

# Multigroup e-SIjRjTj Model for Malware Propagation in Wireless Sensor Networks

Chukwunonso Henry Nwokoye, *Member, IAENG*

**Abstract**—Different fields have established the possibility of the existence and mathematical characterization of heterogeneous populations. Initially, it was seen in biosciences, and this motivated its application to computers and wireless sensor networks (WSNs). These differential equation models for multiple concurrent infections (mostly worms, viruses, and Trojans) try to generate the reproductive ratios and, subsequently, show the stability analyses of existing equilibriums. However, a different approach is proposed, i.e., the combined application of agent-based and mathematical models in order to exploit the benefits of both methods. Specifically, the complex dynamics and interplay between worms, viruses, and Trojans are represented using the e-SIjRjTj model. Additionally, this model was reified through the development of the Multigroup MaliciousCode Simulator (MMS) using the NetLogo language. This is to allow for class heterogeneity and stochasticity factors that are almost impossible to include in mathematical models due to their inherent assumption of homogeneity. With MMS, several possible probabilities, such as infection scanning, were implemented in order to remediate infected nodes. Numerical simulations were conducted for both approaches, and this is aimed at eliciting the impact of security strategies. Note that the mathematical model was solved using the Runge-Kutta-Fehlberg order 4 and 5 (RK45) numerical method. Finally, the study presented discussions concerning optimized control strategies, modification of reproduction ratios to capture long-term behavior, and computational overhead, which are sources of increased resource utilization and depletion.

**Index Terms**—epidemic theory, malware, ordinary differential equation, wireless sensor networks, benchmark analytical model

## I. INTRODUCTION

THE cyberwar against information and communication technology infrastructures has become a vital concern to managers of organizational security [1], and this is due to the continued exposure to vulnerabilities on the web [2]. This cyber-oriented act of aggression and warfare is due to malicious code attacks that amount to a significant threat, which cause damage and disruption to businesses, and organizations, as well as their operations. Specifically, while malware refers to codes, scripts, or other kinds of software designed with malicious intentions, its varieties include Trojans, viruses, worms, spyware, adware, and ransomware [3]. To guarantee the integrity of data and information, experts strive to thwart security threats to the system through

the use of firewalls, anti-malware, and honeypot technology [4]. Many others devote time to proposing models that allow the understanding of malware spread patterns in communication networks [5]–[8]. However, most of these mathematical models represent homogenous populations of one type of malware at a time [5], [9]. This portrays partly the truth because multiple types of malwares can exist on a network at once, and this has been supported by biological evidence originating from the biosciences. These malware kinds now equal heterogeneity, and their representations are called multigroup modeling.

In a study by Driessche and Watmough [10], they posited that, “‘multigroup’ usually refers to the division of a heterogeneous population into several homogeneous groups based on individual behavior... (where) each group is then subdivided into epidemiological compartments”. Here, the existing groups are divided into subclasses in light of their behavior. For disease modeling in the biosciences, this approach is used to characterize sexually transmitted infections, for example, gonorrhea or HIV/AIDS—diseases where behavior is a profound essentiality for the chances of acquiring an infection. Additionally, an SIR model was used by Hyman and Li [11] to describe the specifics of transmitting an infectious disease in a biological network. Therein, the susceptible class was divided into different groups. Owing to several challenges attributed to compartmental models, this paper therefore advocates their use alongside individual-based models in the form of agent-based models (ABM) to represent the intractable epidemic dynamics of the spread of worms, viruses, and Trojans in a wireless sensor network (WSN) case. Fundamentally, this study contemplates the move beyond intractable mathematical exploration to the application of spatial parameters (stochasticity), not easily attainable with the differential equation method. Note that WSN is a network of numerous sensors distributed in such a manner that they sense, monitor, and gather information about the environment [12], [13].

The rest of the paper is arranged in the following manner: Section 2 contains the related works, while Section 3 holds the chosen methodology for the study. In Section 4, the SIjRjTj–S Mathematical Model was described, whereas in Section 5, the Multigroup MaliciousCode Simulator (MMS) was described. Section 6 contains the implementation of the proposed models as well as the simulation experiments. Finally, Section 7 holds the concluding statements and future directions.

## II. RELATED WORK

Here, works that mathematically model multi-group infections in biological networks as well as computer and

Manuscript received August 5, 2024; revised May 03, 2025.

C. H. Nwokoye is a postdoctoral research fellow at the Trustworthy AI Lab, Ontario Tech University, Oshawa, Canada (phone number: +1-437-599-8720, +234-703-385-8720; e-mail: chinonsohwokoye@gmail.com; HenryChukwunonso.Nwokoye@ontariotechu.ca).

wireless sensor networks are listed and reviewed. Additionally, papers that support the utilization of the ABM paradigm to even epidemics in biological networks, as well as other wireless networks are reviewed. Epidemic literature on technological networks has instances of multi-group models that characterize more than one infection type [5]. Most of these models establish the threshold for malware replication in the network and, thereafter, show stability analysis at the existent equilibrium points [5]. Furthermore, numerical methods were used to solve and simulate the system of differential equations in order to enable us comprehend the attacking dynamics of malware variants in computer networks alongside the efficacy of applied countermeasures. In order to safeguard the cyberspace from various kinds of malicious objects, Mishra and Singh [14] considered simple mass action incidence and developed a “susceptible, infectious due to worm, infectious due to virus, Infectious due to Trojan Horse, Recovered and Susceptible (SI<sub>1</sub>I<sub>2</sub>I<sub>3</sub>RS)” model for malware distribution in computer networks. Ref [15] proposed the “differential electronic susceptible-infectious-removed-susceptible (e-SIRS) model” for the transmission of worms and viruses over a computer network. The model divided the susceptible and infectious compartments for worm and virus transmission. Their work had periods of latency, immunity, and self-replication. Ref [16] developed the “differential susceptible e-epidemic model S<sub>j</sub>IR (susceptible class-1 for virus (S<sub>1</sub>) - susceptible class-2 for worms (S<sub>2</sub>)-susceptible class-3 for Trojan horse (S<sub>3</sub>)-infectious (I)-recovered (R))” for malware transfer on a computer network. Here the authors included a distribution/division of the new entrants into three susceptible subgroups: worms, viruses, and Trojans. In the light of the essential dynamics of mixing using the law of mass action, this model involves the inherent assumption of spatial distribution in homogenous terms. Ref [17] developed the “susceptible, infectious due to worm, infectious due to virus, recovered and susceptible (SI<sub>1</sub>I<sub>2</sub>RS)” epidemic model to investigate the effect and transmission of these malicious programs on WSN.

The mathematical modeling of one type of malware in WSN was done by [18], and these kinds of models litter the literature on WSN. However, it was observed that there are a few instances where models characterize the propagation of more than one malicious object at a time; they are discussed below. Ref [19] developed the “susceptible, infectious due to virus, infectious due to worm, infectious due to Trojan, recovered, and susceptible with vaccination (S<sub>j</sub>IR<sub>s</sub>-V)” epidemic model to represent the spread dynamics of multiple types of malicious code. Their work involved distribution density and communication range—characteristic attributes of a WSN. In their words, “the need for performing multigroup modeling for WSN is drawn from two reasons, which include (a) the fact that epidemic models have been used to model the propagation of viruses and that of worms... and (b) the strong possibility that both worms and viruses might exist at the same time”. Ref [20] proposed the “susceptible–exposed (due to worm)–exposed (due to virus)–infectious (due to worm)–infectious (due to virus)–recovered–susceptible model with vaccination (SE<sub>j</sub>I<sub>j</sub>R<sub>s</sub>-V) epidemic model”, with mass action incidence. This is a modification of the SEIR-V model proposed first by [21] and

subsequently modified by [22]. A recent mathematical model incorporating multigroup modeling was proposed by [23] to address the issues of multiple malware contagions, external noise, and time delay.

ABM has been successfully used for the representation of complex networks; the following review highlights some of their objectives. On malware spread in ICT applications/infrastructures, Bose and Shin [24] highlighted challenges and inadequacies of traditional analytical models, which include homogeneity, average-degree distributions, and perfect mixing. Consequently, they developed an agent-based malware modeling framework that combines the capabilities of simulating, on one hand, malware propagation features and, on the other hand, network attributes such as structure, mobility, and application-level exchange. The ubiquity of sensors and transceivers, as well as their importance in the popular IoT, as well as the absence of a standard modeling methodology for modeling complex systems, motivated these authors [25] to propose a cognitive agent-based computing framework for such networks. Note that the emphasis here is not on malware spread but on the representation of IoT technologies using network topologies, for instance, random, lattice, scale-free, small-world networks. Ref [22] conceived a hybrid approach for modeling malicious objects in WSNs. The new approach merges the benefits of agent-oriented software engineering and the epidemic modeling and analysis of cyber-dynamical systems. A case of the mathematical “susceptible-exposed-infected-recovered-vaccinated (SEIR-V) model” was reified using this ABM method, and this resulted in a computational model called the Sensor Worm Spread Simulator. Batista et al. [26] used the agent-based paradigm to represent the “susceptible-exposed-infected-recovered-dead” (SEIRS-D) epidemic model. Their work was analyzed using the Mesa framework to represent its impact on three different topologies, namely mesh, star, and hybrid.

The NetLogo simulation platform has been used to model areas such as routing and resource management. Ref [27] modeled TDMA-based MAC and scheduling as a fraction of the whole proposed algorithm. The powerful visualization advantage of NetLogo was employed by [28] to model mobile ad hoc networks and by [29] to model cognitive wireless networks. Going beyond older achievements, Wasti [30] developed a radio environment model, time division multiple access, and a channel allocation model based on user priorities. Peer-to-peer worms were also modeled by [31] in order to investigate several immunization strategies. More works that include the use of either agent-based models or mathematical models for understanding WSNs [32], [33], [34], [35]. Note that in the work of Driessche and Watmough [10], there is a tendency to confuse multi-group with the multi-strain model. However, the “multi-group” terminology was still maintained in this study. This is because it entails the nuances and specifics of their multigroup model, i.e., dividing the infections in the light of the heterogeneous behaviors of worms, viruses, and Trojans in networks. The proposed multiagent model characterizes a heterogeneous population made up of subgroups of homogenous classes of infectious, recovered, and resistant compartments. The decision to apply the agent paradigm is hinged on the fact that “malware propagation models based on differential equations have

drawbacks, which include homogenous mixing and distribution, inability to represent individual dynamical behavior, and the inability to account for local infections between nodes in a network” [22].

### III. METHODOLOGY

This study employed a recently developed flavor of the ABM called “the analytic-agent cyber dynamical systems analysis and design method (A<sup>2</sup>CDSADM)” [22]. Actually, A<sup>2</sup>CDSADM is a hybrid approach that employs the strengths of epidemic modeling and analysis of cyber dynamical systems and the agent-oriented software engineering approaches. The method allows the creation of a “benchmark analytical model (BAM)” alongside the development of the ABM, i.e., MMS in this case. Note that the BAM is a system of differential equations and is used for initial validation of the MMS. This method was chosen because it makes the developed tool easily modifiable and reproducible, thus diminishing the less-tractable nature of performing complex extended stability analyses of equilibrium points. Additionally, it suggests an approach to performing comparative epidemic investigations between the computational ABMs and traditional analytical models.

### IV. THE E-SIJRJTj-S MATHEMATICAL MODEL

The agent computational model was built alongside the analytical multigroup Susceptible–Infectious due to virus–Infectious due to worm–Infectious due to Trojan horse–Recovered (virus)–Recovered (worm)–Recovered (Trojan horse)–Resistant to virus–Resistant to worm–Resistant to Trojan horse (e-SIJRJTj-S) model, and its parameters are presented in Table 1. Note that this mathematical model is a system of ordinary differential equations (ODEs) that characterizes the transmission and control of malware in a WSN. The transition rules regulating the interaction dynamics are present below. In view of these rules, the diagram of transmission (Fig. 1) and the system of equations were generated and presented as follows:

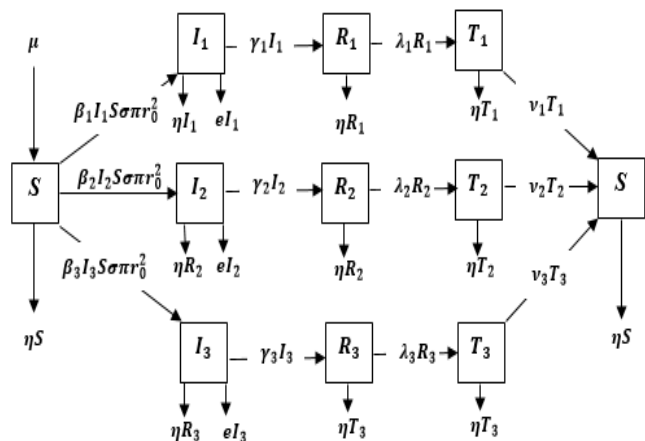


Fig. 1. Schematic diagram of the e-SIJRJTj-S model

Step 1: Sensor Deployment and Network Initialization. Here, sensors are randomly deployed at a rate of  $\mu$ .

Step 2: State Initialization. After deployment, it is assumed that the sensor nodes are susceptible (S) and crash out due to software/hardware failure at the rate of  $\eta$ .

- Case 1: If attacks resulting from viruses, worms, and Trojan horses occur, susceptible sensor nodes change their states to infectious (due to worm ( $I_1$ )), infectious (due to virus ( $I_2$ )), infectious (due to Trojan horse ( $I_3$ )), at the following rates, assuming the mixing and interaction are homogenous;  $\beta_1 I_1 S \sigma \pi r^2$ ,  $\beta_2 I_2 S \sigma \pi r^2$  and  $\beta_3 I_3 S \sigma \pi r^2$  respectively. Alternatively, the WSN population advances towards the carrying capacity given as  $\mu/\eta$ . Note that sensors can also be removed due to software/hardware failure and due to malware infection at rates  $e$  and  $\eta$ , respectively.
- Case 2: If the WSN is integrated with an antimalware capability, which can remediate sensors infected with worms, viruses, and Trojan horses, then sub-classes of the infectious nodes ( $I_1, I_2, I_3$ ) can change to sub-classes of recovered nodes ( $R_1, R_2, R_3$ ) at the following rates;  $\gamma_1, \gamma_2, \gamma_3$ . Additionally, nodes can crash out due to software/hardware failure at the rate of  $\eta$ .
- Case 3: If the antimalware is continually updated, providing protection for sensors against hackers, the recovered nodes ( $R_1, R_2, R_3$ ) become resistant ( $T_1, T_2, T_3$ ) at the following rates;  $\lambda_1, \lambda_2, \lambda_3$ , and they may subsequently return to the susceptible (S) state at rates  $\nu_1, \nu_2, \nu_3$ , after they lose their transient immunity. More so, nodes can crash out due to software/hardware failure at the rate of  $\eta$ .

TABLE I:  
PARAMETER DESCRIPTION

Parameters	Meaning
$\mu$	Rate of inclusion of nodes in the WSN
$r$	Communication range
$\sigma$	distribution density
$\beta_1$	Infection contact rate for worm
$\beta_2$	Infection contact rate for virus
$\beta_3$	Infection contact rate for Trojan horse
$e$	Death rate due to malware attack
$\eta$	Death rate due to software/hardware failure
$\gamma_1$	Rate at which nodes recover from worm infections
$\gamma_2$	Rate at which nodes recover from virus infections
$\gamma_3$	Rate at which nodes recover from Trojan infections
$\lambda_1$	Rate at which nodes recovered nodes become resistant to subsequent worm infections
$\lambda_2$	Rate at which nodes recovered nodes become resistant to subsequent virus infections
$\lambda_3$	Rate at which nodes recovered nodes become resistant to subsequent Trojan infections
$\nu_1$	Rate at which resistant nodes lose their immunity thereby becoming susceptible to worm infection
$\nu_2$	Rate at which resistant nodes lose their immunity thereby becoming susceptible to virus infection
$\nu_3$	Rate at which resistant nodes lose their immunity, thereby becoming susceptible to Trojan infections

The system of equations characterizing the e-SIJRJTj-S epidemic model is as follows;

$$\begin{aligned} \dot{S} &= (\mu - \eta) S - S \sigma \pi r^2 \sum_{j=1}^3 \beta_j I_j + \sum_{j=1}^3 \nu_j T_j \\ \dot{I}_j &= \sum_{j=1}^3 \beta_j I_j S \sigma \pi r^2 - \gamma_j I_j - (\eta + e) I_j \\ \dot{R}_j &= \sum_{j=1}^3 \gamma_j I_j - (\lambda_j + \eta) R_j \\ \dot{T}_j &= \sum_{j=1}^3 \lambda_j R_j - (\nu_j + \eta) T_j \end{aligned} \quad (1)$$

In summary, the susceptible (S) class reduces on account of emerging malware infections and death (from malware attack and software or hardware failure), but rises owing to sensor node inclusion and loss of immunity of resistant nodes. The infectious (I<sub>j</sub>) classes grow owing to new infections, then reduce due to sensor node recoveries and deaths as mentioned above. The recovered (R<sub>j</sub>) classes show that a sensor has recovered from a particular malware infection. These compartments diminish as time passes owing to deaths, as mentioned above. The resistant (T<sub>j</sub>) classes represent sensors that have acquired some form of resistance. This class grows with the outflow from the recovered classes and diminishes as a result of deaths. Finally, due to loss of immunity in cyberspace resulting from outdated anti-malware software, the resistant sensors can become susceptible again.

A. Existence of Equilibrium

The solutions to the system of equations were derived by simply equating to zero. The results are solutions at the malicious code-free equilibrium (MFE) and the endemic equilibrium (EE). The MFE is as follows:

$$S^0 = 0, I_j^0 = 0, R_j^0 = 0, T_j^0 = 0 \tag{2}$$

While the EE include;

$$\begin{aligned} S^* &= \sum_{j=1}^3 \frac{e+\eta+\gamma_j}{\pi^2 \sigma \beta_j} \\ I^* &= \sum_{j=1}^3 \frac{(\eta-\mu)(e+\eta+\gamma_j)(\eta+\lambda_j)(\eta+\nu_j)}{\pi^2 \sigma \beta_j ((e+\eta)(\eta+\lambda_j)(\eta+\nu_j) + \eta \gamma_j (\eta+\lambda_j+\nu_j))} \\ R^* &= \sum_{j=1}^3 \frac{(\eta-\mu) \gamma_j (e+\eta+\gamma_j)(\eta+\nu_j)}{\pi^2 \sigma \beta_j ((e+\eta)(\eta+\lambda_j)(\eta+\nu_j) + \eta \gamma_j (\eta+\lambda_j+\nu_j))} \\ T^* &= \sum_{j=1}^3 \frac{(\eta-\mu) \gamma_j (e+\eta+\gamma_j) \lambda_j}{\pi^2 \sigma \beta_j ((e+\eta)(\eta+\lambda_j)(\eta+\nu_j) + \eta \gamma_j (\eta+\lambda_j+\nu_j))} \end{aligned} \tag{3}$$

B. The Basic Reproduction Number (R<sub>0</sub>)

The reproduction number or ratio is the projected number of resulting cases arising in a totally vulnerable population from a particular infectious host [14]–[17], [21], [22]. To generate the reproduction ratio, the method used in [19] and [36] was applied, where it equals “the inverse of the susceptible compartment at endemic equilibrium” [20]. Note that this approach is based on the study of Diekmann, et. al., [37]. While there are many approaches to finding the R<sub>0</sub>, such as the next generative method (NGM), it is noteworthy that the approach employed here essentially generates the same result as when using NGM.

Therefore, the basic R<sub>0</sub> for the SI<sub>j</sub>R<sub>j</sub>T<sub>j</sub>-S model is given as follows;

$$R_0 = \frac{\pi^2 \sigma \beta_1}{e + \eta + \gamma_1} + \frac{\pi^2 \sigma \beta_2}{e + \eta + \gamma_2} + \frac{\pi^2 \sigma \beta_3}{e + \eta + \gamma_3} \tag{4}$$

C. Local Stability Analysis (LAS) of the MFE

As it is the norm, most WSN papers have conducted analyses for both local and global stabilities [5], [6]. Here, the local stability of the MFE using the Jacobian matrix method and eigenvalue analysis was analyzed. In epidemic models of computer networks [14], [16], and WSNs [21], [22], [36], LAS at MFE was done using the Jacobian matrix (JM) and eigenvalue analysis (EVA) for negative real parts. The SymPy library, which allows for symbolic computation for both LAS and GAS, was utilized. The JM depicts the linearized model around a certain equilibrium point and

involves partial derivatives of equations respecting each class or compartment. If X is the model, then the JM is given as:

$$X = f(X) \tag{5}$$

$$J(X) = \partial f(X) / \partial X \tag{6}$$

$$J(X) = \begin{bmatrix} \frac{\partial S}{\partial S} & \frac{\partial S}{\partial T_1} & \dots & \frac{\partial S}{\partial T_3} \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial T}{\partial S} & \frac{\partial T}{\partial T_1} & \dots & \frac{\partial T}{\partial T_3} \end{bmatrix} \tag{7}$$

The eigenvalues derived when conducting LAS at the MFE are as follows:  $-\eta + \mu$ ,  $-\eta - \lambda_1 - \eta - \nu_1$ ,  $-\eta - \lambda_2 - \eta - \nu_2$ ,  $-\eta - \lambda_3 - \eta - \nu_3$ ,  $S\beta_1\pi^2\sigma - e - \eta - \gamma_1$ ,  $S\beta_2\pi^2\sigma - e - \eta - \gamma_2$ ,  $S\beta_3\pi^2\sigma - e - \eta - \gamma_3$ . For LAS, all the eigenvalues should have negative real parts. Consequently, the system is locally asymptotically stable if  $S\beta_1\pi^2\sigma < e + \eta + \gamma_j$  and  $\mu < \eta$ . For the latter, the implication is that this condition guarantees that the rate of sensor inclusion is not greater than the mortality rate due to software/hardware issues. As long as this holds,  $\mu < \eta$  will remain negative, thus maintaining stability.

D. Global Stability Analysis (GAS) of the MFE

Along with local stability, analysis of global stability gives information regarding a dynamical system's behavior across a prolonged time period. In epidemic models of computer networks [50] and WSNs [21], [22], [36], GAS at MFE was done using the following: a combination of JM and EVA [14], [16], Lyapunov's stability theory (LST) as well as the combination of JM, EVA, and the Perron-Frobenius theorem [21], [50]. However, for the e-SI<sub>j</sub>R<sub>j</sub>T<sub>j</sub> model, the global stability will be analyzed using the Lyapunov stability approach. LST can be done using two strategies: the first strategy involves identifying just the infected compartments, which equal zero, i.e.,  $I_j = 0$  or  $I_1 = I_2 = I_3 = 0$ . This method focuses on the infection dynamics and is adjudged straightforward and sufficient. The second strategy takes an extended outlook, i.e., where the modeler decides to use the whole system instead. This approach has been adjudged holistic as it captures the behavioral dynamics of the entire system. The steps to performing LAS at the MFE are presented below.

Identification of the equilibria: This is the MFE, and the two approaches were employed in the study.

$$I_1 = I_2 = I_3 = 0 \tag{8}$$

Actual Lyapunov function definition: This is a scalar V that is positive definite and enables the determination of the system's stability. This scalar function was applied to the system so as to cater to both approaches. The Lyapunov function was defined below as;

$$V(I_1, I_2, I_3) = \frac{I_1^2}{2} + \frac{I_2^2}{2} + \frac{I_3^2}{2} \tag{9}$$

For the entire system, the corresponding Lyapunov function is as follows:

$$V(S, I_j, R_j, T_j) = S + \frac{I_j^2}{2} + \frac{R_j^2}{2} + \frac{T_j^2}{2} \tag{10}$$

Notice that the scalar function was divided by 2. The division by 2 is not necessarily required for the function to

meet the Lyapunov criteria (i.e., be positive definite and diminishing throughout trajectories), but it simplifies and standardizes derivative computations.

Time derivative computation of the Lyapunov function: Here, V's derivative considering time along the model's trajectory was computed. To calculate the time derivative, the component of 1/2 cancels out while conducting differentiation, resulting in simpler equations. This can be very relevant for analysis of stability and control theory. V's time derivative is as follows:

$$\frac{dV}{dt} = \frac{\partial V}{\partial I_1} \frac{dI_1}{dt} + \frac{\partial V}{\partial I_2} \frac{dI_2}{dt} + \frac{\partial V}{\partial I_3} \frac{dI_3}{dt} \quad (11)$$

Applying the above Lyapunov function results in the following:

$$\frac{\partial V}{\partial I_1} = I_1, \frac{\partial V}{\partial I_2} = I_2, \frac{\partial V}{\partial I_3} = I_3 \quad (12)$$

Hence, the final time derivative using V becomes:

$$\frac{dV}{dt} = I_1 \frac{dI_1}{dt} + I_2 \frac{dI_2}{dt} + I_3 \frac{dI_3}{dt} \quad (13)$$

Substitute the dynamics of the system and analyze the signs of dV/dt. After substituting the expressions for the state variables, the signs of the derived solution are analyzed, i.e., dV/dt ≤ 0, and it should be strictly negative for values of the infectious sub-compartments. This is the case if the modeler is focusing on the infectious compartments alone. The resulting equations are as follows:  $\frac{dI_1}{dt} = f_1(I_1, I_2, I_3)$ ,  $\frac{dI_2}{dt} = f_2(I_1, I_2, I_3)$ ,  $\frac{dI_3}{dt} = f_3(I_1, I_2, I_3)$ .

$$\frac{dV}{dt} = I_1 f_1(I_1, I_2, I_3) + I_2 f_2(I_1, I_2, I_3) + I_3 f_3(I_1, I_2, I_3) \quad (15)$$

Given the following,  $\Psi_1 = \pi S \beta_1 r^2 \sigma + e + \eta + \gamma_1$ ;  $\Psi_2 = \pi S \beta_2 r^2 \sigma + e + \eta + \gamma_2$ ;  $\Psi_3 = \pi S \beta_3 r^2 \sigma + e + \eta + \gamma_3$ ; a more condensed form of this solution for the infectious compartments is given as:

$$\frac{dV}{dt} = -I_1^2 (-\Psi_1) - I_2^2 (-\Psi_2) - I_3^2 (-\Psi_3) \quad (16)$$

Given the following:  $\Psi_1 = \pi S \beta_1 r^2 \sigma + e + \eta + \gamma_1$ ;  $Q_1 = \pi I_1 S \beta_1 r^2 \sigma$ ;  $\Psi_2 = \pi S \beta_2 r^2 \sigma + e + \eta + \gamma_2$ ;  $Q_2 = \pi I_2 S \beta_2 r^2 \sigma$ ;  $\Psi_3 = \pi S \beta_3 r^2 \sigma + e + \eta + \gamma_3$ ;  $Q_3 = \pi I_3 S \beta_3 r^2 \sigma$ ;  $\Omega_1 = I_1 \gamma_1 + R_1 \eta + R_1 \lambda_1$ ;  $\Omega_2 = I_2 \gamma_2 + R_2 \eta + R_2 \lambda_2$ ;  $\Omega_3 = I_3 \gamma_3 + R_3 \eta + R_3 \lambda_3$ ;  $Z_1 = R_1 \lambda_1 + T_1 \eta + T_1 v_1$ ;  $Z_2 = R_2 \lambda_2 + T_2 \eta + T_2 v_2$  and  $Z_3 = R_3 \lambda_3 + T_3 \eta + T_3 v_3$ ; a more condensed form of this solution for the entire system is given as:

$$\frac{dV}{dt} = -I_1^2 (-\Psi_1) - T_1 (-\Omega_1) - R_2 (-\Omega_2) - R_3 (-\Omega_3) - S(\eta - \mu) + T_1 v_1 - T_1 (-Z_1) + T_2 v_2 - T_2 (-Z_2) + T_3 v_3 - T_3 (-Z_3) \quad (17)$$

To assess global asymptotic stability, it is necessary to illustrate that the time derivative is strictly negative, dV/dt < 0. Firstly, the quadratic terms (-I<sub>1</sub><sup>2</sup>, -I<sub>2</sub><sup>2</sup>, -I<sub>3</sub><sup>2</sup>) for each infection type possess the following expressions:  $\Psi_1, \Psi_2, \Psi_3$ . They contribute in a negative manner when the infectious,

recovery, and death rates are positive, resulting in the decay of the infection with time. While these terms; Q<sub>1</sub>, Q<sub>2</sub>, and Q<sub>3</sub> show the infection spread, they are balanced by the negative quadratic terms. Secondly, the terms for the recovered compartments, i.e., R<sub>1</sub>(η + λ<sub>1</sub>), R<sub>2</sub>(η + λ<sub>2</sub>), and R<sub>3</sub>(η + λ<sub>3</sub>) are balanced by negative terms such as -I<sub>1</sub>γ<sub>1</sub>, -I<sub>2</sub>γ<sub>2</sub>, -I<sub>3</sub>γ<sub>3</sub>, which ensures that infection reduces over time. Thirdly, the expression reflecting both the rates of death and inclusion at the susceptible class is -S(η - μ), and for stability η > μ. Fourthly, -R<sub>1</sub>λ<sub>1</sub>, -R<sub>2</sub>λ<sub>2</sub>, -R<sub>3</sub>λ<sub>3</sub> have negative influence and balances the resistant compartments with terms depicting the rates of losing immunity in cyberspace, i.e., T<sub>1</sub>(η + v<sub>1</sub>), T<sub>2</sub>(η + v<sub>2</sub>), T<sub>3</sub>(η + v<sub>3</sub>).

For confirmation, the quadratic approach of Lyapunov function was further illustrated by numerical simulation of the infectious compartments depicted as Figs 2 and 3. The following numerical values were used for the simulation; S = 100, beta values = (0.3, 0.2, 0.1), gamma values = (0.1, 0.1, 0.1), sigma (σ) = 1.0, r = 1.0, η = 0.01 and e = 0.1. Evidently the graph below shows that the infectious compartments approach zero. Also, it was observed that changing the values of S from 1 to 100, the various infections tend toward zero.

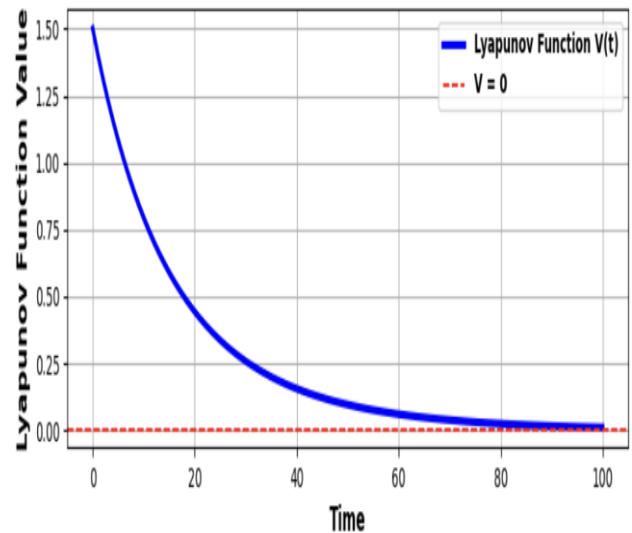


Fig 2. Lyapunov Function and Infected Compartment Dynamics

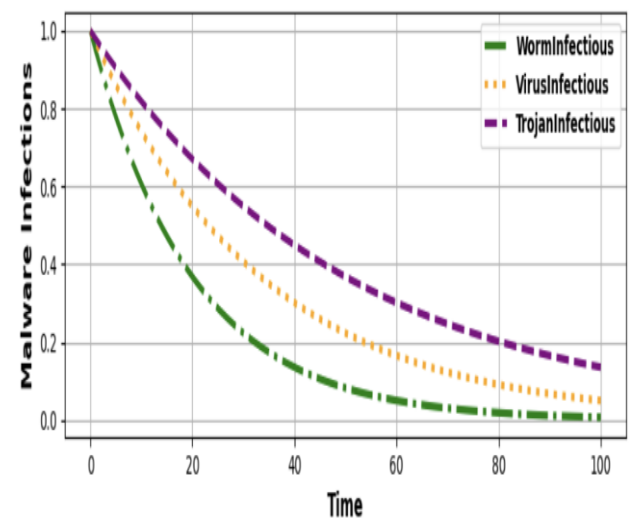


Fig 3. Infected Compartments and Their Rate of Change

V. MULTIGROUP MALICIOUS-CODE SIMULATOR (MMS)

The attempt to reify the analytical SIJRjTj-S model resulted in the agent computational model, i.e., the Multigroup MaliciousCode Simulator (MMS). This requires several agents and the WSN environment. The agents here are basically the sensors and the malware types, i.e., viruses, worms, and Trojan horses. Representing the state variables of the SIJRjTj-S model, the sensor agents are susceptible, recovered due to virus, recovered due to worm, and recovered due to Trojan horse. Additionally, the malware agents are infectious sensors due to viruses, worms, and Trojan horses. For the WSN environment, a spatially clustered network [38], [39] was employed. The size of the sensors is set at 1.5 to allow visualization. In the MMS, time progresses in distinct time steps known as ticks. However, NetLogo comes with a tick counter that allows the modeler to monitor the number of elapsed ticks while the model runs. Consequently, time (ticks) in this model is measured in weeks. In a scientific modeling context like this, pseudo-random numbers are accepted because they allow the reproducibility of simulation experiments. Furthermore, the mixing that takes place in MMS is heterogeneous, as opposed to the mathematical model, which posits homogeneity. Since the computational model involves interaction between agents, each class was represented by the color type contained in Table 2.

TABLE 2  
COLOR DESCRIPTION IN MMS

Parameters	Color used
Susceptible Sensor nodes	Green
Infected with worm	Red
Infected with virus	Blue
Infected with Trojan	Yellow
Worm resistant	Orange
Virus resistant	Brown
Trojan resistant	Pink

This simulator mimics closely the multigroup SIJRjTj-S model for a WSN. For simplicity, of the nodes could exist in any of the 7 states: susceptible, infected with a worm, infected with a virus, infected with a Trojan, resistant to a worm, resistant to a virus, and resistant to a Trojan. In scientific literature, this type of model is known as a SIR multigroup framework for disease epidemics. At each tick, the infected node tries to contaminate its neighboring nodes. Susceptible neighboring nodes (with green color) will become infected with a probability determined by the Worm-Spread-Chance, Virus-Spread-Chance, and Trojan-Spread-Chance sliders. Infective sensors are not instantly aware of the infection they carry, but with the scan frequencies, the sensor nodes are checked for any deviation from normalcy. This check is often determined by the Worm-Check-Frequency, Virus-Check-Frequency, and Trojan-Check-Frequency sliders. This corresponds to a regularly scheduled scan procedure of the sensor field using drones [40] or Unmanned Aerial Vehicles (UAVs) [41], [42], which distribute patches and virtual vaccines through a cure diffusion scheme. Once the malware variant is identified, there is some likelihood that the infection will be eliminated (as decided by the values of the Worm Recovery Chance, Virus-Recovery-Chance, or Trojan-Recovery-Chance sliders). If a sensor node recovers, there is a likelihood that it might acquire resistance to the various future infections as a result of node vaccination (given by the Wresistance-Prob, Vresistance-Prob, and

Tresistance-Prob sliders). Once a sensor becomes resistant, the linkages between it and other neighboring nodes darken because they are no longer viable conduits for transmitting malware infection.

Under agent model analysis and design, the A<sup>2</sup>CDSADM advocated the inclusion of Unified Modeling Language (UML), algorithms, pseudocode (flowcharts), graphical user interface (GUI), and layout design as well as the building of the data dictionary.

A. Unified Modeling Language (UML)

The use of Object-Oriented Programming (OOP) principles has been agreed upon (graphically illustrated using UML diagrams) [43], [44], [45], and this is primarily because it provides an ideal foundation for ABM implementation. The use of UML diagrams in modeling WSN has long been established in literature [46], [47], [48]. With the class UML diagrams of Fig. 4, the multigroup modeling in the agent-based context is depicted. Note that for ABM, class diagrams represent agents, their world (environment), and the relationships that exist therein.

The arrows in the diagram (Fig. 4) are explained as follows: Solid line arrows depict the association and direct connection between classes, i.e., between NetLogo World and WSN, between WSN and Sensor Node, as well as between WSN and Agent. Dashed arrows depict the dependency between sensor nodes and malware agents, and the Observer depends on the WSN for agent and sensor monitoring. Moreover, a dependency relationship exists between the Observer and the Agent as well as between the WSN and Agent. Using the empty arrowhead, inheritance is depicted between the parent class (Agent) and subclasses, which are malware agents (worm, virus, and Trojan horse). Aggregation can also be seen between NetLogo World and WSN, as well as between the WSN class and the Sensor Node class. While there is a strong composition relationship between Observer and the NetLogo World, i.e., the former owns and controls the operations of the latter.

Note that since the individual malware classes inherit from the Agent class, the implication is that by this dependency relationship, the sensor node interacts with the subclasses indirectly. The addition of separate dependency arrows between the Sensor node class and the three malware agents clutters the diagram and will not add new information. Based on Liskov's Substitution Principle, the explicit dependency arrows might lead to duplicate data and should be avoided at all costs.

B. Algorithm

This subsection presents the pseudocode for the implementation of MMS. As Nwokoye and Umeh [22] put it, "including UML and algorithms (either as pseudocodes or flowcharts) in agent-based modeling aids faster and accurate design of the GUI controls". Below is the pseudocode:

- Declare local variables for turtles (sensors, viruses, worms, and Trojans).
- Declare global variables for the observer, the turtles (sensors, viruses, worms, Trojans), and patches.
- Declare setup procedures.
- Set up sensors, spatially-clustered-network and links.
- Declare Go procedures.
- Input procedure to become-worminfected; set color red.

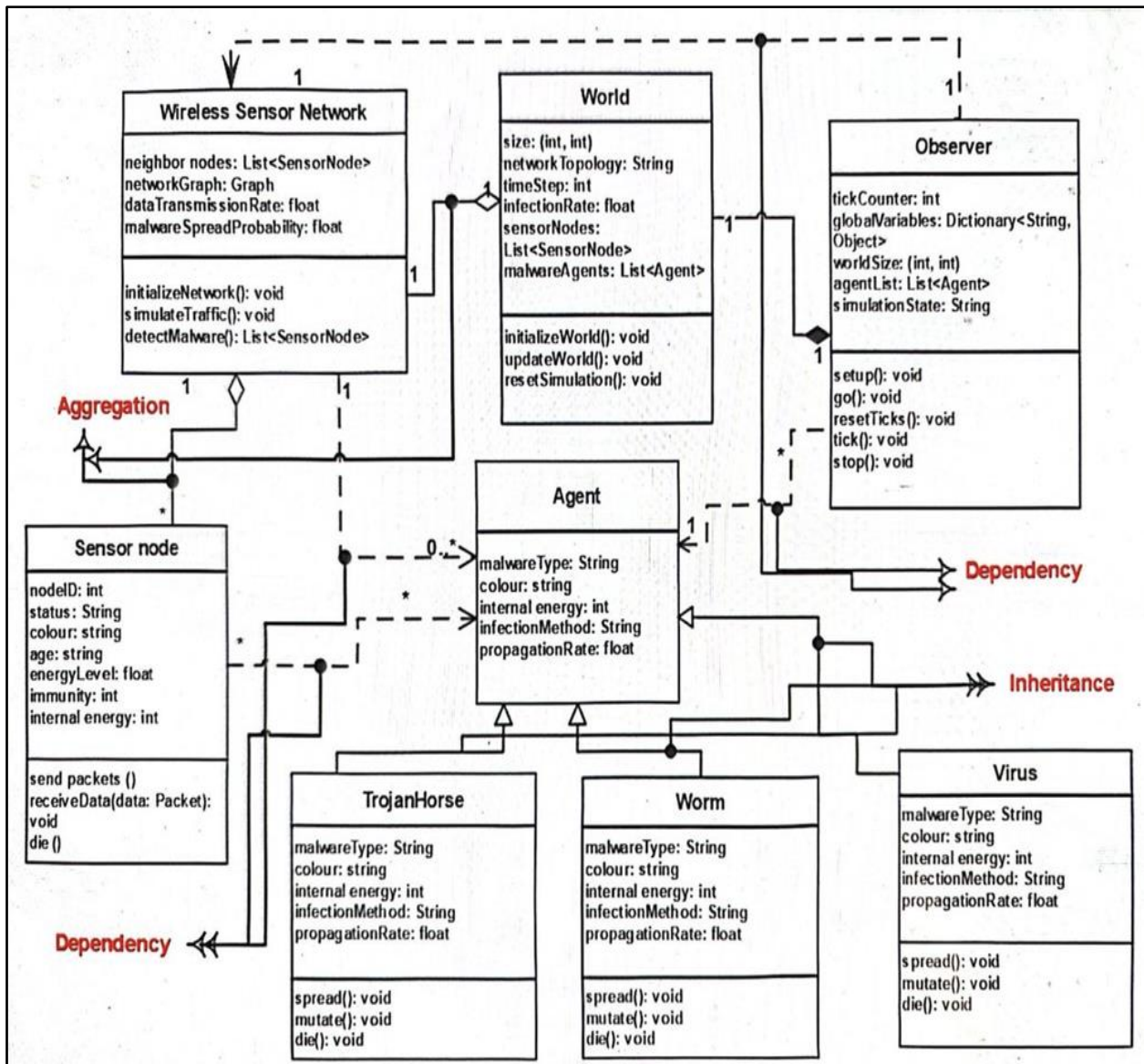


Fig. 4. UML Class diagram for MMS in NetLogo

- Input procedure to become-virusinfected; set color blue.
- Input procedure to become-Trojaninfected; set color yellow.
- Input procedure to become-susceptible; set color green.
- Input procedure to become-wormresistant; set color orange.
- Input procedure to become-virusresistant; set color brown.
- Input procedure to become-Trojanresistant; set color pink.
- Input procedure to spread-worm, spread-virus, spread Trojans.
- Input procedure to do-worm-checks, to do-virus-checks, do-Trojans-checks.

C. Design graphical user interface (GUI) controls and layout

The GUI layout was designed by adding widgets that are most appropriate for the desires and goals of the interaction between the agent and the environment. It is noteworthy that the available widgets in the NetLogo agent toolbox are used

to determine inputs and outputs. Specifically, inputs are buttons, sliders, switches, and choosers, whereas outputs are plots and monitors. Subsequently, using agent-oriented programming, codes will be implemented in NetLogo.

D. Build Data Dictionary

The data dictionary (DD) provides details about each feature of the simulation model (MMS). The features are in the form of sliders, buttons, monitors, plots, and choosers, which are basically NetLogo widgets. DD is of immense importance to programmers or other enthusiasts who may wish to extend the model. The definitions of certain model features are presented in Table 3 below. The attributes have several lengths: number-of-sensors (1000) and buttons, such as design network setup (DNS) and simulate (SIM), cannot have lengths. Worm-initial-outbreak (WIO), virus-initial-outbreak (VIO), and Trojan-initial-outbreak (TIO) have lengths of 100 each; worm-scan-frequency (WSF), virus-scan-frequency (VSF), and Trojan-scan-frequency (TSF) have lengths of 20 each; and worm-recovery-chance (WRC), virus-recovery-chance (VRC), and Trojan-recovery-chance

(TRC) have lengths of 10. Finally, Wresistant-prob (WRP), Vresistant-prob (VRP), and Tresistant-prob (TRP) have lengths of 100. On the widgets, all the attributes are sliders except the DNS and SIM, which are buttons.

TABLE 3  
DATA DICTIONARY FOR MMS

Widget	Type	Description	
NOS	Slider	Number	The number of sensor nodes to be deployed for epidemic analysis.
DNS	Button	N/A	This button activates a sequence of procedures that setup the WSN.
SIM	Button	N/A	Activates several procedures that run iteratively until the model is stopped.
WIO	Slider	Number	Determines the number of worm-infected sensors used for model runs.
VIO	Slider	Number	Determines the number of virus-infected sensors used for model runs.
TIO	Slider	Number	Determines the number of trojan-sensors used for model runs.
WSF	Slider	Tick (Number)	Continuous worm scan for infected sensor nodes.
VSF	Slider	Ticks (Number)	Continuous virus scans for infected sensor nodes.
TSF	Slider	Tick (Number)	Continuous Trojan scans for infected sensor nodes.
WSC	Slider	%age	Probability of worm spread in the network.
VSC	Slider	%age	Probability of virus spread in the network.
TSC	Slider	%age	Probability of Trojan spread in the network.
WRP	Slider	Number	Probability that a sensor would be resistant to future worm infection.
VRP	Slider	Number	Probability that a sensor would be resistant to future virus infection.
TRP	Slider	Number	Probability that a sensor would be resistant to future Trojan infection.

VI. NUMERICAL SIMULATION

This is the concrete realization of the earlier formulations/specifications, and the implication is that reification here demands agent-oriented programming, where the analyst seeks to fix (with a defined syntax) the mental states of the agents to include abstract elements like convictions (concerning their environment, themselves, and each other), potentials, capabilities, and resolutions. Put another way, actual coding during implementation largely means, “the modeler writes rules/instructions, according to the syntax of the NetLogo toolkit, that animate the earlier specifications made” [22]. The NetLogo language is used to model complex scenarios/phenomena evolving through the emergent behavior of interacting agents. On the other hand, the mathematical model was implemented with the Python programming language, i.e., solved using the Runge-Kutta-Fehlberg order 4 and 5 (RK45) numerical method. This method allows the numerical simulation of the BAM. The two approaches represent two distinct scenarios.

A. Simulation using NetLogo Agent Model

At this point, some kind of sensitivity analysis was performed, and this is a kind of parameter-space exploration that concentrates on how the model reacts to variations in the input variables [49], [50]. Here, the aim is to pinpoint the effect of varying the model inputs. Besides equilibrium and stability analyses, numerical simulations have been grossly used to understand and highlight dynamical behaviors of mathematical models [51]–[63]. To perform simulations

aimed at generating the plots below, the analyst should operate the MMS using values contained in the following tables. Table 4 contains the values for simulating scan frequency in the MMS. Additionally, Table 5 contains the values for simulating the impact of recovery and resistance probabilities. While Table 6 contains the values for simulating the impact of spread chance and initial outbreak size in MMS.

TABLE 4  
IMPACT OF SCAN FREQUENCY

Simulator Parameters	Simulation 1	Simulation 2
WSF	20.00 ticks	4.00 ticks
WSC	0.90%	0.90%
WRC	4.30	4.3.00
WRP	100.00	100.00

TABLE 5  
IMPACT OF RECOVERY AND RESISTANCE PROBABILITIES

Parameters	Simulation 1	Simulation 2
NOS	325.00 nodes	325.00 nodes
WIO	6.00 nodes	6.00 nodes
VIO	6.00 nodes	6.00 nodes
TIO	6.00 nodes	6.00 nodes
WSF	9.00 ticks	9.00 ticks
VSF	9.00 ticks	9.00 ticks
TSF	9.00 ticks	9.00 ticks
WSC	5.00	5.00
VSC	5.00	5.00
TSC	5.00	5.00
WRC	4.60	5.00
VRC	0.00	5.00
TRC	0.00	0.00
WRP	100.00	100.00
VRP	0.00	100.00
TRP	0.00	0.00

TABLE 6  
IMPACT OF SPREAD CHANCE AND INITIAL OUTBREAK SIZE

Parameters	Simulation Run for Spread Chance	Simulation Run for Initial Outbreak Size
NOS	726.00 nodes	726.00 nodes
WIO	6.00 nodes	20.00 nodes
VIO	6.00 nodes	6.00 nodes
TIO	6.00 nodes	6.00 nodes
WSF	9.00 ticks	9.00 ticks
VSF	9.00 ticks	9.00 ticks
TSF	9.00 ticks	9.00 ticks
WSC	7.00	5.00
VSC	10.00	5.00
TSC	7.00	5.00
WRC	5.00	5.00
VRC	5.10	5.10
TRC	5.00	5.00
WRP	0.00	0.00
VRP	0.00	0.00
TRP	0.00	0.00

Here, simulation experiments were performed on three malicious code types, namely worms, viruses, and Trojan horses. Using the widgets (sliders), values corresponding to the tables above were set and made ready for simulation tests. Reducing the worm-scan-frequency and the worm recovery chance and increasing the worm spread chance cause the worm to spread, as seen in Fig. 5. Keeping the values that gave rise to Fig. 5 constant and doing the same to corresponding virus settings gave rise to Fig. 6, where favorable values allowed the proliferation of worms and virus infections in the network.

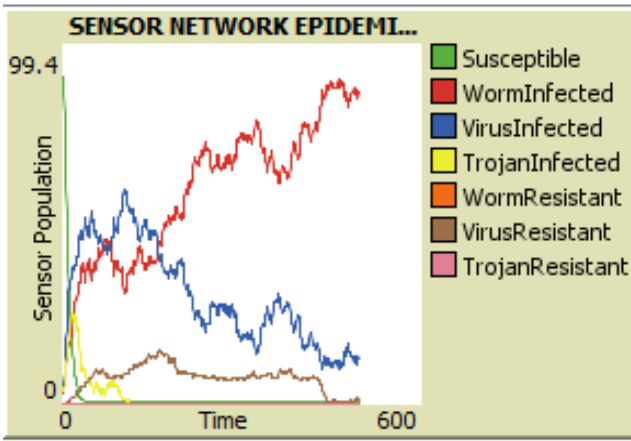


Fig. 5. Worm increase

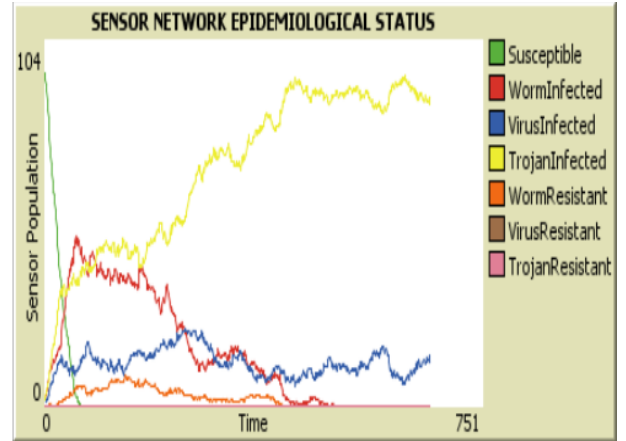


Fig. 9. Impact of recovery and resistance probabilities (1)

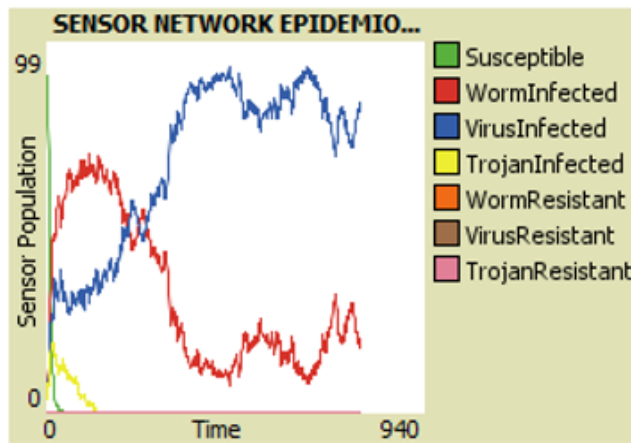


Fig. 6. Worm/Virus increase

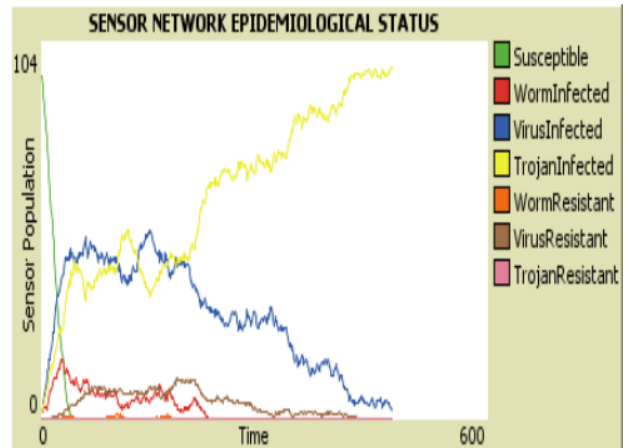


Fig. 10. Impact of recovery and resistance probabilities (2)

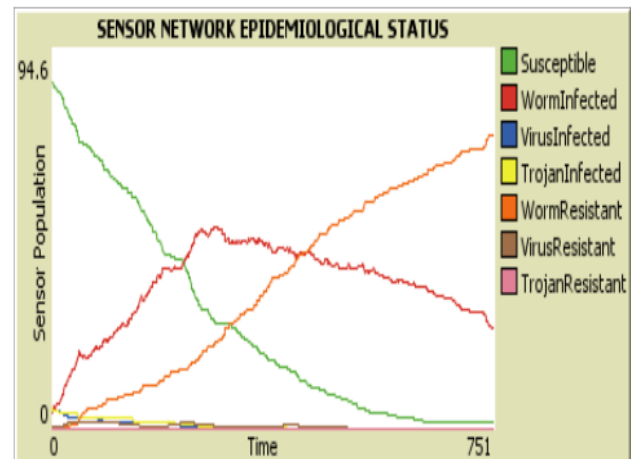


Fig. 7 Impact of scan frequencies (20 ticks)

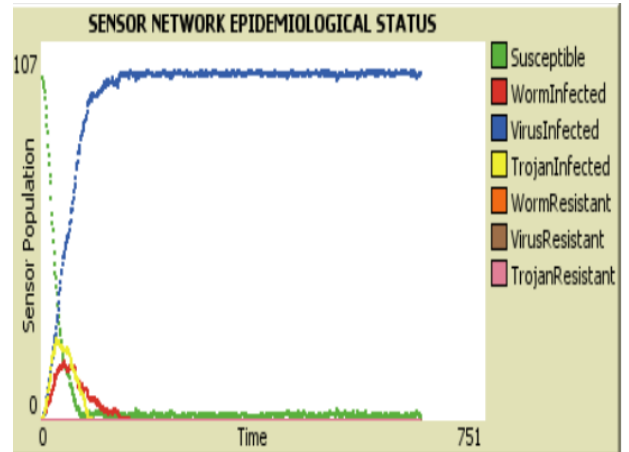


Fig. 11. Impact of spread chance/probability

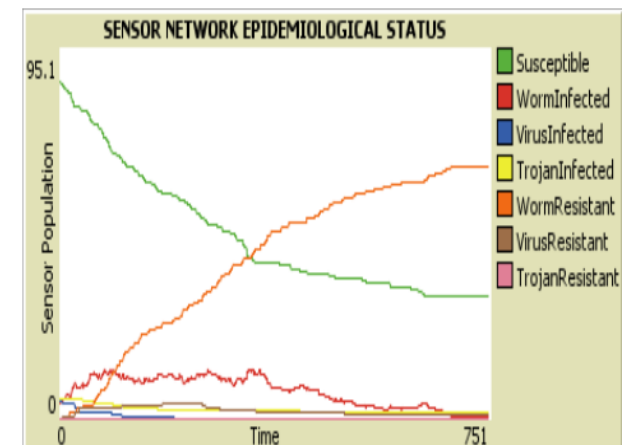


Fig. 8 Impact of scan frequencies (4 ticks)

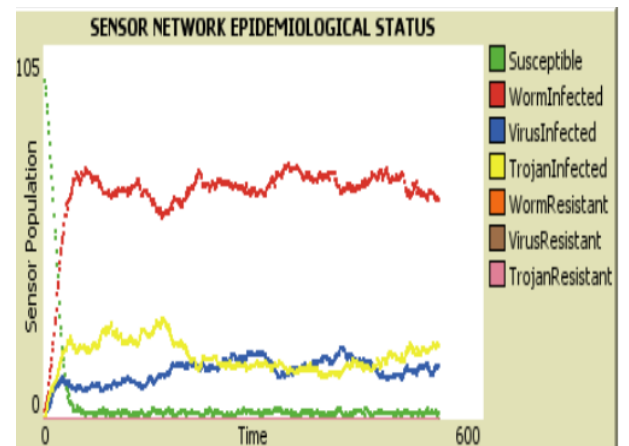


Fig. 12. Impact of initial outbreak size

By analyzing Figs. 7 and 8, it was evident that with low values for scan frequency, worm spread was curtailed. On the other hand, with higher values (such as 20 ticks), there was higher worm spread even though the recovery chance from worm attack was high. Therefore, less time in between scans can aid in the elimination of malicious code in the WSN.

The impacts of recovery and resistance probabilities for MMS are depicted in Fig. 9 and Fig. 10. Keeping all MMS parameters constant at values (in Table 5) except worm-recovery-chance and worm-resistance-probability, which are placed at 4.6 and 100, respectively, showed an initial increase in the number of worm infections, then an increase in the number of worm-resistant sensor nodes. However, they all dipped to zero because the settings subsequently allowed for an increase in Trojan-infected sensor nodes.

In Fig. 10, the recovery chances and resistant probabilities of not only worms but also viruses were increased (as in Table 5). The result shows that virus-resistant sensor nodes also increased alongside the worm-resistant sensors but quickly dipped to zero due to a lack of any recovery/resistance measure for the increase in Trojans in the wireless sensor network. Fig. 11 and Fig. 12 show the impact of spread chance/probability and initial outbreak size (Table 6). While the former shows the increase in viruses as a result of increasing the virus-spread-chance to 10.0, the latter shows the increase of worms as a result of increasing the initial-outbreak size to 20.

### B. Simulation using the Mathematical Model

The implementation of the mathematical model was done in the Python language using several libraries. The libraries utilized in conducting simulations using the model include NumPy, Matplotlib, and SciPy. These are core Python resources for visualization and numeric calculations. NumPy can be used to do efficient manipulation and mathematical calculations, as well as to handle massive datasets along with matrix computations that are required for the analysis of epidemic scenarios. In the implementation, the `solve_ivp` function, which is part of the SciPy library, is used to perform numerical integration of the system of ODEs that characterize the model's classes or compartments. It allows you to choose between several solvers and uses adaptive time-stepping, thus making it perfect for solving complicated, dynamic systems. Additionally, the RK45 solver used in the simulations was applied with the help of the `solve_ivp` function. Lastly, Matplotlib was employed to create visualizations of the simulation outcomes, including compartmentalized time history plots, that are critical for evaluating and conveying the model's conclusions. These libraries collaborate in offering an effective and straightforward method for model development and evaluation of dynamic systems in network epidemiology.

Simulation experiments (cases 1 and 2) were performed using the following initial values for the WSN: Susceptible ( $S = 1500$ ), Infected ( $I_1 = 750, I_2 = 730, I_3 = 700$ ), Recovered ( $R_1 = 0, R_2 = 0, R_3 = 0$ ) and Resistant ( $T_1 = 0, T_2 = 0, T_3 = 0$ ). Note that the following were kept constant during these simulations:  $\mu, \eta, \sigma, r, v_1, v_2, v_3$ , and  $e$ . The parameters:  $\mu, \eta, \sigma, r, \beta_1, \beta_2, \beta_3, \gamma_1, \gamma_2, \gamma_3, v_1, v_2, v_3, \lambda_1, \lambda_2, \lambda_3, e$ . The corresponding values for these parameters for case 1 (weak

defence strategy) are as follows: [5.00, 0.10, 0.10, 1.00, 0.08, 0.05, 0.04, 0.06, 0.05, 0.04, 0.20, 0.20, 0.20, 0.79, 0.57, 0.13, 0.1]. The corresponding values for these parameters for case 2 (strong defence strategy) are as follows: [5.00, 0.10, 0.10, 1.00, 0.40, 0.40, 0.3, 0.98, 0.94, 0.90, 0.2, 0.2, 0.2, 0.82, 0.86, 0.89, 0.1].

Notice that in case 1, the infectious rates were increased from very low ( $\beta_1=0.08, \beta_2=0.05, \beta_3=0.04$ ) to low ( $\beta_1=0.40, \beta_2=0.40, \beta_3=0.30$ ). In case 2, to ensure a strong defence strategy for the sensors, the infection rates were not as low as that of case 1 while the recovery rates were increased i.e., recovery rates for case 1 ( $\gamma_1=0.06, \gamma_2=0.05, \gamma_3=0.04$ ) and recovery rates for case 2 ( $\gamma_1=0.98, \gamma_2=0.94, \gamma_3=0.90$ ). For both cases, the rate of resistance was increased from very low (0.79, 0.57, 0.13) to just low (0.82, 0.86, 0.89).

Fig. 13 represents the time history of case 1, while Fig. 14 represents the time history of case 2. Fig. 15 represents the susceptible and infected populations for case 1, while Fig. 16 shows susceptible and infected populations for case 2. Fig. 17 depicts susceptible and recovered populations for case 1, whereas Fig. 18 shows susceptible and recovered populations for case 2. Fig. 19 shows susceptible and resistant populations for case 1, whereas Fig. 20 shows susceptible and resistant populations for case 2. Figs. 15 to 20 are 3-dimensional depictions of both cases.

Comparing case 1 and case 2, it is evident that in the latter, the resistant class persisted longer than every other class. Conversely, in case 1, the worm infection persisted longer than in every other compartment. In summary, the persistence of the resistant class is more beneficial for the sensor network because nodes possess the required security for the transfer of data and information. With the persistence of infectious compartments, it implies threat survival for a longer period of time. Discussions concerning the computational overhead of sustained threat survival are presented below.

Simulation experiments (for cases 3, 4, and 5) were performed using the following initial values for the WSN: Susceptible ( $S = 1500$ ), Infected ( $I_1 = 750, I_2 = 730, I_3 = 700$ ), Recovered ( $R_1 = 0, R_2 = 0, R_3 = 0$ ), and Resistant ( $T_1 = 0, T_2 = 0, T_3 = 0$ ). The parameters:  $\mu, \eta, \sigma, r, \beta_1, \beta_2, \beta_3, \gamma_1, \gamma_2, \gamma_3, v_1, v_2, v_3, \lambda_1, \lambda_2, \lambda_3, e$ . The corresponding values for these parameters for case 3 are as follows: [5, 0.1, 0.1, 1, 0.08, 0.05, 0.04, 0.98, 0.94, 0.90, 0.2, 0.2, 0.2, 0.92, 0.96, 0.99, 0.1]. The corresponding values for these parameters for case 4 are as follows: [5, 0.1, 0.1, 1, 0.08, 0.05, 0.04, 0.98, 0.94, 0.90, 0.2, 0.2, 0.2, 0.79, 0.57, 0.13, 0.1]. The corresponding values for these parameters for case 5 are as follows: [5, 0.1, 0.1, 1, 0.08, 0.05, 0.04, 0.98, 0.94, 0.90, 0.2, 0.2, 0.2, 0.60, 0.70, 0.75, 0.1]. The corresponding values for these parameters for case 6 are as follows: [5, 0.1, 0.1, 1, 0.4, 0.4, 0.3, 0.98, 0.94, 0.90, 0.2, 0.2, 0.2, 0.60, 0.70, 0.75, 0.1]. Fig. 21, Fig. 22, Fig. 23, and Fig. 24 are time histories of cases 3, 4, 5, and 6, respectively.

From the graphical illustrations of transmission dynamics in the WSN, it is evident that due to the high proportions of malware infections, especially for virtual worms, they seem to persist for a prolonged period. Notice that in case 3, the sensors with worm infections are slightly above the sensors that have recovered from the infection. Further analyses of cases 4 and 5 show that as the parameters for resistance were lowered, the sensors with worm infections started to recover.

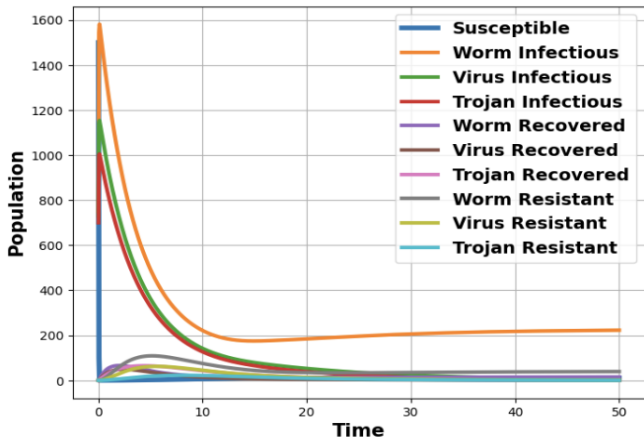


Fig. 13. Time history of the classes for case 1 (weak defence strategy)

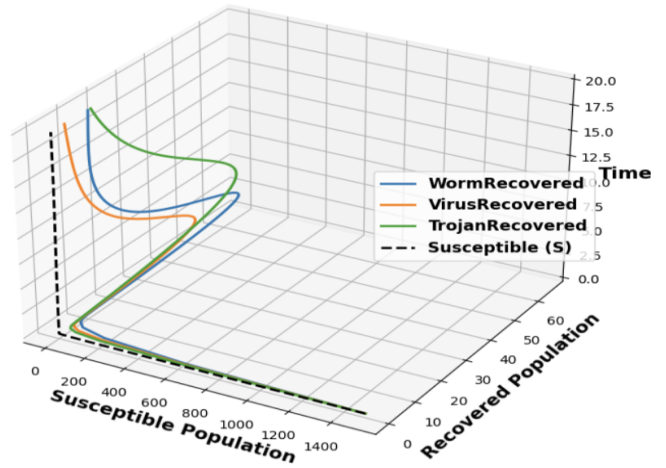


Fig. 17. Susceptible and recovered populations for case 1

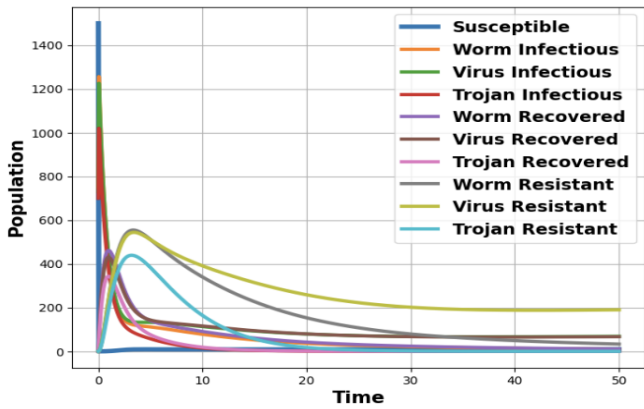


Fig. 14. Time history of the classes for case 2 (strong defence strategy)

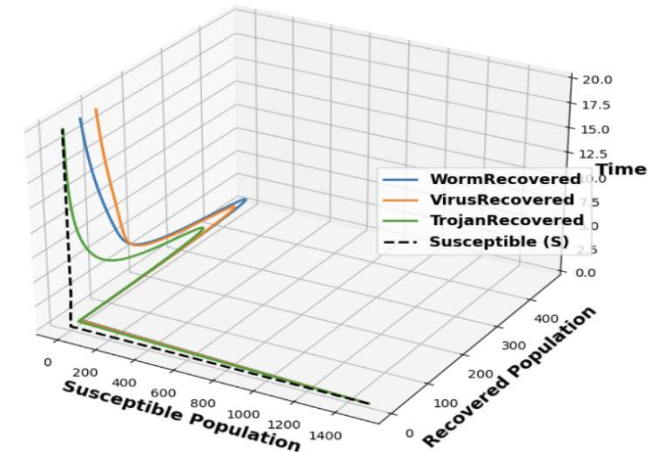


Fig. 18. Susceptible and recovered populations for case 2

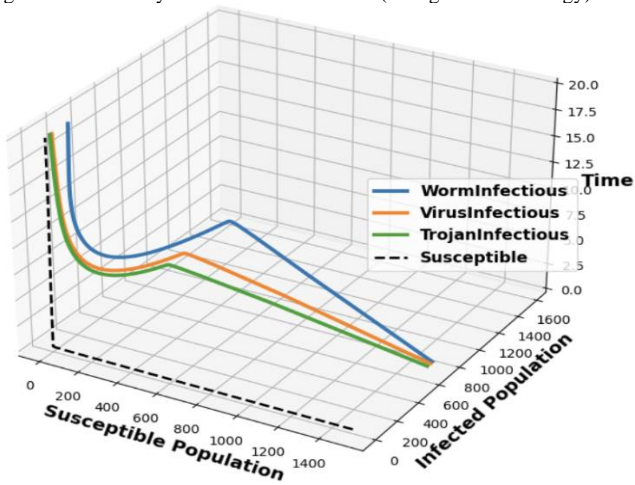


Fig. 15. Susceptible and infected populations for case 1

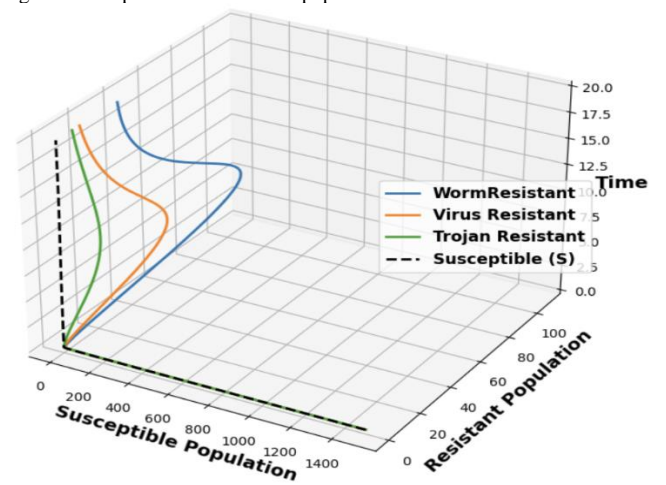


Fig. 19. Susceptible and Resistant populations for case 1

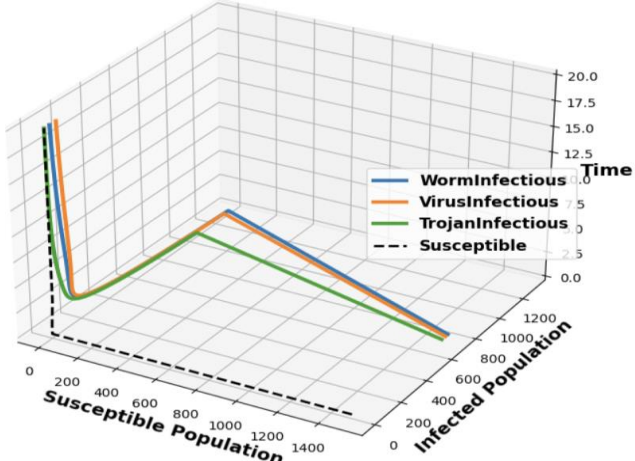


Fig. 16. Susceptible and infected populations for case 2

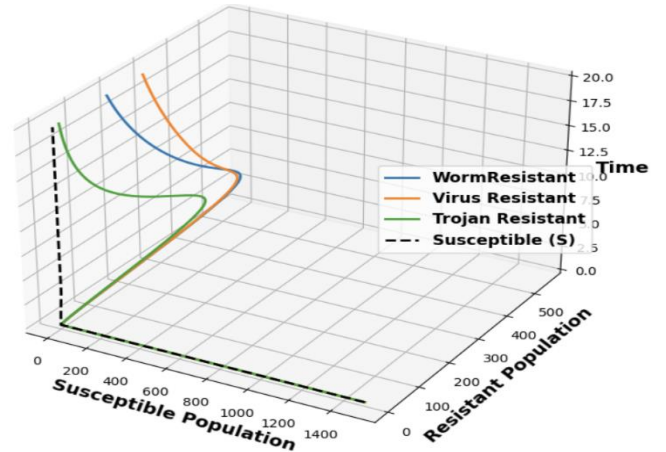


Fig. 20. Susceptible and Resistant populations for case 2

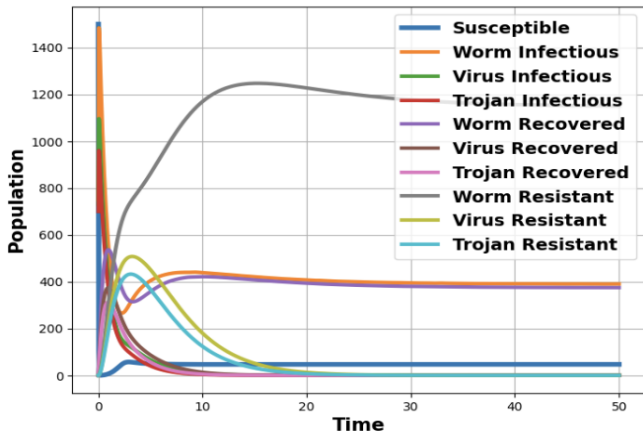


Fig. 21. Time history of the classes for case 3

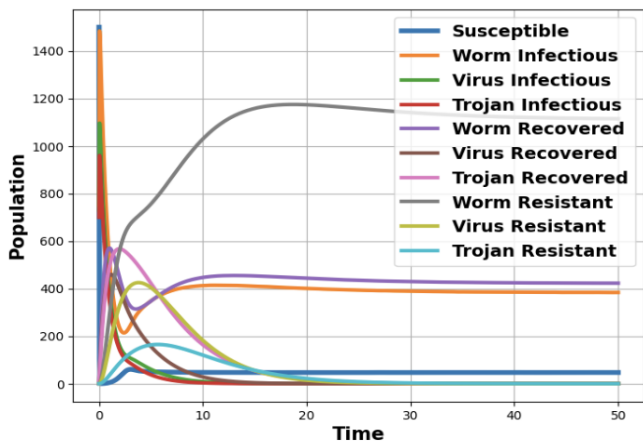


Fig. 22. Time history of the classes for case 4

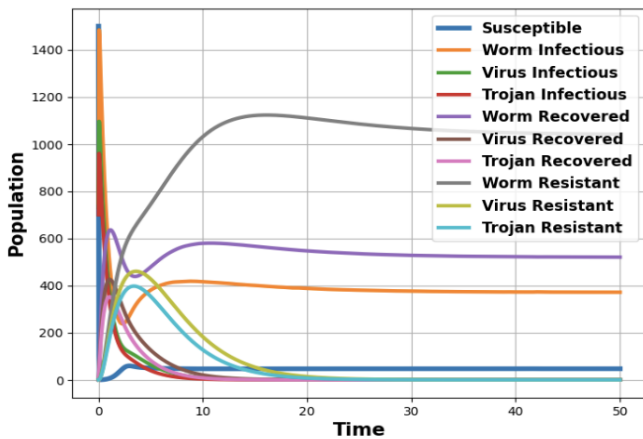


Fig. 23. Time history of the classes for case 5

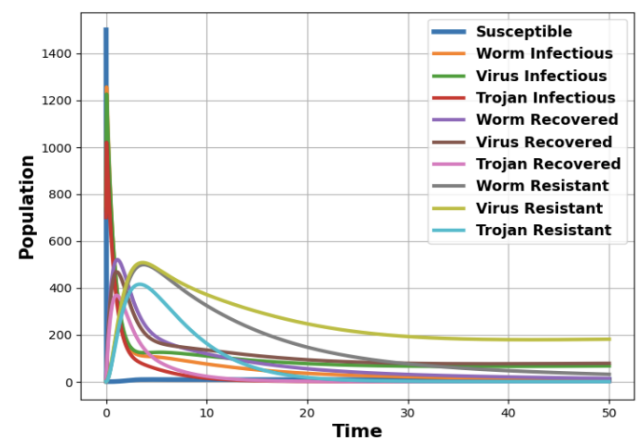


Fig. 24. Time history of the classes for case 6

With the same resistance parameter values (0.60, 0.70, 0.75), case 6 started to bear a striking resemblance to case 2. In fact, case 6 even has higher rates of recovery than case 2. In other words, case 6 can also be referred to as another, stronger defense strategy. However, it should be noted that both cases (2 and 6) have the same infectious parameter values (i.e., 0.40, 0.40, 0.30). Therefore, network security managers should aim to identify measures that mimic cases 2 and 6.

C.Reproduction Ratios ( $R_0$ ) and Long-term Behavior

The reproduction ratio, or epidemic threshold, is a fundamental parameter for quantifying the potential proliferation of an infection in a system (WSN). It reflects the projected number of subsequent infections resulting from an infected host in a vulnerable sensor population. The aim of this analysis is to show that while  $R_0$  has some advantages, it can be both confusing and misleading for network security managers. By employing the equation, the obtained values demonstrate the contrasted effects of weak (or ineffective) and strong defense approaches.

Parameters and compartment values: The variables used in the computation represent the effect of various defensive techniques. At first, low infection rates ( $\beta_1=0.08, \beta_2=0.05, \beta_3=0.04$ ) were chosen for the inadequate and weak defense approach, along with modest recovery rates ( $\gamma_1=0.06, \gamma_2=0.05, \gamma_3=0.04$ ). On the contrary, a robust and strong defense plan included higher rates of recovery ( $\gamma_1=0.98, \gamma_2=0.94, \gamma_3=0.90$ ) and resistance (0.82, 0.86, 0.89) to curb incidences of malware spread resulting from high rates of infection ( $\beta_1=0.40, \beta_2=0.40, \beta_3=0.30$ ). These infection rates are high compared to infection rates of case 1 (weak defense strategy). The values for the compartments are as follows: Susceptible ( $S=1500$ ), Infected ( $I_1=750, I_2=730, I_3=700$ ), Recovered ( $R_1=0, R_2=0, R_3=0$ ) and Resistant ( $T_1=0, T_2=0, T_3=0$ ). Table 7 contains the results obtained when computing the reproduction ratios of the cases above. Note that the actual values for the  $R_0$ s for cases 3, 4, and 5 are 0.0465, but in the table the approximate value (0.05) was used.

TABLE 7  
 $R_0$  COMPUTATION FOR SEVERAL CASES

Cases	Infectious	Recovery	Resistant	$R_0$
1	0.08, 0.05, 0.04	0.06, 0.05, 0.04	0.79, 0.57, 0.13	0.21
2	0.40, 0.40, 0.30	0.98, 0.94, 0.90	0.82, 0.86, 0.89	0.30
3	0.08, 0.05, 0.04	0.98, 0.94, 0.90	0.92, 0.96, 0.99	0.05
4	0.08, 0.05, 0.04	0.98, 0.94, 0.90	0.79, 0.57, 0.13	0.05
5	0.08, 0.05, 0.04	0.98, 0.94, 0.90	0.60, 0.70, 0.75	0.05
6	0.40, 0.40, 0.30	0.98, 0.94, 0.90	0.60, 0.70, 0.75	0.05

In the context of case 1 (weak defense approach), the computed reproduction ratio is around  $R_0=0.21$ . The lower  $R_0$  reflects the infection's restricted spread due to lower rates of infection ( $\beta_1, \beta_2, \beta_3$ ) and modest recovery rates. With moderate recovery rates, which are lower compared to that

of the rates of infection in case 2 (strong defensive approach), low rates of infection minimize the risk of subsequent infections. Besides the low infectious and low recovery rates, this case/strategy was considered weak because it is evident that worm infections persisted, and this is not good for the sensors.

For case 2 (i.e., robust defense approach), the  $R_0$  rises to 0.30, indicating a larger risk of malware infection, which is subsequently curbed through mechanisms that enable higher recovery and resistance in the network. The greater  $R_0$  is due to much higher infection rates ( $\beta_1, \beta_2, \beta_3$ ), which outweigh the impact of improved recovery dynamics. This case/strategy was considered strong because it is evident that worm infections persisted, and this is not good for the sensors. This study shows that while increased infection capability may reduce its efficacy in preventing disease spread, a stronger defensive strategy should include resilient recovery and resistant processes for vulnerable sensors. These findings are especially important for planning interventions. It shows that, although speed of recovery is very important, managing the propagation dynamics (through the infectivity parameters) may have a greater influence on lowering and, hence, spreading infection.

Low  $R_0$  values are desirable, as in the case of 1 to 5, because they may imply the production of fewer secondary infections, easy-to-control systems, lower allocation of resources, system stability, effectiveness of remedial measures, and the potential eradication of the threat. However, the resulting  $R_0$  for cases 3, 4, and 5, where the epidemic is 0.05, is truly misleading, as the formula does not account for changes in the resistant compartments, i.e., the equations aimed at oversimplification, thereby excluding the parameters for resistance. In other words, the  $R_0$  is showing an easy-to-manage system, and this may not be the case in the long run. Other reasons why reproduction ratios are misleading include issues of data quality, misinterpretation of results, dependence on obsolete interventions due to past low  $R_0$  values, and the inability to capture long-term trends, etc.

Therefore, due to these issues, it is absolutely necessary to modify the  $R_0$  with considerations for the dynamics of resistance and to understand long-term behavioral patterns in the WSN. The modified epidemic threshold for the mathematical model is as follows:

$$R_0^* = \sum_{j=1}^3 \left( \frac{\pi^2 \sigma \beta_j}{e + \eta + \gamma_j + v_j} + \frac{\pi^2 \sigma \lambda_j \gamma_j}{(e + \eta + \gamma_j)(\lambda_j + \eta)(v_j + \eta)} \right) \quad (18)$$

Note that this modified reproduction number was derived by substituting the steady state of the resistant classes ( $T_j$ ) into the equation for the recovered compartments. Next, find the equilibrium of the recovered compartments ( $R_j$ ) and then substitute it into the steady state of resistant classes ( $T_j$ ). This ratio now accounts for the dynamics of the resistant compartment, i.e., both entry (from the recovered compartments) and exit of sensors (back into the susceptible compartment). Computing the new reproduction ratios resulted in the following values contained in Table 8. From this table, it is glaring that the epidemic thresholds of table 7 are misleading. Table 8 depicts the complexity of the system through much higher reproduction numbers.

Incorporating resistance characteristics in the changing dynamics impacts the way  $R_0$  represents the behavior of the system. A substantially elevated  $R_0$  indicates a significant infection pressure, regardless of how resistant compartments

limit infection transmission in the long run. The balance regarding infection along with resistance dictates the network's ability to successfully regulate any outbreak of infection. The substitution of steady-state parameters for recovery and resistance is critical for comprehension of disease persistence and possibilities for subsequent infection waves, particularly in the case of strains that are resistant. This method guarantees that resistance is not just addressed as a short-term effect but also as an integral component of the entire network dynamics, which facilitates an increased understanding of malware growth and control.

TABLE 8  
COMPUTATION FOR THE MODIFIED  $R_0$  FOR SEVERAL CASES

Cases	Infectious	Recovery	Resistant	$R_0$
1	0.08, 0.05, 0.04	0.06, 0.05, 0.04	0.79, 0.57, 0.13	0.61
2	0.40, 0.40, 0.30	0.98, 0.94, 0.90	0.82, 0.86, 0.89	2.58
3	0.08, 0.05, 0.04	0.98, 0.94, 0.90	0.92, 0.96, 0.99	2.38
4	0.08, 0.05, 0.04	0.98, 0.94, 0.90	0.79, 0.57, 0.13	2.03
5	0.08, 0.05, 0.04	0.98, 0.94, 0.90	0.60, 0.70, 0.75	2.30
6	0.40, 0.40, 0.30	0.98, 0.94, 0.90	0.60, 0.70, 0.75	2.51

#### D. Computational Overhead for Network Resources

In the case of WSNs, applying cybersecurity countermeasures to eliminate malware infestations incurs some computational overhead, resulting in extra processing, memory, and energy resources necessary to carry out these procedures. Security procedures, including intrusion detection frameworks, encryption, and infection scanning, sometimes need computing power that surpasses the lightweight abilities of sensor nodes. Such overhead might emerge in a variety of ways, such as elevated processing times, congestion, and enormous utilization of energy and memory during security operations. Considering the cases above along with the values for the reproduction numbers, most especially for cases 3, 4, and 5, the graphs show that the system may have to sustain long-term operation of the mechanisms for sensor recovery and resistance. And this might cause the depletion of the already constrained resources of these battery-powered sensor nodes.

#### E. Modeling Approaches: Pros and Cons

On the pros and cons, both modeling approaches present several benefits along with challenges for modelers and enthusiasts alike. The agent method allows the application of AOP, thus depicting the existence of agents in an environment coded to mimic the actual network in question. This attempt to create a semblance of a sensor network using a spatially clustered arrangement of agents is not very possible with mathematical modeling using the e-SIjRjTj model. Both approaches allow 3D simulation, but with some significant differences. Note that the agent MMS model was built in a 2-dimensional format, and its codes may need to be modified to allow a 3D view. But even if it was programmed for 3D, its display may not look like the 3D phase portraits generated using Python. Checking the stability of the e-SIjRjTj model can be very tough and time-consuming because as the dimensions of a compartmental model are increased, its complexity increases. Instead of the complex and cumbersome stability analyses, the agent model allows

the implementation of sensor node scanning for faster malware detection.

With the agent-based simulator, it is qualitatively evident that reducing the probabilities of spread of one type of malicious code (such as worms) does not reduce the probabilities of spread of other malicious code types existent in the WSN. Additionally, increasing the scan frequency for worms in order to check instances of worm infection does not increase the scan frequency of other types of malicious code in the networks; other malicious code types (with lower scan frequency values) may propagate. The implication here is that the drones/UAVs used for distributing patches to infected nodes should be updated in order to address all existent types of infection in the network. Otherwise, this curative approach may not be entirely beneficial. However, aside from obtaining novelty through the development of this simulator, using this tool will provide even more insights through the setting and resetting of model attributes (sliders, choosers, buttons). The MMS involves variables ("worm-scan-frequency, virus-scan-frequency, and Trojan-scan-frequency") that frequently scan the sensor field for any kind of malicious code infection. These variables seek to detect node malfunctions or slower speeds in data transmission—a known symptom of "exposed" sensor nodes—and protect them. Through drones or UAVs, the distrusted nature of WSNs and their existence in hostile environments no longer become a challenge for network managers who would wish to cure sensor nodes in case they acquire infections.

The analyses and discussions concerning the reproduction ratios also showed that reproduction ratios that capture only infectious and recovery rates are limited and are not suitable for a model that considers mechanisms for making the nodes resistant after recovery. In addition, the discussions on the computational overhead are also very important considering the miniaturized sensors whose functionalities depend on the limited resources. In strong terms, this study advocates for the usage of both approaches so as to achieve immense benefits.

## VII. CONCLUSION AND FUTURE DIRECTIONS

This study characterized multiple malware propagations in WSNs using the e-SIJrTj-S model and an agent computational model (MMS). Also, it describes containment strategies that exploit both modeling techniques. Interestingly, the study highlighted variables that provide more insights in the context of multi-group modeling. In view of the several strengths associated with both modeling approaches, the study advocates their combined use for representing complex real-world phenomena. In addition to the scan mechanisms of the simulators, the reproduction ratios of numerous cases were explicitly calculated, and discussions were provided to understand the import of the applied control strategies. The findings of this research can influence decisions regarding policy and the development of the optimal remedial or treatment strategies in identical networked systems. In the future, the MMS will be extended to include routing protocols, other compartments (exposed, vaccination), and other phenomena existent in a typical network, as well as the unfriendly and inaccessible terrain of WSN. The environment coded in the agent models did not characterize WSN's unfriendly and inaccessible terrain, such as that found in underwater WSN. Furthermore, the multi-strain or staged progression variety of epidemic models will be reified using the MAS. Additionally, the delay analyses and optimal control

implications would also be subsequently studied for the mathematical model.

## REFERENCES

- [1] L. Jaeger, A. Eckhardt, and J. Kroenung, "The role of deterrability for the effect of multi-level sanctions on information security policy compliance: Results of a multigroup analysis," *Information and Management*, vol. 58, no. 3, p. 103–318, 2021. doi: <https://doi.org/10.1016/j.im.2020.103318>. [Online].
- [2] B. Dickson. "Car companies massively exposed to web vulnerabilities." (2023), [Online]. Available: <https://portswigger.net/daily-swig/car-companies-massively-exposed-to-web-vulnerabilities>. (accessed:25.07.2024).
- [3] B. Lenaerts-Bergmans. "Malicious code: What it is and how to prevent it." (2023), [Online]. Available: <https://www.crowdstrike.com/cybersecurity-101/malicious-code/>. (accessed: 29.07.2024).
- [4] Q. Fu, Y. Yao, C. Sheng, and W. Yang, "Interplay between malware epidemics and honeynet potency in industrial control system network," *IEEE Access*, vol. 8, pp. 81 582–81 593, 2020.
- [5] C. H. Nwokoye and V. Madhusudanan, "Epidemic models of maliciouscode propagation and control in wireless sensor networks An in-depth review," *Wireless personal communications*, vol. 125, no. 2, pp. 1827–1856, 2022.
- [6] Q. Yan, L. Song, C. Zhang, J. Li, and S. Feng, "Modeling and control of malware propagation in wireless iot networks," *Security and Communication Networks*, vol. 2021, no. 1, p. 4 133 474, 2021.
- [7] N. P. Dong, N. L. Giang, and H. V. Long, "Interconnected takagi-sugeno system and fractional sirs malware propagation model for stabilization of wireless sensor networks," *Information Sciences*, vol. 670, p. 120 620, 2024.
- [8] Q. Zhu, X. Luo, Y. Liu, C. Gan, Y. Wu, and L.-X. Yang, "Impact of cybersecurity awareness on mobile malware propagation: A dynamical model," *Computer Communications*, vol. 220, pp. 1–11, 2024.
- [9] C. H. Nwokoye, I. I. Umeh, N. N. Mbeledogu, and V. O. Okeke, "Scanbased worms: The impact of ipv4 address space on epidemic computer network models," *Engineering Letters*, vol. 29, no. 2, 2021.
- [10] P. Van den Driessche and J. Watmough, "Reproduction numbers and subthreshold endemic equilibria for compartmental models of disease transmission," *Mathematical biosciences*, vol. 180, no. 1-2, pp. 29–48, 2002.
- [11] J. M. Hyman and J. Li, "Differential susceptibility epidemic models," *Journal of mathematical biology*, vol. 50, no. 6, pp. 626–644, 2005.
- [12] A. M. Mart' in del Rey et al., "A novel model for malware propagation on wireless sensor networks," *Mathematical Biosciences and Engineering*, vol. 21, no. 3, pp. 3967–3998, 2024.
- [13] M. Faris, M. N. Mahmud, M. F. M. Salleh, and A. Alnoor, "Wireless sensor network security: A recent review based on state-of-the-art works," *International Journal of Engineering Business Management*, vol. 15, p. 18 479 790 231 157 220, 2023.
- [14] B. K. Mishra and A. K. Singh, "Sijrs e-epidemic model with multiple groups of infection in computer network," *International journal of nonlinear science*, vol. 13, no. 3, pp. 357–362, 2012.
- [15] B. K. Mishra and G. M. Ansari, "Differential epidemic model of virus and worms in computer network," *Int. J. Netw. Secur.*, vol. 14, no. 3, pp. 149–155, 2012.
- [16] B. K. Mishra and A. Prajapati, "Dynamic model on the transmission of malicious codes in network," *International Journal of Computer Network and Information Security*, vol. 5, no. 10, p. 17, 2013.
- [17] B. K. Mishra, "Mathematical model on attack of worm and virus in computer network," *Int. J. Future Gener. Commun. Netw.*, vol. 9, no. 6, pp. 245–254, 2016.
- [18] R. P. Ojha, P. K. Srivastava, and G. Sanyal, "Pre-vaccination and quarantine approach for defense against worms propagation of malicious objects in wireless sensor networks," in *Sensor Technology: Concepts, Methodologies, Tools, and Applications*, IGI Global, 2020, pp. 1233–1251.
- [19] C. Nwokoye, I. Umeh, and O. Ositanwosu, "Characterization of heterogeneous malware contagions in wireless sensor networks: A case of uniform random distribution," in *ICT Analysis and Applications: Proceedings of ICT4SD 2020, Volume 2*, Springer, 2020, pp. 813–821.
- [20] C. H. Nwokoye, C. Umeugoji, and I. Umeh, "Evaluating degrees of differential infections on sensor networks' features using the sejjr-v epidemic model," *Egyptian Computer Science Journal*, vol. 44, no. 3, 2020.
- [21] B. K. Mishra and I. Tyagi, "Defending against malicious threats in wireless sensor network: A mathematical model," *International Journal of Information Technology and Computer Science*, vol. 6, no. 3, pp. 12–19, 2014.

- [22] C. Nwokoye and I. Umeh, "Analytic-agent cyber dynamical systems analysis and design methodology for modeling temporal/spatial factors of malware propagation in wireless sensor networks," *Methodx*, 2018.
- [23] C. H. Nwokoye, V. Madhusudan, M. Srinivas, and N. Mbeledogu, "Modeling time delay, external noise and multiple malware infections in wireless sensor networks," *Egyptian Informatics Journal*, vol. 23, no. 2, pp. 303–314, 2022, issn: 1110-8665. doi: <https://doi.org/10.1016/j.eij.2022.02.002>.
- [24] 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.
- [25] 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, pp. 1–19, 2017.
- [26] F. K. Batista, A. Martin del Rey, and A. Queiruga-Dios, "A new individual based model to simulate malware propagation in wireless sensor networks," *Mathematics*, vol. 8, no. 3, p. 410, 2020.
- [27] A. Gonzalez, I. Marshall, and L. Sacks, "A self-synchronised scheme for automated communication in wireless sensor networks," in *Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference*, 2004., IEEE, 2004, pp. 97–102.
- [28] M. Niazi and A. Hussain, "Agent-based tools for modeling and simulation of self-organization in peer-to-peer, ad hoc, and other complex networks," *IEEE Communications Magazine*, vol. 47, no. 3, pp. 166–173, 2009.
- [29] F. Albiero, F. H. Fitzek, and M. D. Katz, "Introduction to netlogo," in *Cognitive Wireless Networks: Concepts, Methodologies and Visions Inspiring the Age of Enlightenment of Wireless Communications*, Springer, 2007, pp. 579–602.
- [30] K. Wasti, "Usability of multi-agent simulators in simulation of wireless networks," M.S. thesis, K. Wasti, 2014.
- [31] H. Alharbi and A. Hussain, "An agent-based approach for modelling peer to peer networks," in *2015 17th UKSim-AMSS International Conference on Modelling and Simulation*, IEEE, 2015, pp. 532–537.
- [32] M. Chen, T. Kwon, Y. Yuan, and V. C. Leung, "Mobile agent based wireless sensor networks," *J. Comput.*, vol. 1, no. 1, pp. 14–21, 2006.
- [33] A. M. del Rey, J. H. Guill'en, and G. R. S'anchez, "Modeling malware propagation in wireless sensor networks with individual-based models," in *Advances in Artificial Intelligence: 17th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2016, Salamanca, Spain, September 14-16, 2016. Proceedings 17*, Springer, 2016, pp. 194–203.
- [34] F. K. Batista, 'A. Mart'in del Rey, and A. Queiruga-Dios, "Malware propagation software for wireless sensor networks," in *Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection - 15th International Conference, PAAMS 2017, F. De la Prieta, Z. Vale, L. Antunes, et al., Eds., Cham: Springer International Publishing, 2018, pp. 238–241, isbn: 978-3-319-61578-3*.
- [35] C. H. Nwokoye, N. N. Mbeledogu, R. U. Paul, and C. Ugwunna, "Complementing malware epidemic agent-based models with routing protocols of communication networks using netlogo," in *Information and Communication Technology for Competitive Strategies (ICTCS 2022)*, A. Joshi, M. Mahmud, and R. G. Ragel, Eds., Singapore: Springer Nature Singapore, 2023, pp. 833–847, isbn: 978-981-19-9638-2.
- [36] C. Nwokoye and I. I. Umeh, "The seiqr-v model: On a more accurate analytical characterization of malicious threat defense," *Int. J. Inf. Technol. Comput. Sci.*, vol. 9, no. 12, pp. 28–37, 2017.
- [37] J. A. P. H. O. Diekmann and J. A. J. Metz, "On the definition and the computation of the basic reproduction ratio  $r_0$  in models for infectious diseases in heterogeneous populations," *Journal of Mathematical Biology*, vol. 28, pp. 365–382, 1990.
- [38] I. Akila, S. Manisekaran, and R. Venkatesan, "Modern clustering techniques in wireless sensor networks," *Wireless Sensor Networks-Insights and Innovations*, pp. 141–156, 2017.
- [39] A. I. Al-Sulaifanie, B. K. Al-Sulaifanie, and S. Biswas, "Recent trends in clustering algorithms for wireless sensor networks: A comprehensive review," *Computer Communications*, vol. 191, pp. 395–424, 2022.
- [40] N. R. Zema, E. Natalizio, G. Ruggeri, M. Poss, and A. Molinaro, "Medrone: On the use of a medical drone to heal a sensor network infected by a malicious epidemic," *Ad Hoc Networks*, vol. 50, pp. 115–127, 2016.
- [41] Y. Gao, L. Ren, T. Shi, T. Xu, and J. Ding, "Autonomous obstacle avoidance algorithm for unmanned aerial vehicles based on deep reinforcement learning," *Engineering Letters*, vol. 32, no. 3, 2024.
- [42] G. Liu, B. Peng, X. Zhong, and X. Lan, "Differential games of rechargeable wireless sensor networks against malicious programs based on sildr propagation model," *Complexity*, vol. 2020, no. 1, p. 5 686 413, 2020.
- [43] V. Grimm, U. Berger, F. Bastiansen, et al., "A standard protocol for describing individual-based and agent-based models," *Ecological modelling*, vol. 198, no. 1-2, pp. 115–126, 2006.
- [44] C. M. Macal and M. J. North, "Tutorial on agent-based modeling and simulation part 2: How to model with agents," in *Proceedings of the 2006 Winter simulation conference, IEEE, 2006, pp. 73–83*.
- [45] A. T. Crooks and C. J. Castle, "The integration of agent-based modelling and geographical information for geospatial simulation," in *Agent-based models of geographical systems*, Springer, 2011, pp. 219–251.
- [46] S. Uke and R. Thool, "Uml based modeling for data aggregation in secured wireless sensor network," *Procedia Computer Science*, vol. 78, pp. 706–713, 2016.
- [47] J. Wang, A. O. Fapojuwo, C. Zhang, and H. Tan, "Uml modeling of cross-layer attack in wireless sensor networks," in *Interoperability, Safety and Security in IoT: Second International Conference, InterIoT 2016 and Third International Conference, SaSeIoT 2016, Paris, France, October 26-27, 2016, Revised Selected Papers 2*, Springer, 2017, pp. 104–115.
- [48] S. Teixeira, B. A. Agrizzi, J. G. Pereira Filho, S. Rossetto, and R. de Lima Baldam, "Modeling and automatic code generation for wireless sensor network applications using model-driven or business process approaches: A systematic mapping study," *Journal of Systems and Software*, vol. 132, pp. 50–71, 2017.
- [49] J.-S. Lee, T. Filatova, A. Ligmann-Zielinska, et al., "The complexities of agent-based modeling output analysis," *Journal of Artificial Societies and Social Simulation*, vol. 18, no. 4, 2015.
- [50] B. K. Mishra and A. Prajapati, "Mathematical model on attack by malicious objects leading to cyber war," *International Journal of Nonlinear Science*, vol. 17, no. 2, pp. 145–153, 2014.
- [51] Y. Chong, Q. Zhu, Q. Li, and F. Chen, "Dynamic Behaviors of a Two Species Amensalism Model with a Second Species Dependent Cover," *Engineering Letters*, vol. 32, no. 8, pp1553-1561, 2024.
- [52] S. Lo, and C. Lin, "Stability Analysis of Dynamic Competition-Cooperation Model with Economic Factors for Transportation Systems," *Engineering Letters*, vol. 32, no. 7, pp1424-1435, 2024.
- [53] Z. Bai, X. Zhao, H. Song, H. Qin, Y. Zhang, and H. Yao, "Numerical Simulation Study on Ventilation Effect in Utility Tunnel," *Engineering Letters*, vol. 32, no. 6, pp1090-1096, 2024.
- [54] Y. Chong, Y. Hou, S. Chen, and F. Chen, "The Influence of Fear Effect to the Dynamic Behaviors of Lotka-Volterra Ammensalism Model," *Engineering Letters*, vol. 32, no. 6, pp1233-1242, 2024.
- [55] D. Li, X. Zhao, Z. Zhao, C. Su, and J. Meng, "Stability Analysis of the Floating Multi-robot Coordinated Towing System Based on Ship Stability," *Engineering Letters*, vol. 32, no. 6, pp1191-1200, 2024.
- [56] J. Zhang, Z. Zhang, C. Zhou, and X. Ma, "A New SIQR Model and Residual Power Series Method in Wireless Sensor Networks," *IAENG International Journal of Computer Science*, vol. 51, no. 9, pp1240-1249, 2024.
- [57] A. K. Alomari, and Y. Massoun, "Numerical Solution of Time Fractional Coupled Korteweg-de Vries Equation with a Caputo Fractional Derivative in Two Parameters," *IAENG International Journal of Computer Science*, vol. 50, no.2, pp388-393, 2023.
- [58] Y. Liu, and W. Sun, "Stability of Riemann Solutions for the Hyperbolic System," *IAENG International Journal of Computer Science*, vol. 50, no.2, pp683-687, 2023.
- [59] L. Fei, and H. Lv, "Multi-Host Transmission Dynamics of Schistosomiasis and Effective Control," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 11, pp2316-2329, 2024.
- [60] L. Xu, Y. Xue, Q. Lin, and F. Chen, "Stability and Bifurcation Analysis of Commensal Symbiosis System with the Allee Effect and Single Feedback Control," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 8, pp1586-1596, 2024.
- [61] R. Ramesh, and G. Arul Joseph, "The Optimal Control Methods for the Covid-19 Pandemic Model's Precise and Practical SIQR Mathematical Model," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 8, pp1657-1672, 2024.
- [62] Y. Chang, and K. Hsu, "Mathematical Modeling and Analysis of Coupled-Inductor Cockcroft-Walton-Switched-Capacitor Inverter," *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2021, 20-22 October, 2021, Hong Kong*, pp116-121.
- [63] S. Ryota, I. Jun, O. Yukihiko, and Y. Kyosuke, "Numerical Study of the Effect of Measurement Noise on the Accuracy of Bridge Parameter Estimation in VBI System Identification," *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2021, 7-9 July, 2021, London, U.K.*, pp10-15.

**Chukwunonso Henry Nwokoye** obtained a BSc and PhD degrees in computer science. He is a two-time ACM SIGCHI Gary Marsden Student Award recipient. His interests include simulation and modeling of complex systems, agent-based modeling, wireless sensor networks, and network security. More so, he has garnered skills and published findings in areas of human-computer interaction, such as social computing and computer-supported cooperative work. Additionally, he has conducted research on modeling and analysis of the propagation of malicious objects in network environments using analytical and agent-based modeling approaches. Besides the development of artificial intelligence (AI) models, at the Trustworthy AI Lab, he is conducting research in the area of accessible explainable AI for persons who are blind and partially sighted.