Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 01, March 2020.

MODELLING MALWARE PROPAGATION ON THE INTERNET OF THINGS USING AN AGENT-BASED APPROACH ON COMPLEX NETWORKS

Karanja Evanson Mwangi[1], Shedden Masupe[2] and Jeffrey Mandu[3]

(Received: 10-Sep.-2019, Revised: 26-Oct.-2019, Accepted: 10-Nov.-2019)

ABSTRACT

Malware threat is a major hindrance to efficient information exchange on the Internet of Things (IoT). Modelling malware propagation is one of the most imperative applications aimed at understanding mechanisms for protecting the Internet of Things environment. Internet of Things can be realized using agent-based modelling over complex networks. In this paper, a malware propagation model using agent-based approach and deep-reinforcement learning on scale free network in IoT (SFIoT) is assiduously detailed. The proposed model is named based on transition states as Susceptible-Infected-Immuned-Recovered-Removed (SIIRR) that represents the states of nodes on large-scale complex networks. The reliability of each node is investigated using the Mean Time To Failure (MTTF). The factors considered for MTTF computations are: degree of a node, node mobility rate, node transmission rate and distance between two nodes computed using Euclidean distance. The results illustrate that the model is comparable to previous models on effects of malware propagation in terms of average energy consumption, average infections at time (t), node mobility and propagation speed.

KEYWORDS

Internet of things, Agent-based modelling and simulation, Modelling malware propagation, Large-scale-free networks, Deep-reinforcement learning.

1. INTRODUCTION

Today, any device connected to communication systems may be subject to unscrupulous and malicious individuals, whose main purpose is to access sensitive information. To achieve their goals, they use different specimens of malware [1]. This malware often goes unnoticed for a period long enough to study the behavior of the internal network and its elements, in order to extract valuable information. Considering that there are large numbers of nodes deployed on communication systems and, in many cases, they are usually deployed on hostile unattended environments without human supervision, they become a principal target for malware attacks [2]. Agent-based modelling and simulation (ABMS) is an effective way to model and analyze complex networks [3]. Network consists of agents and the activities of these agents are monitored concurrently. ABMS offers the set of transition rules with consideration to individual device characteristics thus appropriate for malware modelling, where individual device variability is a key consideration [4]-[5]. This paper postulates the malware propagation process on scale-free networks by proposing agent-based model and simulation. In scale-free networks, nodes are added with maximum probability node. Agent-based modelling and simulation are instigated for modelling the dynamics of malware propagation scale-free networks. The diversity of nodes in scale-free network by varying parameters, such as node mobility, energy consumption and propagation speed that affect the malware spread in the network. The proposed model is further compared with analytical results obtained from previous agent-based modelling and simulation schemes [6]–[9]. The major contributions of this paper are outlined as follows:

1) Creation of an agent-based model and simulation with a decision maker for modelling the malware propagation on large networks using a deep-reinforcement learning algorithm.
2) The node state transition model Susceptible-Infected-Immuned-Recovered-Removed (SIIRR) is developed and the individual node performance measurement is estimated for computing the node reliability using mean-time-to-failure metric.

[1] K. E. Mwangi is with Faculty of Engineering, University of Botswana Gaborone, Botswana. Email: sun-dayfeb29@gmail.com

[2] S. Masupe is with BITRI, Botswana. Email: smasupe@bitri.co.bw

[3] J. Mandu is with Department of Electrical Engineering, University of Botswana Gaborone, Botswana. Email: jef-freym@mopipi.ub.bw

The rest of the paper is structured as follows: The related literature on malware propagation is explored in section 2. In section 3, the proposed model is presented and the succinct details on the application of deep-reinforcement learning in modelling malware propagation are given. The experimental set-up and simulation of the proposed scheme are discussed in section 4. Analysis is performed to compute the metrics, such as average energy consumption, average infections over time, node mobility and propagation speed. The simulation results are validated and compared with analytical results obtained from previous agent-based modelling and simulation schemes. Finally, the conclusion of this paper and future research directions are given in section 5.

2. RELATED LITERATURE

The rise in use of IoT devices to launch malware attacks in the recent past has invoked researchers' interest in understanding IoT malware propagation and control. In this section, we review recent literature in malware propagation with a bias towards agent-based modelling which is the approach taken in the our proposed model.

A Markov Random Filed (MRF)-based spatio-stochastic framework is applied in complex communication networks, where malicious threats spread through direct interactions and follow the SI state model proposed by Karyotis [8]. It also combines Gibbs sampling with simulated annealing to analyze the behaviour of the systems under various topological and malware-related metrics. The disadvantage of MRF is that it is not isotropic, since it varies in magnitude according to the direction of measurement. Besides, the reliability of individual nodes is not assessed. The rumor diffusion process is proposed to model the outbreak of malware in [7]. The limitation of this agent-based analytical model is that it is difficult to prove the validity of the malware-free equilibrium stability (global and local).

In [10], the four aspects of malware propagation modelled were; user mobility, application-level interactions among users, local network structure and network coordination of malware (Botnets). The model was tested for a malicious virus like Cabir spreading among the cellular network subscribers using Bluetooth. A queuing-based malware propagation modelling approach was proposed in complex networks with churn [11]. Churn refers to dynamic node variation which captures the dynamics of SIS-type malware in time- varying networks. It quantified network reliability and improved the robustness of the network against some generic malware attacks. With the dynamic nature of node variation, it does not consume less energy and also the spreading speed is high. Malware propagation over wireless sensor networks has been proposed in [12], where the network topologies are based on complete or regular graphs. The first disadvantage of this network model is that it does not consider the individual characteristics of sensor nodes which form an important attribute in modelling heterogeneity of nodes and the second disadvantage involved in this model is that parameters such as transmission rate and recovery rate are not explicitly defined.

Batool et al. [9] demonstrates that Internet of Things networks can be modelled using a hybrid approach of using complex network and agent-based models. The construction of IoT elaborated models addressing the emergence and individual characteristics represent an existing research challenge. To model the IoT as a scale-free network, when a new node wants to join the network, it requires the degree and distance of all nodes (centrality measures) in the whole network in order to compute the probability of connecting to each existing node. The centrality measure is a critical measure of how central the node is to communication and connectivity. Betweenness and closeness centralities are calculated in each subnet. Betweenness centrality of a node is the probability for the shortest path between two randomly selected nodes to go through that node and is calculated as:

$$C_B(i) = \frac{1}{(n-1)(n-2)} \sum_{j \neq i, k \neq i, j \neq k} \frac{N_{sp}(j \overset{i}{\rightarrow} k)}{N_{sp}(j \rightarrow k)} \tag{1}$$

where, $Nsp(j \rightarrow k)$ is the number of shortest paths from node i to node k and $Nsp(j \xrightarrow{i} k)$ is the number of shortest paths from j to node k that pass through i.

Closeness centrality is a measure of how accessible a node is from other nodes and is calculated as:

$$C_c(i) = (\frac{\sum_j d(i \rightarrow j)}{n-1})^{-1} \tag{2}$$

which is an inverse of the average distance from node *i* to all other nodes. If *Cc*(*i*) =1, then you can reach each other node in the network *via* one step. The centrality measures are key to determine the influence of malware propagative and spreading nodes.

The inherent weakness of the deterministic and stochastic models surveyed in our previous work in literature is the full mix assumption [13]. The full mix assumption holds that every node has equal chances of coming into contact with others in the network, which is not necessarily the case in malware propagation on IoT networks where heterogeneity is a key factor. The introduction of the decision maker in the model overcomes the key challenge of arriving at an infection decision based on individual node interaction and individual node parameters, not just contact.

3. The Proposed Model

In this section, a model is formulated to model malware propagation over large-scale-free communication networks. A scale-free network environment for heterogeneous IoT devices is visually illustrated in Figure 1.

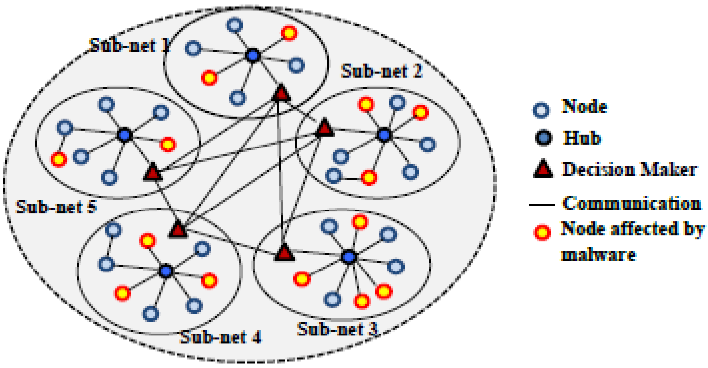


Figure 1. Scale-free Internet of Things networks.

The notion for modelling of malware propagation on large-scale-free networks is as follows; mitigate effects of malware over large-scale-free IoT networks, set flexible simulation parameters (number of nodes/devices are high and transmission range is also high), reduce the malware propagation speed in SFIoT networks and analyze regular changes in the subnets due to the node mobility rate between subnets within a time-varying environment. We consider a network as a graph with *N* nodes and *M* edges. The total population of *N* nodes is divided into *T* subnets, with n_i nodes where i=1,2,. . . , m nodes. The total population of nodes is given by Equation 3:

$$\sum_{i=1}^{m} n_i = N \tag{3}$$

For each subnet *T*, the probability P_i is used to add a link between two nodes that should satisfy Equation 4:

$$\sum_{i=1}^{m} n_i P_i . \frac{1}{2} n_i (n_i - 1) = \frac{N(K)}{2} \tag{4}$$

where, K denotes average degree of nodes in the entire network. When a new node is announced to the network to be attached to N nodes with high degree K, the announcement of the new node and preferential attachment continue until a network with !=t+N has been deployed. The principle of the decision maker-based model of malware propagation on sub-netting-based scale-free networks is based on the SIIRR model states. Decision maker is denoted as an agent considered for modelling malware propagation. Each node in the network has defined heterogeneity behaviour and set of rules is used for modelling the node behaviour. While modelling the malware propagation, nodes are classified into five states. In each time stamp, a node transits to one of the five possible states as listed below.The state transition diagram for SIIRR model is depicted in Figure 2.

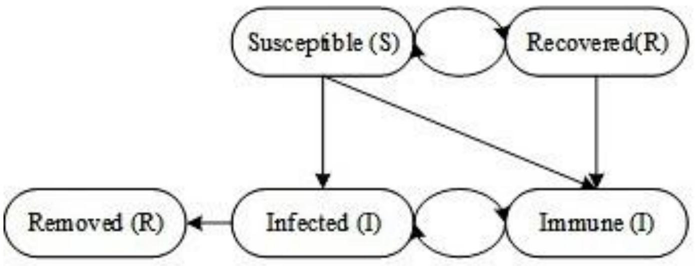


Figure 2. SIIRR model transition diagram.

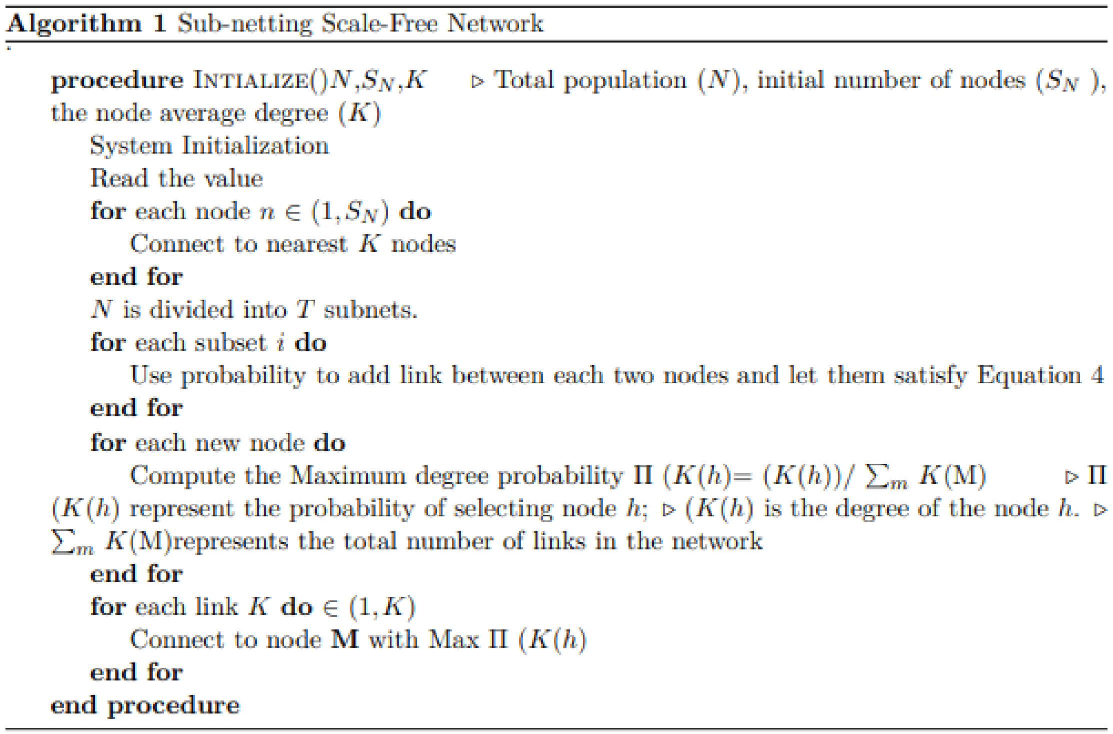
1) Susceptible (S): It is the first state of a node or hub and it often refers to infected in future.

2) Infected (I): The node attracted by malware is called the infected node. In this state, a node propagates the malware infections to all their neighbours.

3) Immune (I): The node that is unable to become infected by any node is called immune. This type of nodes has an immunization scheme, such as an anti-malware solution, to detect and block malware.

4) Recovered (R): It refers to infection removed state and does not get infected again.

5) Removed (R): The node or hub is attracted by the malware and can spread malware at time, *t*.



```
Algorithm 1 Sub-netting Scale-Free Network
.
    procedure INTIALIZE()$N$,$S_N$,$K$    ▷ Total population ($N$), initial number of nodes ($S_N$ ),
    the node average degree ($K$)
        System Initialization
        Read the value
        for each node $n \in (1, S_N)$ do
            Connect to nearest $K$ nodes
        end for
        $N$ is divided into $T$ subnets.
        for each subset $i$ do
            Use probability to add link between each two nodes and let them satisfy Equation 4
        end for
        for each new node do
            Compute the Maximum degree probability $\Pi$ ($K(h)$= ($K(h)$)/ $\sum_m K$(M)        ▷ $\Pi$
    ($K(h)$ represent the probability of selecting node $h$; ▷ ($K(h)$ is the degree of the node $h$. ▷
    $\sum_m K$(M)represents the total number of links in the network
        end for
        for each link $K$ do $\in (1, K)$
            Connect to node M with Max $\Pi$ ($K(h)$
        end for
    end procedure
```

The flowchart in Figure 3 shows the steps in model formulation. Algorithm 1 shows the detailed procedure for sub-netting-based network construction.

3.1 Modelling Deep-reinforcement Learning in Malware Propagation

A Deep-reinforcement Learning (DRL) scheme is adopted to illustrate the variables used for a Continuous Markov Chain Model (CMCM). The main goal of the CMCM in a DRL problem is to increase the obtained rewards. The tuples of DRL are as follows:

$$T = S, A, R, E, H, \gamma \tag{5}$$

where, S denotes the set of states, A is a possible set of actions, E is the environment, R is the reward function for state and action. In DRL, the agent has the ability to act where each action influences its

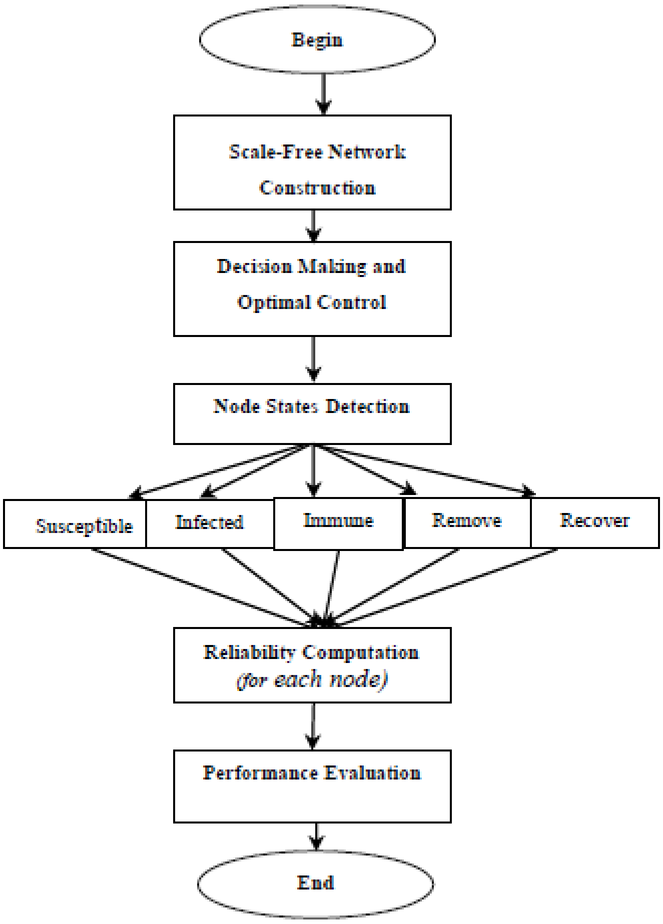

Begin
Scale-Free Network Construction
Decision Making and Optimal Control
Node States Detection
Susceptible
Infected
Immune
Remove
Recover
Reliability Computation (for each node)
Performance Evaluation
End

Figure 3. Scale-free Internet of Things (SFIoT) malware propagation model.

future state of the agent and success can be estimated using scalar reward signal. Q-learning-based reinforcement learning algorithm solves the decision making problems. Q-learning is defined as the quality of action in given state S at time t.

Environment (E): The environment is the area in which agents communicates with each other.
Agents (A): In a given environment, an agent receives information and performs the corresponding action. The main goal of agents is to pick the best policy that increases the total reward.
States (S): This is the condition defined by agent characteristics within the defined transitions.
Actions (A): A state transition from one state S_t to another state $S_{(t+1)}$at time t+1 is called action.
Reward (R): It represents the closeness of the current state to the true class. It is formulated by Equation 6.

$$R\left(S_t A_t S_{(t+1)}\ Y\right) = C\ \left(S_{(t+1)}, Y\right) \tag{6}$$

Rewards depend on the current state and the action performed.
Discount factor (γ): The discount factor controls the importance of future rewards ($\gamma \in [0,1]$).
State transition distribution: It is the transition probability that action A in state S at time t will lead to state S^t at time $t+1$: $PA(S,S^t) = PR(S^t \mid S,A)$. The policy ($\pi$) where ($\pi$)= At and the policy for a state is denoted $(\pi)(S) \longrightarrow A$ which changes with the reward policy as:

$$\Re_t = \sum_{t=0} \gamma^t\ R_t\, \gamma \tag{7}$$

where $0 \leq \gamma < 1$.

In the Q-learning approach, an approximate reinforcement machine learning algorithm is presented for IoT devices. Consider the Q-value updated equation as formulated in Equation 8.

$$Q(S_{t+1}, A_{t+1}) \Leftarrow (1-\alpha) Q(S_t, A_t) + \ \alpha\ [\Re\ (S_t, A_t) + \gamma\ \max_{a'} Q(P(S_t, A_t), a')] \tag{8}$$

where, $Q(S_t,A_t)$ is the Q-value of current state S_t when action A_t is selected at time t, α is the learning rate, γ is the discount factor, where γ is set between 0 and 1, $\max_{a'} Q(P(S_t, A_t), a')$ is the maximum possible Q-value in the next state $S(+1)$ if selects possible action a'. $\Re\ (S_t, A_t)$ denotes the reward function when state St selects At.

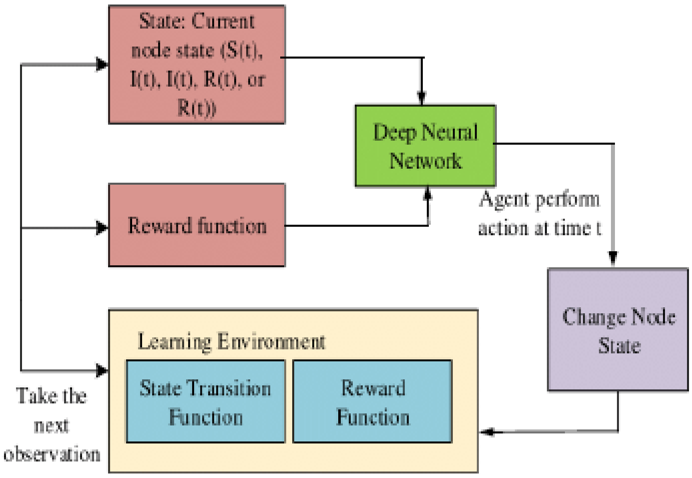

State: Current node state (S(t), I(t), I(t), R(t), or R(t))
Deep Neural Network
Reward function
Agent perform action at time t
Change Node State
Learning Environment
State Transition Function
Reward Function
Take the next observation

Figure 4. Deep-reinforcement learning.

Figure 4 visually illustrates the deep-reinforcement learning approach adopted in the model. The Q-learning model is used to classify the nodes as part of five possible transition states. It specifies transition of nodes between states from $S \rightarrow I$, $I \rightarrow R$ and $R \rightarrow S$, where the recovered state and removed state are terminal. The nodes do not transition to another state after being in the removed state or the recovered state. It is represented as the SIIRR model and mathematically formulated as:

$$\frac{dS(t)}{dt} = -\sigma \frac{S(t)\,I(t)}{N} \tag{9}$$

$$\frac{dI(t)}{dt} = -\sigma \frac{S(t)\,I(t)}{N} - \alpha\, I(t) \tag{10}$$

$$\frac{dI(t)}{dt} = \beta\, I(t) \tag{11}$$

$$\frac{dR(t)}{dt} = \alpha\, I(t) \tag{12}$$

$$\frac{dR(t)}{dt} = \sigma\, I(t) \tag{13}$$

where, _ is the infection rate *S* ! *I*, _ is the recovery rate *I* ! *R*, _ is the removed rate. The total population *N* (network size) at time *t* is computed as:

$$N(t) = S(t) + I(t) + I(t) + R(t) + (R(t) \tag{14}$$

After the scale-free network formation, all the hubs, decision makers and ordinary nodes are set to susceptible state. At time slot t = 1, one or more nodes are set into infected state and each time slot t = 2, 3 or 4 . . . n, malware propagates from infected nodes to their adjacent nodes through communication links. The node state changes continuously at each time slot.

3.1.1 Reliability Computation

The reliability function for a node is computed by using Mean Time To Failure (MTTF). However, most of the previous schemes in malware modelling have not considered the reliability factor. Specifically, reliability is the probability that the system will perform its intended function according to the specified design. To improve the network performance, we consider several metrics for computing the reliability. These are; node degree, node mobility rate, node transmission rate and distance between two nodes.

Node degree is the number of links (in degree and out degree) that lead into or out of the node. For each sub-net, the mobility of the node (i) is computed as follows:

$$M\,(i) = \sum_{i=1}^{n} \frac{NCP - NOP}{Mobility\ Speed} \tag{15}$$

where NCP is the Node Current Position and NOP is the Node Origin Position. A transmission rate (in Kbps) between two nodes depends on the message size (D_s) and distance between two nodes (DN) given as:

$$T\,R(i) = \ C_1 \times \ D_S + \ C_2\,DN \tag{16}$$

where C_1 and C_2 are constant variables. The distance between two nodes is computed using the Euclidean distance metric, which is calculated as:

$$d(a,b)^2 = \ (b_1 - a_1)^2 + (b_2 - a_2)^2 \tag{17}$$

The reliability of a node $R(N(t))$ is the probability that the node will be successful in the interval between time 0 and t as shown in Equation 18:

$$R\big(N(t)\big) = P(r > t) \quad t \geq 0 \tag{18}$$

In Equation 18, r is a random variable that denotes the time-to-failure or failure time. The mean time to failure is computed by Equation 19:

$$MTTF = \int_0^{\infty} t\,f(t)dt, Then\ f(t) = -\frac{dx}{dt}[R(t)] \tag{19}$$

Performing integration operation yields;

$$MTTF = \int_0^{\infty} td\,[R\,(N(t))] = \int_0^{\infty} t\,[R\,(N(t))]\ + \int_0^{\infty} R\,\big(N(t)\big)dt \tag{20}$$

In Equation 20, t(R(N (t) → 0 and x → ∞. It yields the second term, which equals:

$$MTTF = \int_0^{\infty} R\,\big(N(t)\big)\,dt \tag{21}$$

For each sub-net in a scale-free network, the reliability of a sub-net at time t can be computed by:

$$R\big(S(t)\big) = 1 - \prod\nolimits_{sub-net\in path} (1 - R\big(N(t)\big) \tag{22}$$

Moreover, any path composed of sub-nets in a scale-free network R(P(t)) at time t can be computed as:

$$R\big(P(t)\big) = \prod\nolimits_{sub-net\in path} (1 - R\big(N(t)\big) \tag{23}$$

As a result, a sub-net-based scale-free network consists of reliable paths. Hence, the reliability of the network (R(t)) is computed by;

$$R(t) = 1 - \prod\nolimits_{path} (1 - R\big(P(t)\big) \tag{24}$$

The topology of a scale-free network is constructed based on the actual parameters (node degree and maximum probability of a node) in a sub-net. The proposed scheme is implemented in the field of Internet of Things. The reliability for each node in the scale-free network is under malware propagation situation.

4. Simulation

In this section, the modelled propagation algorithm is simulated. The proposed scheme was compared to analytical results obtained from published works as follows: for energy consumption, to the work of Batool et al. [9]; for average infection rate, to the works of [6], [7]-[14]; for propagation speed and node mobility to the work of [8] based on the performance metrics described in sub-section 4.2.

4.1 Experimental Set-up

The model is implemented using NS-3 (version 3.26) for simulation. NS-3 is a network simulator which is mainly supported for Linux and written using C++. But, the binding of NS-3 is written in Python. In our experiment, the Gaussian Markov (GM) mobility model is used. Gauss-Markov (GM) mobility model is used to simulate mobility of device agents. Gauss-Markov mobility model caters for temporal dependency; i.e., it has a memory to correlate previous states. In Gauss-Markov, the velocity of the device is modeled as a Gauss- Markov stochastic process, as it is assumed to be correlated over time. In this model, node speed and direction are considered with respect to time, taking into account the previous speed s_{n-1}, previous direction d_{n-1}, the mean speed $\bar{s}$ and direction $\bar{d}$. The randomness parameter α has a Gaussian distribution. Current speed and direction are given by:

$$s_t = \alpha\, s_{t-1} + (1-\alpha)\,\bar{s} + \sqrt{(1-\alpha^2)s_{xn-1}}$$

$$d_t = \alpha\, d_{t-1} + (1-\alpha)\,\bar{d} + \sqrt{(1-\alpha^2)ds_{xn-1}} \qquad (25)$$

where, s_{xn-1} and d_{xn-1} are random variables from a Gaussian distribution. The simulation of the proposed scheme uses 200 node moves in a 5000 m × 5000 m rectangular region for 100 seconds of simulation. These nodes are vehicles deployed along the road perimeters and 20 sensors are used for sensing information.

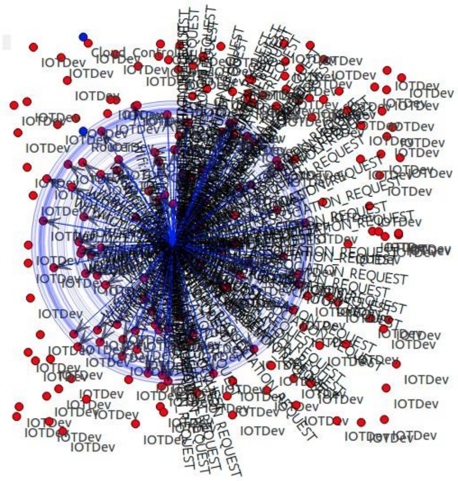


Figure 5. Scale-free network formation visualization.

Four traffic lights for each road lane entering the intersection are considered. The blue circle in the upper right section represents the decision maker entity that manages the traffic light timing. Assume that each node moves independently with the same average speed. All nodes in the network have the same transmission range of 250 m. The simulated traffic is of a Constant Bit Rate (CBR). The proposed scheme is implemented in a single intersection-based road traffic system, then the sub-net construction process is performed. The process is based on the node residual energy and degree of nearest node. In each sub-net, decision maker is selected. All nodes are connected into hub. If the node is not connected

Table 1. Simulation settings and parameters.

Simulation parameters	**Values**
Network simulator	NS-3.26
Area size	5000 m×5000 m
No.of nodes	200
Communication range	250m
Simulation time	100 seconds
Packet size	1024 bytes
Mobility model	Gauss-Markov Model
Node speed	2, 4, 6, 8 and 10 m/s
Pause time	5 seconds
No. of runs	100
No. of packets	100 packets /simulation

to hub, the route between the node and hub is found using FIFO rule. Next, the node sense data and decision maker classify the node state as susceptible, infected, immune, recovered and removed using Deep-Reinforcement Learning (DRL). A visualization showing the formation of a scale-free network can be seen in Figure 5. The simulation settings and parameters are summarized in Table 1.

4.2 Simulation Performance Metrics

The proposed scheme is evaluated for performance based on the following metrics:

1) Energy consumption: It is the rate of energy used for packet transmission. Energy conservation is an important issue while communicating with other nodes.
2) Average infection rate: It is the number of nodes found to be infected during packet transmission.
3) Propagation speed: It can be computed by finding the number of infected nodes at time *t* and is based on the threshold value for different states.
4) Node mobility: It has long been recognized as an efficient metric for modeling malware propagation in Internet of Things; e.g. road traffic systems and smart office application systems. It causes major issues, such as increased energy consumption and connectivity failure. Hence, it needs to be considered in complex networks, so that it brings benefits of reduced energy consumption and reduced spread of malware over communication networks.

4.3 Comparative Analysis

The statistical analysis of the obtained simulation raw data is carried out. Average (means) and the confidence intervals are calculated. The confidence interval of the data realized from the simulation is calculated as follows. Simulations $x1,x2,...,x5$ are carried out for each set of network size in the simulation. Since the number of sample simulations is less than 30, that is *n* = 5, the *t* distribution with n-1 degrees of freedom is adopted as the statistical test. In order for the the *t* distribution to be applied, the data needs to follow normal distribution. The test for normality is carried out to provide evidence that the simulation data is normally distributed. The normal probability plots are used to depict the outcome of the normality test. Shapiro-Wilk normality measure is also applied, since simulation instances are less than 2000. Shapiro-Wilk test is carried out at all network sizes. The confidence interval is given as [L, U], where L is the lower bound and U is the upper bound of the interval. This can be expressed as [L, U] = [average – margin of error, average + margin of error]. The confidence interval is calculated as:

$$[L,U] = \left[\bar{x} - t_c\frac{s}{\sqrt{n}}, \bar{x} + t_c\frac{s}{\sqrt{n}}\right] \tag{26}$$

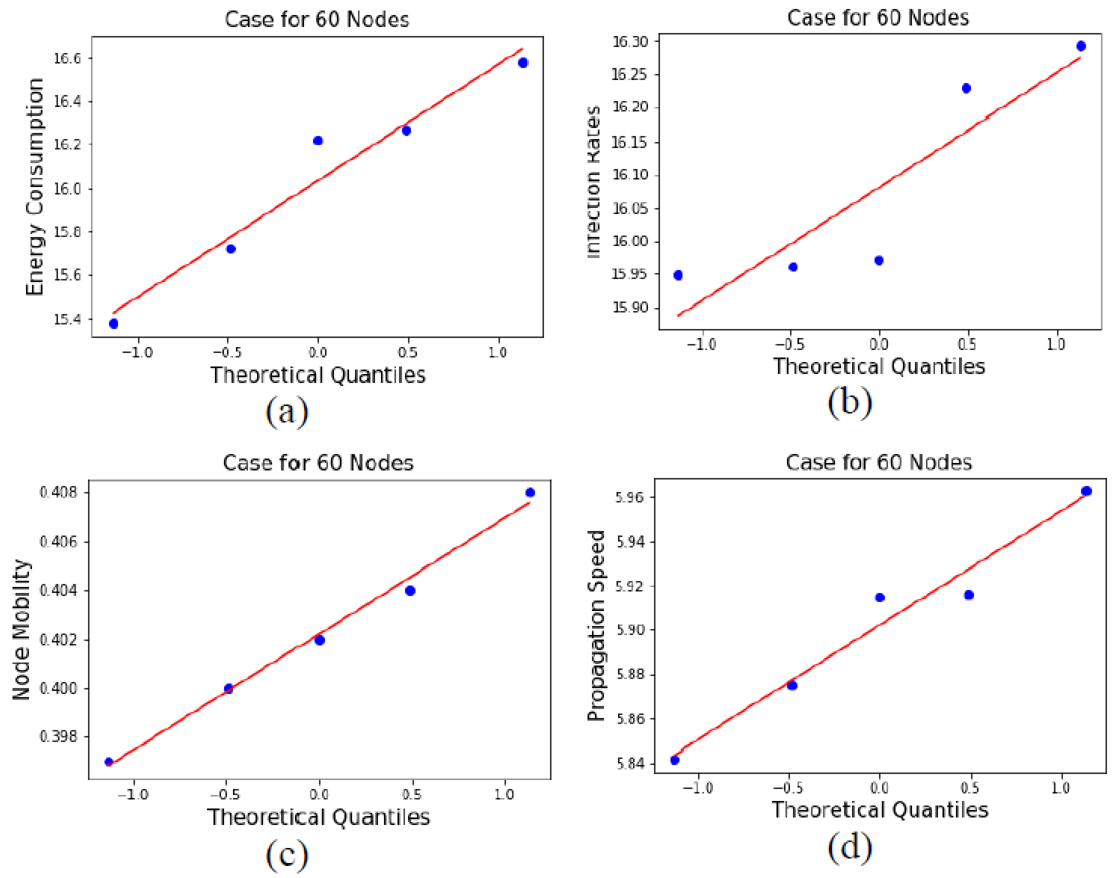
where, t_C is the critical value from the *t* distribution depending on the confidence level. The confidence level of 95% is used in this study.

The simulation results are subject to the test of normality for each of the network sizes and parameters. Shapiro-Wilk test statistics and the normal probability plots are derived for each of the network sizes and parameters. The normal probability plot is a visual illustration showing whether the data fits a normal probability distribution. The simulation raw results are plotted against the theoretical quantiles. If the data lies along the straight, that data fits the normal probability distribution. The test proved that the results on all network sizes were normally distributed as required for the use of Student *t* distribution in the calculation of confidence interval. For illustration purposes, the example of the normal probability plots for network size of 60 nodes is shown here. Figure 6 shows the normal probability plots for a network of 60 nodes.

Shapiro-Wilk test statistics are calculated based on the following hypotheses:

H0: The population is normally distributed.
H1: The population is not normally distributed.

If the significance level Sig. = α > 0.05, we can't reject H0, thus the population is normally distributed. Shapiro-Wilk test statistics indicate that all the data from the simulations is normally distributed at 95% confidence interval.For example, the network size of 60 nodes shown in Figure 6 yielded Shapiro-Wilk test statistics and confidence levels as shown in Table 2.



(a) (b) (c) (d)

Figure 6. Normal probability plots for network size = 60 nodes for (a) Energy consumption (b) Infection rates (c) Node mobility and (d) Propagation speed.

From Table 2, the significance level **Sig.** = $\alpha > 0.05$ satisfies $H0$ and the data is normally distributed. The 95% confidence level upper and lower bounds are also calculated.

Table 2. Test on network size of 60 nodes.

	Shapiro-Wilk test significant levels	Mean Difference	95% confidence level of the difference	
	If (Sig.>0.05), Accept H0		Upper (U)	Lower (L)
Propagation	0.871	5.902200	5.8525	5.95915
Energy	0.706	16.034200	15.44065	16.62775
Mobility	0.995	0.40220	0.39705	0.40735
Infection Rate	0.55	16.080600	1544065	16.2865

4.3.1 Energy Consumption

First, we examine the energy consumption for our proposed scheme and then compare with the previous scheme. Energy consumption is the practice of quantity of energy used. It can be achieved through efficient energy use over complex communication environment. The tasks that are considered for energy consumption include: sensing, transmission and communication. The total energy consumption was estimated in milli joule (mJ). It is formulated as follows:

$$E_c = E_T + E_R + E_I \qquad (27)$$

Energy consumption for transmission, E_T is computed by:

$$E_T = (\alpha_1 + \alpha_2 D^{\sigma})m \qquad (28)$$

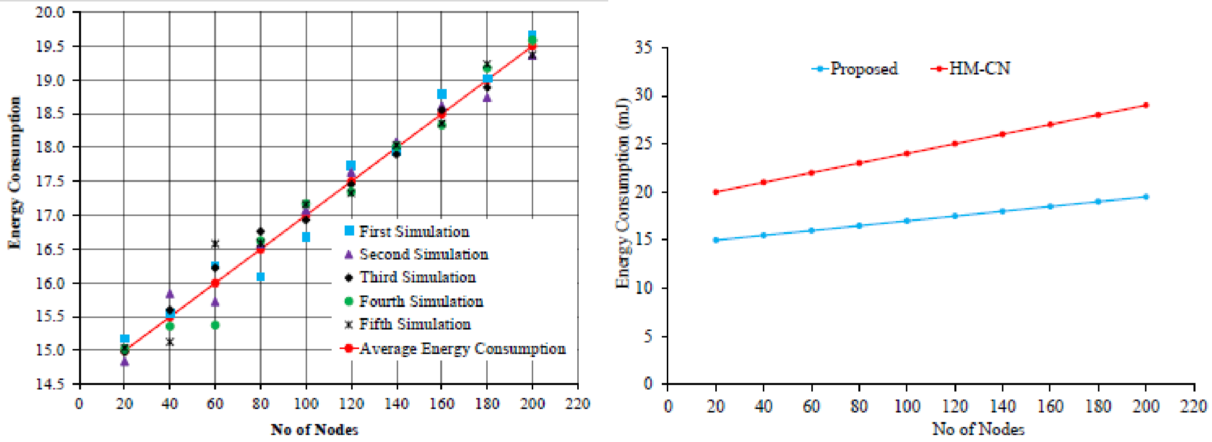
Energy consumption for reception E_R is computed by:

$$E_R = (\alpha_3)\, m \tag{29}$$

Energy consumption for idle state E_I is computed by:

$$E_I = \alpha_4 t_I\, P_m \tag{30}$$

In Equations from (27) to (30), D is the transmission distance, m is the packet length, α_1 - α_4 are the system dependent parameters, t_I is the idle time and P_m is the packet processing rate of the node. Five simulations were carried out for each network size and energy consumption measurements were noted for each run. Figure 7 shows the average energy consumption comparative analysis. Sub-Figure 7(a) shows the energy consumption rates at varied network sizes on the proposed scheme and in Sub-Figure 7(b), the average rate of energy consumption for the proposed scheme and that of HM-CN [6] are compared.



((a)) Simulated Energy Consumption.

((b)) Average Energy for the proposed model Vs HM-CN [6].

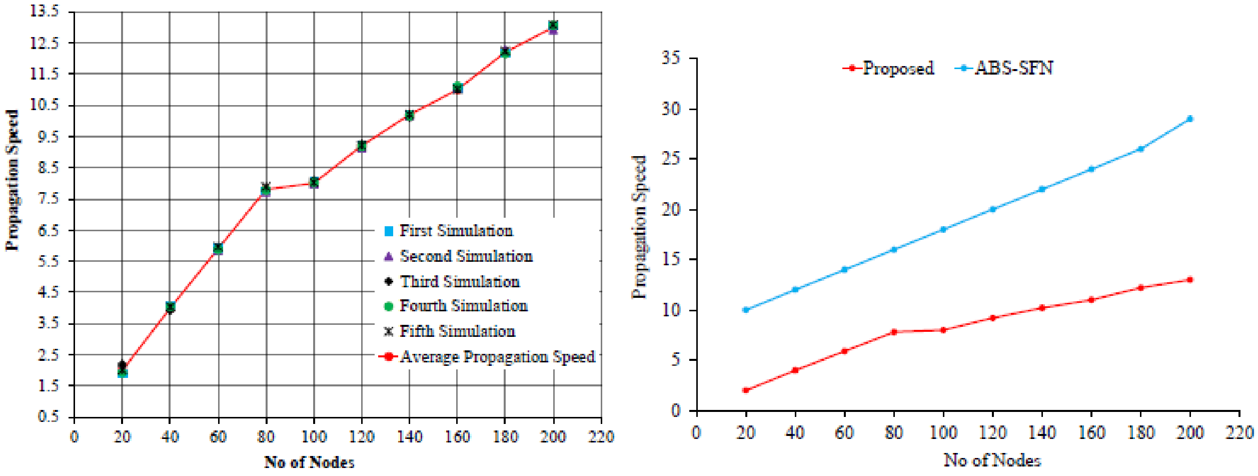
Figure 7. Energy consumption analysis.

Previous work; namely, HM-CN [6], noted that sensing and communication are the most energy-consuming tasks. Transmission and reception cost is high, especially for short-range communication. These drawbacks are solved and our proposed scheme provides a realistic estimation of energy consumption in networks. The proposed scheme is simulated for N=200 nodes (nodes varying as 20, 40,...,200). The decision maker isolated malware-infected nodes which are not allowed for communication and sensing. Furthermore, we follow FIFO rule for packet transmission. Hence, we obtained minimum energy consumption.

4.3.2 Average Infection Rate

Infection rate is an important parameter in modelling malware dynamics and propagation. During malware behaviour modelling, there is a need to examine the effect of the infection rate of each node and compute the average infection rate for various network sizes. Simulations were taken for network size variations. Figure 8(a) illustrates the infection rates at varied network sizes. The proposed scheme infection rates are based on the scale (threshold) of malware prevalence and the scheme is compared to the scheme with Dynamic Analysis and Control (DAC) scheme [6], Rumour Spreading Process-Scale Free Networks (RSP-SFN) [14] and Markov Random Field-Complex Communication Networks (MRF-CCN) [8]. A snapshot of the proposed *vs*. previous schemes in terms of infection rate is depicted in Figure 8(b).

From the simulation results, the proposed scheme gave less number of infections per given number of nodes. The threshold of α is directly proportional to the malware infections. If α is small, the number of infected hosts will largely increase. In Dynamic Analysis and Control (DAC) [6], the propagation control strategies did not perform well, hence decreasing the real-time immunity rate and increasing the proportion of infected nodes. In Rumour Spreading Process-Scale Free Networks (RSP-SFN) [14], the density of infected nodes varied and increased under different vaccination rates, such as λ=0.3, ε=0.21, γ = 0.1, δ=0.05 and Λ = μ=0.07. In Markov Random Field-Complex Communication Networks (MRF- CCN) [8],



((a)) Simulated Propagation Speed.

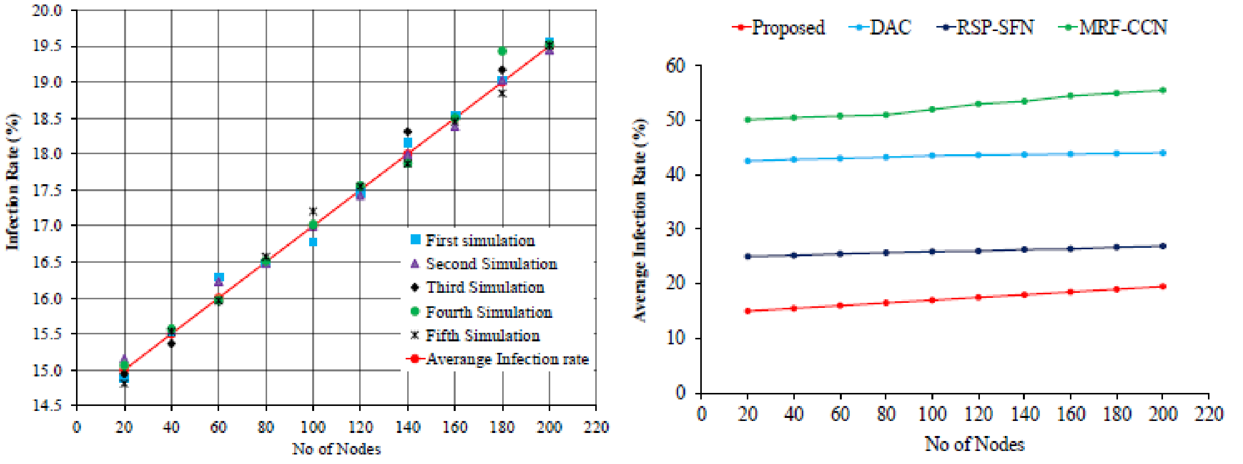
((b)) Average Simulated Propagation vs ABS-SFN [7].

Figure 8. Average infection rate.

the nodes are not reliable for long time. This leads to increasing the number of infection hosts. In our proposed scheme, the reliability is computed each time interval and also during packet transmission to monitor infection rates of the nodes in each sub-net.

4.3.3 Propagation Speed

Propagation speed was computed based on the density of nodes. The network topology greatly affects the modelling of malware propagation on IoT-based communication networks. In malware propagation, characterization of propagation speed is important. Understanding how propagation speed impacts the network is also necessary. The network size was varied in each simulation and the results of the five simulations are shown in Figure 9(a). The proposed scheme propagation speed was compared with those of the previous schemes with respect to number of nodes on varied network size as shown in Figure 10(b). In Agent-based Simulation- Scale-Free Networks (ABS-SFN) [7], the following analytical values were considered for the parameters α(k) = k−3, k = 1, 2, ...n, β = 0.3, ε = 0.01, γ = 0.08 and µ = 0.008. In addition, the reproductive ratio R0 = 3.9245 was used. If the density of infected nodes increases, the malware propagation speed also increases. The number of infected nodes increases in the ABS-SFN, whereas in our proposed scheme, the decision maker on each sub-net reduces the number of infected nodes. The proposed decision maker monitors each sub-net to determine whether it is attracted by the malware or not.



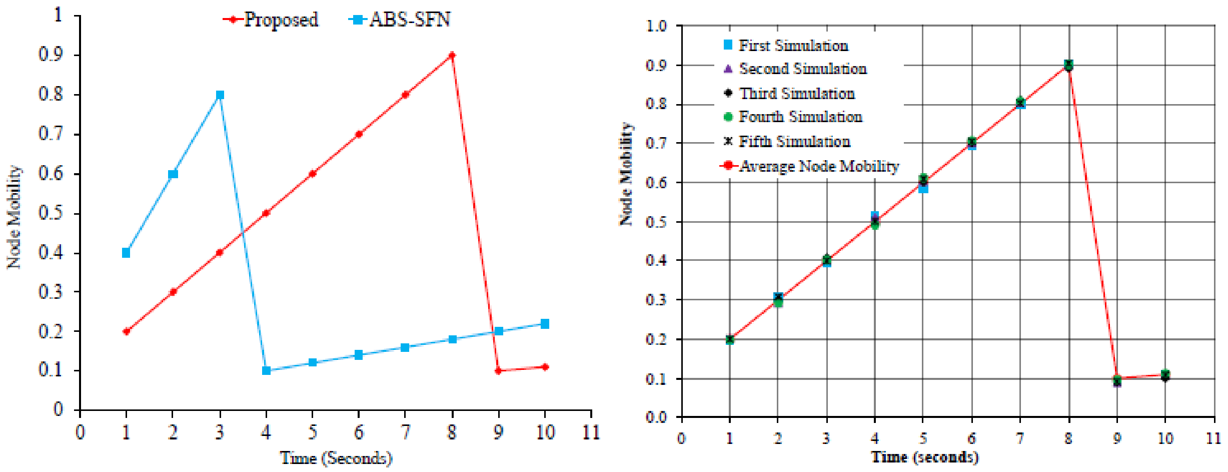
((a)) Simulated Infection Rates.

((b)) Average Infection Rate Compared with Analytical Schemes .

Figure 9. Propagation speed analysis.

4.3.4 Node Mobility

In agent-based simulation modelling, the node mobility is managed by three factors: movement detection, network connectivity or structure and location tracking. To observe node mobility, the performance at iterations i to i + 1 (between 2-4 seconds) was set in the proposed scheme. When the mobility increases above its threshold level, hub fails as noted by the decision maker and data packet transfer times between intermediate nodes are increased. In the proposed scheme, five simulations on the influence of node mobility on malware propagation were carried out. Figure 10(a) plots the mobility of nodes in a malware prone simulation against time for the five simulations. In Agent-based Simulation- Scale-Free Networks (ABS-SFN) [7], if the node mobility increases beyond the threshold, the scale-free network may disconnect. The time of the malware on the network and the malware outbreak in the sub-nets are dependent on the mobility rate. Mobility rate highly influences the spreading of network malware. When the mobility rate is smaller than the threshold value, the node in the sub-network dies. A performance comparison for node mobility between the proposed scheme and Scale-Free Networks (ABS-SFN) [7] can be seen in Figure 10.



((a)) Simulated Malware Effect on Node Mobility . ((b)) Average Node Mobility vs ABS-SFN [7].

Figure 10. Malware effect on node mobility.

5. Conclusion and Future Work

Agent-based modelling simulation in complex networks is a challenging issue. In this paper, we developed a malware propagation model using agent-based approach and deep-reinforcement learning on a scale-free network in IoT. In the modelled system, Susceptible-Infected-Immuned-Recovered-Removed (SIIRR) transitions were formulated. The effect of malware propagation on the model was evaluated based on performance metrics, such as average energy consumption *vs.* number of nodes, average infections over time, node mobility over time period t and spreading/propagation speed. Our simulations showed that the introduction of a DRL-based decision maker results in a more versatile IoT model, where malware propagation is not just based on contact.

As future work, we intend to explore model stability analysis and the effect of immunization on different devices in IoT. The stability analysis will entail global and local model equilibrium. For the effect of immunization, we plan to incorporate mechanisms, such as targeted and proportional immunization, in the model. Employing immunization and quarantine mechanisms can offer a promising approach to make the model more realistic and resilient.

Acknowledgements

This research was partly supported by the European Commission, which is the sponsor of the METEGA project through its Education, Audio-visual and Culture Executive Agency (EACEA).

REFERENCES

[1] S.-M. Cheng, W. C. Ao, P.-Y. Chen and K.-C. Chen, "On Modeling Malware Propagation in Generalized Social Networks," IEEE Communications Letters, vol. 15, no. 1, pp. 25–27, 2011, [Online], Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5638768.

[2] S. Sneha, L. Malathi and R. Saranya, "A Survey on Malware Propagation Analysis and Prevention Model," International Journal of Advancements in Technology, vol. 6, no. 1, pp. 1–4, 2015, [Online], Available: http://dx.doi.org/10.4172/0976-4860.1000148.

[3] M. Yasir, M. A. Habib, M. Shahid and M. Ahmad, "Agent-based Modeling and Simulation of Virus on a Scale-free Network," Proceedings of the International Conference on Future Networks and Distributed Systems (ICFNDS '17), New York, NY, USA: ACM, pp. 59:1–59:6, 2017, [Online], Available: http://doi.acm.org/10.1145/3102304.3109819.

[4] H. Kırer and Y. A. Çırpıcı, "A Survey of Agent-based Approach of Complex Networks," Ekonomik Yaklasim, vol. 27, no. 98, pp. 1–28, 2016, [Online], Available: https://www.ejmanager.com/mnstemps/94/94-1404633261.pdf?t=1552958497

[5] A. M. del Rey, A. H. Encinas, J. M. Vaquero, A. Q. Dios and G. R. Sánchez, "A Cellular Automata Model for Mobile Worm Propagation," International Work-Conference on the Interplay between Natural and Artificial Computation, Springer International Publishing, pp. 107–116, 2015, [Online], Available: http://dx.doi.org/10.1007/978-3-319-18833-1_12.

[6] L. Feng, X. Liao, Q. Han and H. Li, "Dynamical Analysis and Control Strategies on Malware Propagation Model," Applied Mathematical Modelling, vol. 37, no. 16-17, pp. 8225–8236, 2013, [Online], Available: https://doi.org/10.1016/j.apm.2013.03.051.

[7] S. Hosseini, M. Abdollahi Azgomi and A. Rahmani Torkaman, "Agent-based Simulation of the Dynamics of Malware Propagation in Scale-free Networks," Simulation, vol. 92, no. 7, pp. 709–722, 2016, [Online], Available: https://doi.org/10.1177/0037549716656060.

[8] V. Karyotis, "A Markov Random Field Framework for Modeling Malware Propagation in Complex Communications Networks," IEEE Transactions on Dependable and Secure Computing, 2017, [Online], Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7926392.

[9] K. Batool and M. A. Niazi, "Modeling the Internet of Things: A Hybrid Modeling Approach Using Complex Networks and Agent-based Models," Complex Adaptive Systems Modeling, vol. 5, no. 1, p. 4, 2017, [Online], Available: https://doi.org/10.1186/s40294-017-0043-1.

[10] A. Bose and K. G. Shin, "Agent-based Modeling of Malware Dynamics in Heterogeneous Environments," Security and Communication Networks, vol. 6, no. 12, pp. 1576–1589, 2013, [Online], Available: http://doi.acm.org/10.1145/1378600.1378626.

[11] V. Karyotis and S. Papavassiliou, "Macroscopic Malware Propagation Dynamics for Complex Networks with Churn," IEEE Communications Letters, vol. 19, no. 4, pp. 577–580, 2015, [Online]. Available: https://ieeexplore.ieee.org/iel7/4234/5534602/07029645.pdf.

[12] A. M. del Rey, A. H. Encinas, J. M. Vaquero, A. Q. Dios and G. R. Sánchez, "A Method for Malware Propagation in Industrial Critical Infrastructures," Integrated Computer-aided Engineering, vol. 23, no. 3, pp. 255–268, 2016, [Online], Available: http://dx.doi.org/10.3233/ICA-160518.

[13] E. M. Karanja, S. Masupe and J. Mandu, "Internet of Things Malware: A Survey," International Journal of Computer Science & Engineering Survey, vol. 8, no. 3, pp. 1–20, Jun. 2017, [Online], Available: http://aircconline.com/ijcses/V8N3/8317ijcses01.pdf.

[14] S. Hosseini and M. A. Azgomi, "A Model for Malware Propagation in Scale-free Networks-based on Rumor Spreading Process," Computer Networks, vol. 108, pp. 97–107, 2016, [Online], Available: https://doi.org/10.1016/j.comnet.2016.08.010.

ملخص البحث:

يُعدّ تهديد الاختراقات الضّارة عائقاً رئيسياً أمام تبادل المعلومات بشكلٍ فعّالٍ في إنترنت الأشياء. ويعدّ موضوع نمذجة الاختراقات الضّارة أحد أهم التطبيقات الملحّة الهادفة الى فهم آليات حماية بيئة إنترنت الأشياء. ويمكن تحقيق الحماية المطلوبة لإنترنت الأشياء باستخدام نمذجة قائمة على الاستفادة من وسائل الحماية في الشبكات المعقّدة.

تُقدم هذه الورقة تفصيلاتً متعمقة حول نموذج مقترح للحماية من الاختراقات الضّارة يقوم على استخدام وسائل حماية، إضافة الى تقنية التعلم المستند الى التعزيز العميق، في شبكة غير محددة الحجم من شبكات إنترنت الأشياء.

ويُسمى النموذج المقترح في هذا البحث طبقاً لحالات الانتقال التي يتضمنها: (مشكوكٌ فيه؛ مصابٌ بالعدوى؛ محصَّن؛ متماثلٌ "اللشِّفاء:؛ منزوعٌ)، وهي الحالات التي تعبر عن حالات العُقد في الشبكات المعقدة كبيرة الحجم. ويتم استقصاء موثوقية كل عُقدة باستخدام متوسط الوقت حتى الفشل. أمّا العوامل التي تؤخذ بعين الاعتبار في حساب متوسط الوقت حتى الفشل فهي: درجة العقدة، ومعدل حركية العُقدة، ومعدل الإرسال بالنسبة للعُقدة، والمسافة بين غقدتين محسوبةً وفق المسافة الإقليدية.

ويتضح من النتائج أن النموذج المقترح في هذه الدراسة قابلٌ للمقارنة مع نماذج سابقة مماثلة تتعلق بتأثيرات انتشار الاختراقات الضّارة من حيث معدل استهلاك الطاقة، ومعدل العَدْوى في زمنٍ معين، وحركية العُقد، وسرعة الانتشار.