations of the National Institute of Standards and Technology. US Department of Commerce, Technology Administration, National Institute of Standards and Technology.
[26] Erturk, E., & Rajan, A. (2017). Web Vulnerability Scanners: A Case Study. arXiv preprint arXiv:1706.08017.
[27] Hsu, C. L., Chen, W. X., & Le, T. V. (2020). An Autonomous Log Storage Management Protocol with Blockchain Mechanism and Access Control for the Internet of Things. Sensors, 20(22), 6471.
[28] Liang, D. (2020). U.S. Patent No. 10,616,258. Washington, DC: U.S. Patent and Trademark Office.
[29] De Oliveira, M. G., & Jatoba, P. (2020). U.S. Patent No. 10,579,995. Washington, DC: U.S. Patent and Trademark Office.

Leila Rikhtechi received the B.S. degree in Computer Engineering from Azad University, arak Branch, Iran in 1999, and M.S. degree in Software Systems from Azad University, south Tehran Branch, Iran, in 2002. Currently she is Ph.D. Candidate in Arak University, Iran. Her research interests include software security.

Vahid Rafe is an associate professor at Arak University. His research interests are model checking, software testing and search based software engineering.

Afshin Rezakhani received the Ph.D. degree in computer engineering from Malek-Ashtar University of Technology, Tehran, Iran, in 2016. Now, he works as assistant professor in the Ayatollah Boroujerdi University, Boroujerd, Iran. His area research interests include Software Engineering, Information Security Management System (ISMS), IT Governance and Management (Cobit and ITIL), and software security.