Applied Network Science

RESEARCH Open Access

A clustered-LPSEIRS malware propagation model in complex networks

Elham Asadi[1] and Soodeh Hosseini[1*]

*Correspondence:
so_hosseini@uk.ac.ir

[1] Department of Computer Science, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran

Abstract

In this paper, we present a new malware propagation model that integrates epidemic spread, clustering, and link prediction techniques, tailored for complex network networks. Our model is based on the clustered-link prediction-susceptible-exposed-infected-recovered (clustered-LPSEIRS) epidemic model, which simulates malware dissemination within the network. Our findings reveal a significant decrease in the rate of malware spread compared to the traditional SEIR model, with this enhancement in containment attributed to the integration of clustering and link prediction methods. We also compute the basic reproduction ratio (R_0) for our model, providing insights into the potential ramifications of malware within the network. By examining parameter variations, we enhance our understanding of the model's behavior under diverse scenarios. Additionally, we assess the influence of clustering and link prediction on mitigating malware spread, emphasizing its effectiveness in diminishing the overall impact.

Keywords: Malware propagation, Link prediction, Clustering, Basic reproductive ratio, Complex networks, Epidemic model

Introduction

The rapid advancement of the internet and artificial intelligence has ushered in a new era of technological transformation. A significant consequence of the widespread use of the internet is the emergence of cybercrime, where malicious actors exploit technology to commit various illegal activities. Malware, harmful software designed to compromise computer systems, is a primary tool used in cybercrime. It is encompassing several types of malicious software. Common types of malware include viruses, worms, Trojan horses, rootkits, and ransomware. These malicious programs exploit vulnerabilities in computer systems to cause harm, such as damaging files, stealing sensitive data, or gaining unauthorized access (Aslan and Samet 2020).

Complex networks can be classified into two main categories: homogeneous and heterogeneous networks. Within heterogeneous networks, scale-free networks with non-uniform degree distributions are particularly significant. These networks exhibit unique characteristics, such as high clustering coefficients, power-law degree distributions, and relatively short average path lengths(Hofstad 2024). The Internet is an example of a scale-free network. The vertices are the network nodes and the edges show the connection between the nodes (Boccaletti et al. 2006). Due to the dynamic nature of complex networks and the unique behavior of individual nodes, diffusion phenomena often occur within these networks. Malware propagation modeling is also applied to prevent the spread of malware in complex networks.

Mathematical modeling uses mathematical language and concepts to represent and analyze a system. It serves as a complementary tool to theory and experimentation in scientific research (Wang et al. 2023). In the context of malware propagation and network patching, mathematical modeling offers a practical approach to understanding the underlying mechanisms and optimizing patching strategies. Testing malware spread and patching devices on a large scale can be impractical. Mathematical modeling provides a virtual environment for experimentation. Also, by analyzing the mathematical equations that describe malware propagation, researchers can gain insights into how malware spreads and interacts with network components (Wang et al. 2023). The general process of mathematical modeling involves the following steps (Rey 2015).

Step 1: Clearly define the problem to be addressed.

Step 2: Identify the key factors and variables that impact the studied phenomenon.

Step 3: Express the relationships between these factors using mathematical equations.

Step 4: Create a computational representation of the model and simulate its behavior.

Step 5: Analyze the results obtained from the simulation.

The epidemic model (Rey 2015) is one type of mathematical model. In epidemic disease modeling, the population is divided into groups. Based on the node's behavior, entering and leaving the groups is considered. Population groups are susceptible(S), exposed(E), infected(I), recovered(R), quarantined(Q), and vaccinated(V). Epidemic models can help researchers understand the dynamics of malware outbreaks and evaluate the effectiveness of various control measures, such as vaccination, quarantine, and social distancing.

Clustering has been employed as an analytical technique to group unlabeled data and extract meaningful information. In addition, clustering is a widely recognized topology management method, particularly in wireless sensor networks, where it is used to group nodes for efficient management or task execution, such as resource management in a distributed manner. The nodes within each cluster are highly similar in terms of their characteristics, while the degree of similarity between clusters is typically the lowest. The primary goal of clustering is to assign labels to objects that indicate the membership of each node in a cluster. While clustering is frequently employed to enhance energy efficiency, it can also address other quality-related objectives. In this paper, we focused on leveraging clustering to mitigate malware propagation.

Since the process of propagation in the network is done through the communication of nodes, edges are considered the main elements of spreading malware in the network. Therefore, by predicting the links that will be formed in the future and preventing the spread of malware through those links, we can reduce the spread of malware in the network.

This paper presents a novel approach to modeling the spread of malware in complex networks, utilizing the SEIRS epidemic disease model, clustering, and link prediction method. The model consists of multiple clusters. Each cluster follows the SEIR epidemic model. The optimal number of clusters is determined by leveraging the characteristic values of the Laplace matrix. In addition to network clustering, we employ the PA link prediction algorithm to forecast future links on the network. The infection rate of the susceptible node is combined with the probability of forming a link for each node, and in this way, the infection in the cluster network is reduced. The SEIRS model is employed in this process due to its simplicity and comprehensiveness.

The remaining sections of the paper are organized as follows: in "Related work" section provides a thorough review of the existing literature and related works that discuss malware spreading models in complex networks. In "The proposed model" section, a detailed description of the proposed model is provided, and its main components and mechanisms are described. In "Dynamical analysis of the model" section is dedicated to the analysis of the model's dynamics. The equilibrium points and the basic reproduction number are calculated, which helps to understand the spread of malware in the network. In "Numerical simulations" section, practical results obtained through simulation are presented. This section examines the results and consequences of model implementation in different scenarios. Lastly, in "Conclusions" section, the results and findings are summarized, and the discussion ends with the potential avenues for future research and improvement in this area.

Related work

Since malware can cause significant damage in a network, researchers have proposed numerous models to assess and quantify the extent of malware damage in various networks. In the context of malware propagation, the internet network, characterized by its scale-free and heterogeneous nature, is particularly relevant. A heterogeneous network can be represented as a graph, where nodes represent objects and edges represent their connections. In modeling the spread of malware with epidemic models, the network is divided into groups based on their characteristics., like the degree of node, centrality, and membership in the community is divided. McKendrick (1927), presented the fundamental model. In his model, the population is divided into three groups: susceptible, infected, and recovered. In subsequent models, additional groups were incorporated into the population, enabling a more nuanced investigation of the diffusion phenomenon. Some of them include Exposed (E), Vaccinated (V), and Quarantined (Q) groups. The exposed group represents the nodes that have been exposed to the infection but the infection is still hidden in them, the vaccinated group includes the nodes that are immunized to the infection, and the quarantine group includes the infected nodes that are quarantined to prevent the spread of the pollution.

Shen et al. (2019a) proposed the HSEIR-V model, which is based on epidemic disease dynamics and incorporates four groups: susceptible, infected, recovered, and out-of-network systems. The model analyzes the stability, equilibrium points, and basic reproduction ratio of WSNs (Wireless Sensor Networks). The system can exit the network in two ways: through malware infection or without malware infection. The authors also introduced a parameter to account for the heterogeneity of the network, which is influenced by communication connections.

Yadav and Kumar (2022) introduced a model that incorporates quarantined and vaccination states into the malware propagation process, taking into account the role of quarantine and vaccination in preventing the spread of malware.

Shen et al. (2019b) introduced the Susceptible-insidious-infectious-recovered-Dysfunctional (SNIRD) model. Their model simulates the behavior of wireless sensor networks with heterogeneous sensor nodes. The model considers the communication connectivity between nodes and incorporates the characteristics of malware and dysfunctional sensor nodes. In the SNIRD model, malware only infects nodes in the infectious state (I), while nodes in the dysfunctional state (D) are physically damaged or intentionally destroyed by the resident malware. Shen et al. compared their model with traditional SIS and SIR models, demonstrating that their SNIRD model can effectively reduce the spread of malware.

Lahrouz et al. (2020) proposed the SIRI model, a mathematical framework for understanding the propagation of epidemics in heterogeneous networks. Their study focused on the impact of network topology and model parameters on the epidemic threshold, which is a critical point at which the spread of an epidemic becomes self-sustaining. The authors also explored the concept of temporary immunity, where individuals who have recovered from an infection may still be susceptible to re-infection, particularly in heterogeneous networks where individuals may transition from an improved state to an infected state.

Chen et al. (2022), considering the potential spread of malware through infrastructure-based (INF) communication links and device-to-device (D2D) connections in heterogeneous networks, introduced the SIRD model to investigate the dynamic nature of propagation through these connections. The results of analysis and simulation showed that the mobility and utilization of both INF and D2D connections significantly contribute to malware propagation in networks. They demonstrated that increasing security awareness among users and improving the recovery rate can considerably reduce the extent and intensity of malware spread.

In another paper, Shen et al. (2020) proposed a novel heterogeneous and mobile model, referred to as the VCQPS (vulnerable, compromised, quarantined, patched, scrapped) model, which takes into account the heterogeneity and mobility of wireless sensor network nodes. By comparing their model with the SIS and SIR models, they demonstrated a significant reduction in the spread of malware. The authors also explored the dynamics of malware propagation in heterogeneous wireless sensor networks, highlighting the importance of considering these factors in modeling the spread of malware. The results of this study can provide valuable insights for network managers, helping them make informed decisions to mitigate the impact of malware propagation in WSNs.

Zhu and Huang (2020) proposed a cluster-based model for Wireless Sensor Networks (WSNs), which classified nodes into two categories: cluster head nodes and ordinary nodes. They treated recovered nodes as part of the entire network. The primary focus of their model was to investigate the data transmission characteristics between different nodes in WSNs. The authors incorporated real-world parameters of WSNs, including the communication radius between nodes, node density, and node death rate, to develop a more accurate network representation. Through theoretical analysis and numerical simulations, they demonstrated a significant correlation between the communication radius, node density, node death rate, and the dynamics of malware propagation in WSNs.

Roberto et al. (2021) introduced a simplified model with two interconnected cluster networks. They analyzed this model to determine the critical parameters that must be regulated to prevent the emergence of new attacks. The authors demonstrated that when a virus or malware is known, the most effective strategy to prevent its spread is to introduce antidote nodes containing programs capable of disseminating throughout the network and protecting other nodes. They considered conversion rates from susceptible to antidote and infected to antidote for susceptible and infected nodes, respectively. Numerical results showed that having at least one device equipped with an antidote program is sufficient to prevent the spread of viruses and malware, ensuring a disease-free equilibrium point is always achieved.

Most existing research on modeling malware spread in networks has focused on single-cluster or two-cluster models, neglecting the impact of clustering on the spread of malware. In this paper, we investigate the effect of clustering on the spread of malware in networks and demonstrate how predicting future links can reduce the spread in clustered networks.

The proposed model

When the malware infiltrates the network, it infects vulnerable nodes. If the susceptible node is placed in front of the infected node, it will be infected. In this way, the spread of malware will occur in the network. The proposed model consists of the following steps.

Step 1: The link prediction process is performed on the network.

Step 2: Obtain the optimal number of clusters (c).

Step 3: The network is divided into c clusters.

Step 4: The LPSEIR epidemic disease model is obtained for each cluster according to the new network.

Figure 1 shows the architecture of the proposed model.

Clustering method

Clustering is the process of dividing data into groups, or clusters, such that data points within a cluster are more similar to each other than to data points in other clusters. In network analysis, clustering often aims to divide a network into k subgraphs, or communities. Spectral clustering is a nonlinear method for clustering data represented as a graph. This algorithm involves the following steps (Forouzandeh et al. 2023):

Step 1: Construct the similarity matrix: A similarity matrix is created to represent the pairwise relationships between data points. Each element in the matrix indicates the similarity or dissimilarity between two data points.

Step 2: Compute the Laplacian matrix: The Laplacian matrix is a transformation of the similarity matrix that captures the graph's structure. It is often calculated as the difference between the degree matrix and the similarity matrix.

$$L = D - A \tag{1}$$

where A represents the adjacency matrix and D is the degree matrix as Eq. 2.

$$d_i = \sum_{j|(i,j)\in E} w_{ij} \tag{2}$$

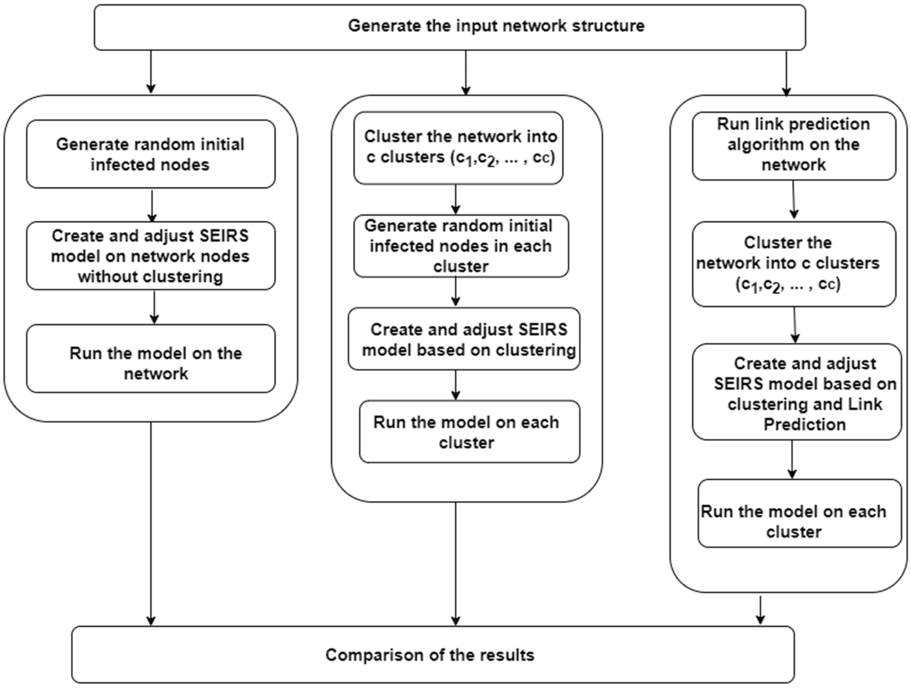

Generate the input network structure
Generate random initial infected nodes
Create and adjust SEIRS model on network nodes without clustering
Run the model on the network
Cluster the network into c clusters (c1,c2, ... , cc)
Generate random initial infected nodes in each cluster
Create and adjust SEIRS model based on clustering
Run the model on each cluster
Run link prediction algorithm on the network
Cluster the network into c clusters (c1,c2, ... , cc)
Create and adjust SEIRS model based on clustering and Link Prediction
Run the model on each cluster
Comparison of the results

Fig. 1 General architecture of the proposed model based on clustering and link prediction

Step 3: Find the eigenvectors: The k largest eigenvectors of the Laplacian matrix are computed. These eigenvectors represent the low-dimensional embedding of the data points.

Step 4: Cluster the eigenvectors: The embedded data points are clustered using a traditional clustering algorithm like k-means. The number of clusters, k, is specified beforehand.

A large spectral gap in the Laplacian matrix suggests a clear separation between clusters in the network. the eigenvectors associated with the smaller eigenvalues represent distinct and well-separated clusters (Li et al. 2021). This feature was utilized to identify the optimal number of clusters for the network.

Link prediction method

Link prediction is a process of predicting the presence or absence of links between nodes in a network. Based on the network structure, it is possible to predict whether two currently unconnected nodes will be connected in the future or not connected. In Fig. 2, graph G= (V, E) is given. C is the neighbor of D, and D is the neighbor of B. Consequently, there exists a possibility of a future link between B and C. A is also a 3-hop neighbor of B, which predicts the potential for a future link between B and A.

Link prediction approaches can be categorized into several groups, as shown in Fig. 3 (Kumar et al. 2020). We will now discuss some similarity-based approaches in more detail.

The simplest link prediction approach is similarity-based methods, which have lower computational complexity (Daud et al. 2020; Yuan et al. 2015). In the proposed model, we employ this approach to predict potential links.

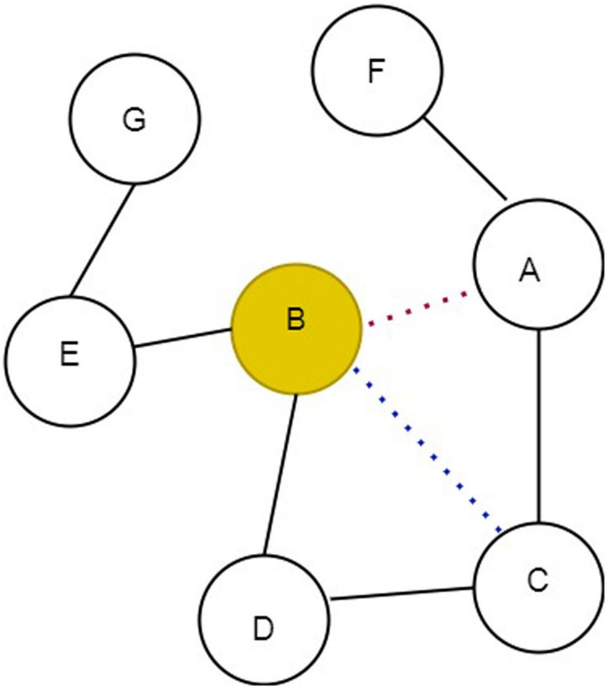


Fig. 2 The representation of link prediction

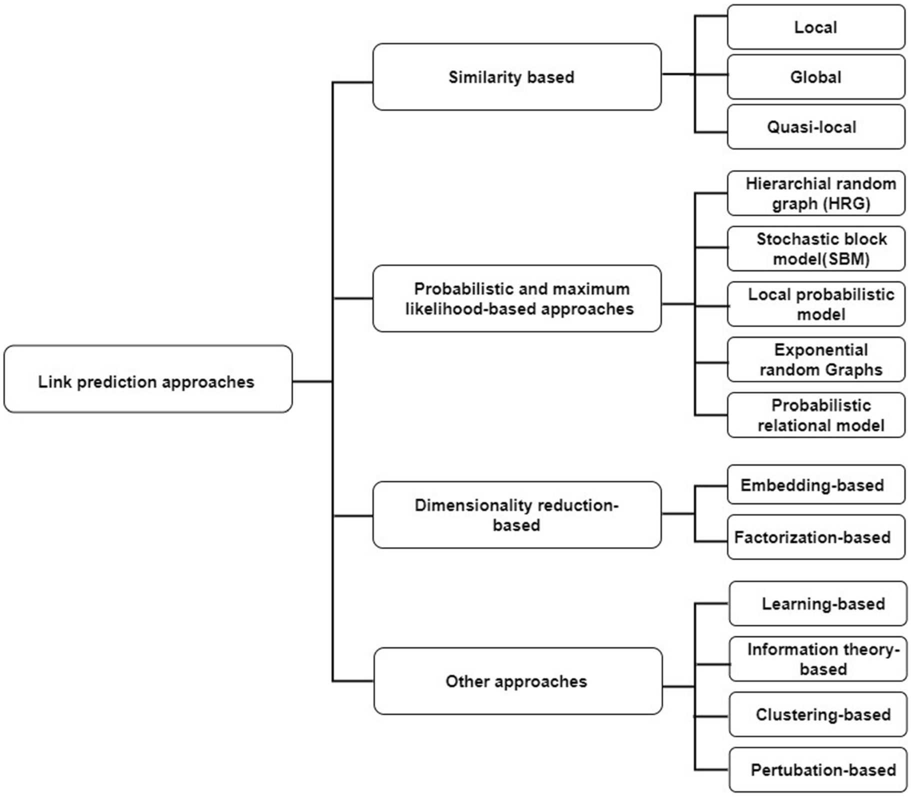

Link prediction approaches
Similarity based
Local
Global
Quasi-local
Probabilistic and maximum likelihood-based approaches
Hierarchial random graph (HRG)
Stochastic block model(SBM)
Local probabilistic model
Exponential random Graphs
Probabilistic relational model
Dimensionality reduction-based
Embedding-based
Factorization-based
Other approaches
Learning-based
Information theory-based
Clustering-based
Pertubation-based

Fig. 3 Classification of link prediction approaches (Kumar et al. 2020)

Similarity-based approaches work based on the similarity between two nodes that currently do not have a link between them in the graph structure. The similarity ranking between two nodes is determined based on their neighboring nodes or the degree of nodes. Then, the rankings are sorted in descending order, and the top-ranked entry is considered as a predicted link (Yuan et al. 2015). In the proposed model, we will use preferential attachment similarity for link prediction due to the lower complexity of the similarity measure. This criterion is taken from the rule of preferential attachment in the growth of scale-free networks. If we take the probability of connecting node x with other nodes, the preferential attachment nodes x and y will be in the form of Eq. 3 (Kumar et al. 2020).

$$S(xy) = k_x \times k_y \tag{3}$$

The preferential attachment mechanism based on node degrees only, without considering the neighbors, is a simplified model with less computational complexity. However, it may not perform well on many networks (Kumar et al. 2020; Lü and Zhou 2011).

The link prediction steps are (Rafiee et al. 2020):

Step 1: Calculate the degree of each node.

Step 2: Dividing the network edges into two sets, test, and train, so that 10% of the edges are in the test set and the remaining 90% are in the train set.

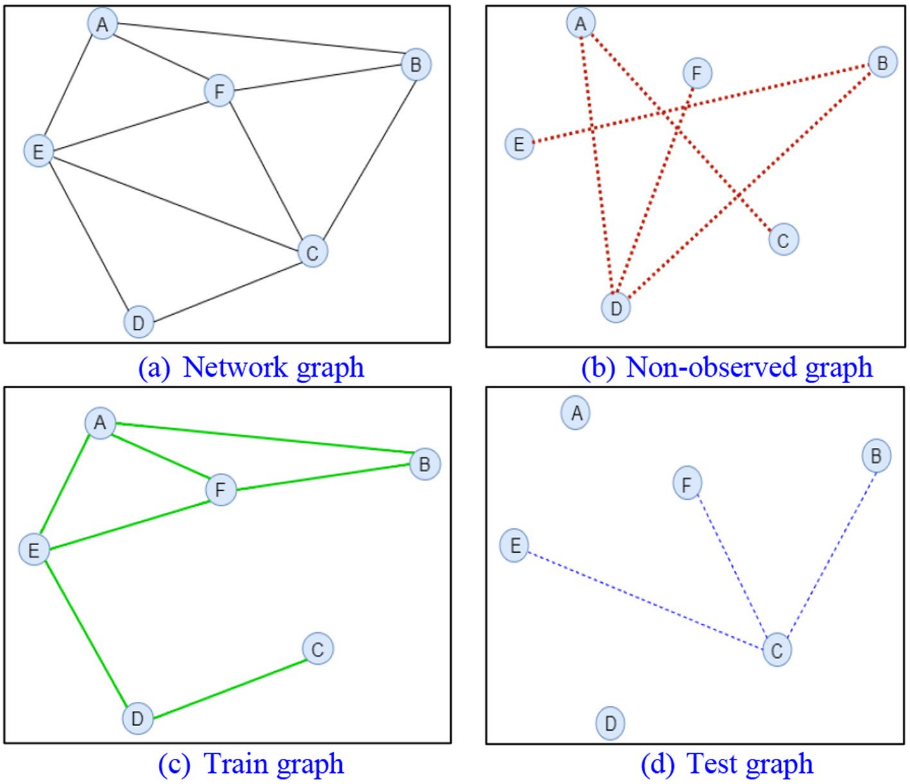
Step 3: We calculate the similarity measure based on the similarity measure PA for all edges that don't exist in the train graph. These edges are the union of the test set and the edges that do not exist in the network graph.

Step 4: Sorting the similarity measure obtained in the previous step in descending order.

Step 5: Insert the most similar edge to the train set.

Step 6: Calculate AUC and Precision.

Algorithm 1 presents a pseudocode implementation of the link prediction algorithm. Figure 4 presents an example of a network graph, along with visual representations of the graph at various stages of the algorithm's execution. Figure 4a depicts the main network graph. Figure 4b represents the graph of non-observed edges, which is essentially the complement of the network graph. Together, Fig. 4a and b form a complete graph containing all possible edges between the network's nodes. Step 2 introduces the test and train graphs, derived from the network graph. Figure 4c shows a portion of the network edges as the training graph, while Fig. 4d depicts the remaining edges as the test graph. In step 3, the similarity measure is calculated for both the non-observed edges (Fig. 4b) and the test edges (Fig. 4d). This involves combining the edges from these two graphs.



(a) Network graph

(b) Non-observed graph

(c) Train graph

(d) Test graph

Fig. 4 A display of the network graph and the selection of the test and train graph

Algorithm 1 Link Prediction Algorithm.

```
Input:  $G = (V,E)$ , $v = |V|$ , $e = |E|$.
Output:     aveAuc, avePrecision.
Begin
      For each node n in G
           Compute the degree of n.
      End;
      Divide G into training set $G_{train}$ and test set $G_{test}$.
       For non-observed edges $(x,y)$ in $G_{train}$ do
              Calculate the similarity score of the edge $(x,y)$ as $S_{xy} = d_x * d_y$.
       End;
       Sort the list of $S_{xy}$ in descending order.
       Add edges with the largest similarity $S_{xy}$ to $G_{train}$
       Calculate AUC and Precision.
       Calculate the average of AUC and the average of Precision.
End.
```

Equation 4 was used to calculate the AUC measure (Rafiee et al. 2020).

$$AUC = \frac{n_1 + 0.5n_2}{n} \tag{4}$$

where n is the total number of comparisons. n_1 is the number of times that the score of the link selected from the test set is more than the other links. n_2 is the number of times that both links have the same score. If the AUC is greater than 0.5, the algorithm's performance is acceptable.

Precision is another effective tool for validating link prediction, which is used in this paper. Precision is the percentage of correctly predicted links and is calculated as Eq. 5.

$$Precision = \frac{P}{T} \tag{5}$$

where P is the number of correctly predicted links and T is the total number of predicted links. A correctly predicted link is a link that belongs to both the prediction set and the test set.

Model description

The SEIR model is such that all the nodes of each cluster are divided into four categories: susceptible (S), exposed (E), infected (I), and recovered (R). A node with vulnerability is considered a susceptible node. when the susceptible node is placed in front of the infected node, it moves to the exposed state. After a certain incubation period, the malware becomes active and an exposed node becomes an infected node. When the infected node is detected and security mechanisms are applied, the node goes to the recovered state. This node may still be vulnerable, and it is placed in the susceptible state again. The infection of each cluster affects the other cluster. Due to the heterogeneity of the network, the nodes follow a power law distribution $P(k) \sim k^{-r}(2 < r \leq 3 \;\; k = 1 \ldots \Delta$ where $P(k)$ stands for the probability of selecting a node with degree kr is an indicator and Δ is the maximum node degree of the network. $P(k) = 0$ for all $k > \Delta$. Figure 5 shows the relationship between groups in the clustered network, and Table 1 shows the symbols and their descriptions. The states of the model are as follows:

Susceptible ($S_i^k(t)$): The number of susceptible nodes of cluster i with degree k at time t.

Exposed ($E_i^k(t)$): The number of exposed nodes of cluster i with degree k at time t.

Infected ($I_i^k(t)$): The number of infected nodes of cluster i with degree k at time t.

Recovered ($R_i^k(t)$): The number of recovered nodes of cluster i with degree k at time t.

The L parameter is determined by the link prediction algorithm, which we have used in our proposed model. Specifically, the value of L is based on the preferential attachment similarity criterion.

$$L = \frac{1}{Average(PA)} \tag{6}$$

The links created in the future can increase the degree of some nodes, making them more likely to be classified as important within the network. with security measures, we can effectively secure them and reduce malware propagation. securing process leads to a decrease in the density of susceptible nodes. We consider the degree of density reduction to be proportional to the reverse of the average preference attachment in the similarity measure of the link prediction method.

At the beginning of the modeling process, one of the challenges we encounter is defining the assumptions. In the presented model, the assumptions are as follows:

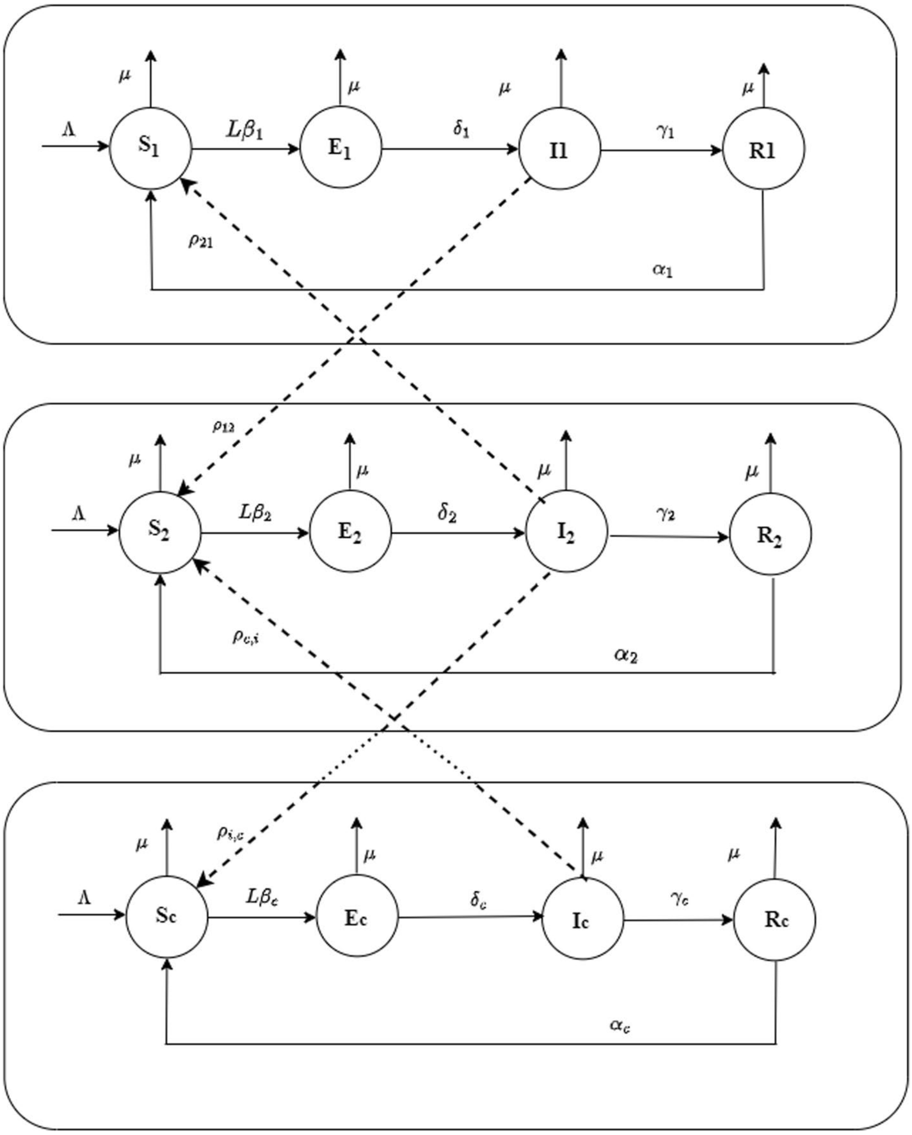


Fig. 5 State transition diagram of the clustered-LPSEIRS model

Table 1 Table of symbols and descriptions

Symbol	Descriptions
∧	The rate of adding a new node to the network (nodes are placed in susceptible mode(
μ	The rate of nodes that leave the network (death rate)
c	The number of clusters
C_i	The cluster set i that $i = 1..c$
β_i	Malware propagation rate in C_i where $0 \leq \beta_i \leq 1$and $i = 1..c$
δ_i	The transfer rate of exposed nodes to infected nodes in C_i where $0 \leq \delta_i \leq 1$and $i = 1..c$
γ_i	The transfer rate of infected nodes to recovered nodes in C_i where $0 \leq \gamma_i \leq 1$and $i = 1..c$
α_i	The transfer rate of recovered nodes to susceptible nodes in C_i where $0 \leq \alpha_i \leq 1$and $i = 1..c$
n_i	The number of nodes in C_i and $i = 1 \ldots c$
ρ_{ij}	The rate of infection transmission from the infected nodes of C_i to the susceptible nodes of the C_j

1. The network is heterogeneous, with N nodes which are divided into c clusters, and for each cluster, we have:

$$S_i^k(t) + E_i^k(t) + I_i^k(t) + R_i^k(t) = N_i \tag{7}$$

   where w $\sum_{i=1}^{c} N_i = N$ and N is the number of nodes in the network.
2. Initially, the modeling assumes that 10% of the nodes within a cluster are infected, while the remaining 90% are susceptible.
3. The number of births occurring within the cluster is balanced by the number of deaths ($\Lambda = \mu$).
4. The same leaving rate (μ) for each node is considered.

Model formulation

In this section, a mathematical model is developed to investigate the dynamics of network diffusion based on epidemic diseases, clustering, and link prediction. Transferring between states is possible for each cluster separately. In addition, it is possible to transfer malware from one cluster to another cluster. When a susceptible node encounters an infected node, either within its cluster or another, it becomes infected. This results in a decrease in the density of susceptible nodes within that cluster and a corresponding increase in the density of exposed nodes. The transition between the states can be shown in the following differential equations.

$$\begin{aligned}
\frac{dS_i^k(t)}{dt} &= \Lambda - \frac{L}{c}\beta_i S_i^k(t)\theta_i^k(t) - \mu S_i^k(t) + \alpha_i R_i^k(t) - \sum_{j|j\neq i, j\in C} \rho_{ji}\theta_j^k(t)S_i^k(t) \\
\frac{dE_i^k(t)}{dt} &= \frac{L}{c}\beta_i\theta_i^k(t)S_i^k(t) - \mu E_i^k(t) - \delta_i E_i^k(t) + \sum_{j|j\neq i, j\in C} \rho_{ji}\theta_j^k(t)S_i^k(t) \\
\frac{dI_i^k(t)}{dt} &= \delta_i E_i^k(t) - \mu I_i^k(t) - \gamma_i I_i^k(t) \\
\frac{dR_i^k(t)}{dt} &= \gamma_i I_i^k(t) - \mu R_i^k(t) - \alpha_i R_i^k(t)
\end{aligned} \tag{8}$$

$\theta_i^k(t)$ is the probability of the neighboring node being infected with degree k in C_i, which can be written as the following equation:

$$\theta_i^k(t) = \frac{1}{\langle k_i \rangle} \sum_{k=m_i}^{y_i} kP(k)I_i^k(t) \tag{9}$$

The average degree of C_i is equal to:

$$\langle k_i \rangle = \sum_{k} jp(j) \tag{10}$$

On the other hand, $p(k) = k^{-\upsilon}$ is equal to the probability of connecting a network node to k other nodes, ν is the power of power distribution, m_i is the minimum degree of C_i and y_i is the maximum degree of C_i.

One of the important issues in modeling is the estimation of model parameters. This issue will be time-consuming and expensive when there are many estimated parameters. To solve this problem, researchers have proposed different methods. One method is the Monte Carlo, which we have used in parameter estimation (Severt et al. 2023).

Dynamical analysis of the model

In this section, to investigate the dynamics of the proposed model, we calculate the equilibrium points of the system and the basic reproduction ratio.

Equilibrium points

In this section, the equilibrium points of the model are determined.

Malware-free equilibrium points are states where the disease is no longer present in the population, signifying the complete elimination of infection from the network. Malware-free equilibrium points are critical in comprehending the network dynamics and evaluating the efficacy of control measures in preventing and eradicating malware propagation. There are three equilibrium points:

- *Initial Malware-Free Equilibrium*: Initially, when all nodes are susceptible, the network reaches a malware-free equilibrium. In this state, no nodes are infected, and the network is entirely free of malware. Equation 11 represents the initial malware-free equilibrium for each cluster.

$$EQ_i = (N_i, 0, 0, 0) i = 1..c \tag{11}$$

- *Endemic Malware-Free Equilibrium*: At the end of the malware epidemic, when all nodes have recovered and the network has been completely cleared of infection, the malware-free equilibrium is reached. In this state, no nodes remain infected, and the network is restored to a malware-free state. For each cluster, This point is as follows.

$$EQ_i = (0, 0, 0, N_i) \quad i = 1 \dots c \tag{12}$$

  The initial or endemic equilibrium point for the network is when all clusters are at the initial or endemic equilibrium point.

- *Malware-free Equilibrium*: An additional equilibrium point arises when the network is infected, but the system remains in a state of equilibrium. This point is calculated by setting Eq. 8 equal to zero and solving for the variables. Equation 13 represents this equilibrium point. Table 2 provides the values of specific symbols used in simplifying Eq. 13.

Table 2 Table of symbol and descriptions

Symbol	Description
f_1	$\alpha_i\delta_i + \alpha_i\gamma_i + \delta_i\gamma_i + \frac{L}{c}(\alpha_i\beta_i\theta_i + \beta_i\delta_i\theta_i + \beta_i\gamma_i\theta_i) + \sum_{j=1,j\neq i}^{c}(\alpha_i\rho_{ji}\theta_j + \delta_i\rho_{ji}\theta_j + \gamma_i\rho_{ji}\theta_j)$
f_2	$\alpha_i\delta_i\gamma_i + \frac{L}{c}(\alpha_i\beta_i\delta_i\theta_i + \alpha_i\beta_i\gamma_i\theta_i + \beta_i\delta_i\gamma_i\theta_i) + \sum_{j=1,j\neq i}^{c}\alpha_i\delta_i\rho_{ji}\theta_j + \alpha_i\gamma_i\rho_{ji}\theta_j + \delta_i\gamma_i\rho_{ji}\theta_j)$

$$Q_i = \left(S_i^*, E_i^*, I_i^*, R_i^*\right) \tag{13}$$

$$\begin{aligned}
S_1^* &= \left(\frac{\Lambda\left(\mu^2(\alpha_i+\delta_i+\gamma_i)+\mu(\alpha_i\delta_i+\alpha_i\gamma_i+\delta_i\gamma_i)+\mu^3+\alpha_i\delta_i\gamma_i\right)}{\mu^3\left(\alpha_i+\delta_i+\gamma_i+\frac{L}{c}\beta_i\theta_i+\sum_{j=1,j\neq i}^{c}\rho_{ji}\theta_j\right)+\mu^4+\mu^2 f_1+\mu f_2}\right)\\
E_1^* &= \frac{\Lambda\left(L\beta_i\theta_i+\sum_{j=1,j\neq i}^{c}\rho_{ji}\theta_j\right)\left(\alpha_i\gamma_i+\alpha_i\mu+\mu\gamma_i+\mu^2\right)}{\mu^3\left(\alpha_i+\delta_i+\gamma_i+\frac{L}{c}\beta_i\theta_i+\sum_{j=1,j\neq i}^{c}\rho_{ji}\theta_j\right)+\mu^4+\mu^2 f_1+\mu f_2}\\
I_1^* &= \frac{\Lambda\delta_1(\alpha_i+\mu)\left(\frac{L}{c}\beta_i\theta_i+\sum_{j=1,j\neq i}^{c}\rho_{ji}\theta_j\right)}{\mu^3\left(\alpha_i+\delta_i+\gamma_i+\frac{L}{c}\beta_i\theta_i+\sum_{j=1,j\neq i}^{c}\rho_{ji}\theta_j\right)+\mu^4+\mu^2 f_1+\mu f_2}\\
R_i^* &= \frac{\Lambda\delta_i\gamma_i\left(\frac{L}{c}\beta_i\theta_i+\sum_{j=1,j\neq i}^{c}\rho_{21}\theta_2\right)}{\mu^3\left(\alpha_i+\delta_i+\gamma_i+\frac{L}{c}\beta_i\theta_i+\sum_{j=1,j\neq i}^{c}\rho_{ji}\theta_j\right)+\mu^4+\mu^2 f_1+\mu f_2}.
\end{aligned}$$

The basic reproductive ratio

The basic reproduction number (R_0) is a threshold limit to determine the presence or absence of a malware epidemic in the network. This ratio is the number of secondary infections that result from a primary infection. If $R_0 > 1$, there is a malware epidemic in the network. In this case, each infected node produces, on average, more than one new infection. It is caused the malware propagation across the network. If ($R_0 < 1$), the spread of the malware will stop. In this state, each infected node produces, on average, fewer than one new infection, which will eliminate the malware epidemic on the network (Driessche 2017).

The basic reproduction ratio, R_0, plays a significant role in assessing the severity and impact of the malware outbreak within a network. By understanding and estimating this value, it becomes possible to implement the appropriate control measures and preventive strategies to mitigate the spread of malware.

In calculating the basic reproduction ratio, the next-generation method is employed. This method is commonly used in mathematical epidemiology to estimate R_0 based on the underlying dynamics of the malware propagation within a network (Upadhyay and Iyengar 2013). In this way, R_0 is the spectral radius of the matrix G and $G = FV^{-1}$ (Guillén et al. 2017). The steps to obtain R_0 according to the next-generation method are as follows.

Step 1: Find the infected groups.

Step 2: Obtaine the matrix F according to the transmission of infection rate in infected groups.

Step 3: Obtaine the matrix V according to the transmission between infected groups.

Step 4: Calculate the matrix FV^{-1}.

Step 5: Calculate the spectral radius of the matrix FV^{-1} as R_0.

In the presented model, we obtain *the*R_0 value for each cluster. According to the steps of the next-generation method, the infected groups are $\mathrm{E}_\mathrm{i}^\mathrm{k}(\mathrm{t})$ and $\mathrm{I}_\mathrm{i}^\mathrm{k}(\mathrm{t})$ where $i = 1..c$. The function f_i is written based on the infection rate within groups $\mathrm{E_i^k(t)}$ and $\mathrm{I_i^k(t)}$, while the function v_i is written based on the transmission rate between these groups.

$$\begin{aligned} f_i(x) &= \begin{bmatrix} \frac{L}{c}\beta_i S_i^k(t)\theta_i^k(t) + \sum_{j=1,j\neq i}^{c} \rho_{ji}\theta_j^k(t)S_i^k(t) \\ 0 \end{bmatrix} \\ v_i(x) &= \begin{bmatrix} (\mu+\delta_i)E_i^k(t) \\ -\delta_i E_i^k(t) + (\mu+\gamma_i)I_i^k(t) \end{bmatrix} = \begin{bmatrix} u_1 E_i^k(t) \\ -\delta_i E_i^k(t) + u_2 I_i^k(t) \end{bmatrix} \end{aligned} \tag{14}$$

The Jacobian matrices of $F_i(x) = \frac{df_i(x)}{dx}$ and $V_i(x) = \frac{dv_i(x)}{dx}$ where $x = (E_i I_i)^T$ at the malware-free equilibrium $EQ_i = (N_i 000), \quad i = 1..c$ are

$$F_i = \begin{bmatrix} 0 & A_{ii} \\ 0 & 0 \end{bmatrix}, \quad V_i = \begin{bmatrix} u_1 & 0 \\ -\delta_i & u_2 \end{bmatrix} \tag{15}$$

Some equivalent values for the notations are listed in Table 3. We have obtained $R_{0\_i} = \rho\left(F_i V_i^{-1}\right)$ where $i = 1..c$.

$$R_{0\_i} = \frac{L\beta_i * \delta_i}{c(\mu+\delta_i) * (\mu+\gamma_i)} * \frac{\langle K_i^2 \rangle}{\langle k_i \rangle} * \frac{\Lambda}{\mu} \tag{16}$$

where $\langle K_i^2 \rangle = \sum_k k^2 p_i(k)$ and $p_i(k)$ is the degree distribution of cluster i.

In the case where the network is not clustered, the calculation of R_0 is as follows.

$$R_0 = \frac{\beta * \delta}{(\mu+\delta) * (\mu+\gamma)} * \frac{\langle K^2 \rangle}{\langle K \rangle} * \frac{\Lambda}{\mu} \tag{17}$$

The terms $\frac{\langle K^2 \rangle}{\langle K \rangle}$ and $\frac{\langle K_i^2 \rangle}{\langle k_i \rangle}$ show the effect of network or cluster topologies on the value of R_0 and R_{0_i}.

Therefore, the epidemic threshold of the clustered-LPSEIRS model is (Piqueira et al. 2021):

$$R_0 = R_{0\_1} \quad \text{or} \quad \ldots \quad \text{or} \quad R_0 = R_{0\_c} \tag{18}$$

Numerical simulations

In this section, the proposed model is analyzed. Our model is based on the SEIRS model, clustering, and link prediction method. It has been tested on a hypothetical Barabasi Albert (BA) and four real-world datasets: Soc-dolphins, ia-infect-dublin,

Table 3 Equivalent values for the used notations

Notations	Descriptions
A_{ii}	$\frac{L\beta_i}{c\langle k_i \rangle} S_i^k(0) \begin{bmatrix} 1 \\ 2 \\ \vdots \\ y_i \end{bmatrix} \left[\, p(1).2p(2).\ldots.y_i p(y_i) \,\right] = \frac{L\beta_i \langle K_i^2 \rangle}{c\langle k_i \rangle} \frac{\Lambda}{\mu}$
u_1	$\mu + \delta_i$
u_2	$\mu + \gamma_i$

Table 4 Network Characteristics

Name	Number of Vertices	Number of edges	Type
Hypothetical Barabasi Albert (BA)	1002	4002	Scale-free
Soc-dolphins	62	159	Social networks
Ia-infect-dublin	410	2765	Interaction networks
Bn-macaque-rhesus-brain-2	91	628	Brain networks
Eco-everglades	66	208	Ecology networks

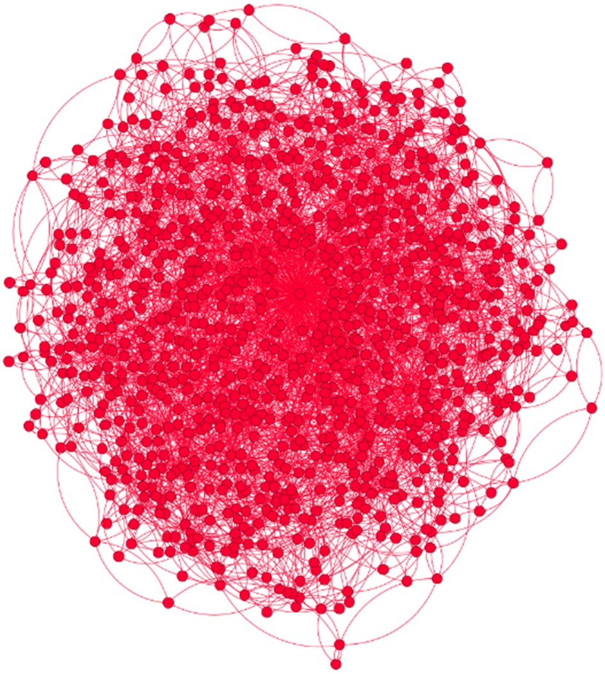


Fig. 6 The hypothetical Barabasi-Albert network

bn-macaque-rhesus-brain-2, and eco-everglades.They are available at https://networkrepository.com. Their characteristics are summarized in Table 4.

Numerical simulation in the BA

The hypothetical Barabasi Albert network is a type of scale-free network that has 1002 nodes and 4002 edges. Figure 6 is a view of this network. This picture is drawn in Gephi 0.10.2. The number of hypothetical network clusters is determined by analyzing the eigenvalues of the Laplacian matrix.

Figures 7 and 8, indicate changing the node densities in the SEIRS model and the proposed model. Initially, the density of susceptible nodes decreases while the densities of exposed nodes rise. Over time, exposed nodes transition to the infected state, further increasing the density of infected nodes. Upon implementing security mechanisms, infected nodes enter the recovery phase, leading to a growth in the density of recovered nodes.

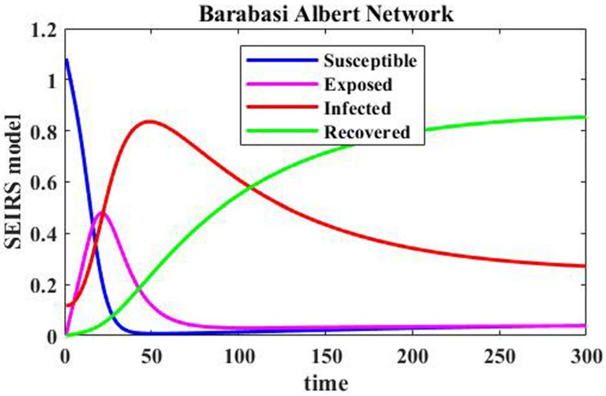


Fig. 7 The diagram of the SEIRS model on the hypothetical BA

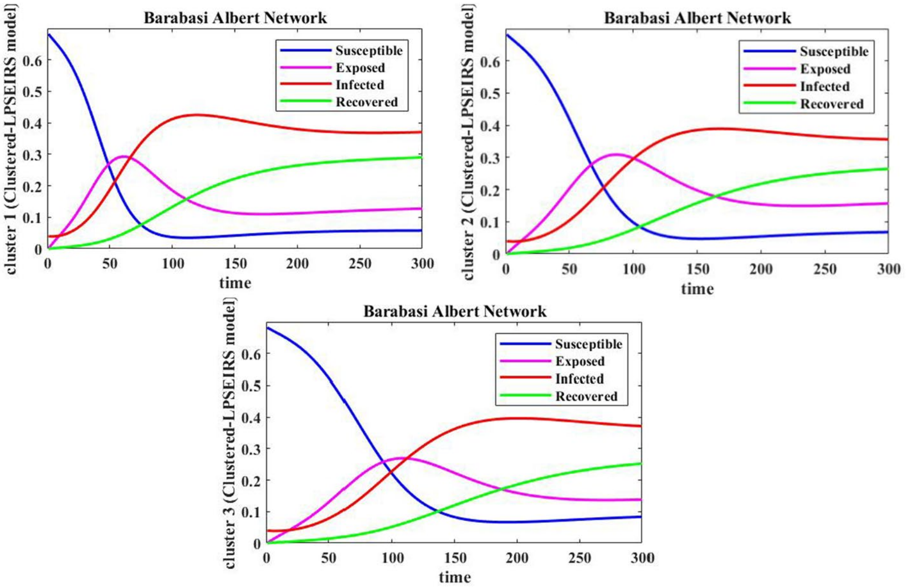


Fig. 8 The density of nodes in the cluster-LPSEIRS model on the hypothetical BA

Figure 9 illustrates the spread of malware within a hypothetical BA network under three clustering scenarios: a single cluster, two clusters, and three clusters. The results indicate that clustering significantly reduces malware propagation in the hypothetical BA network.

Figure 10 compares the density of infected nodes in three scenarios: without clustering, clustering with the optimal number of clusters, and clustering with both the optimal number of clusters and the link prediction parameter. The results demonstrate that the clustered-LPSEIRS model outperforms the other models in mitigating malware propagation within the hypothetical network. Figure 11 further highlights the superiority of the clustered-LPSEIRS model, exhibiting a lower density of recovered nodes due to its reduced pollution compared to the other approaches.

In the cluster-LPSEIRS model, a higher malware infection rate (β_i) leads to more rapid malware propagation within each cluster. Figure 12

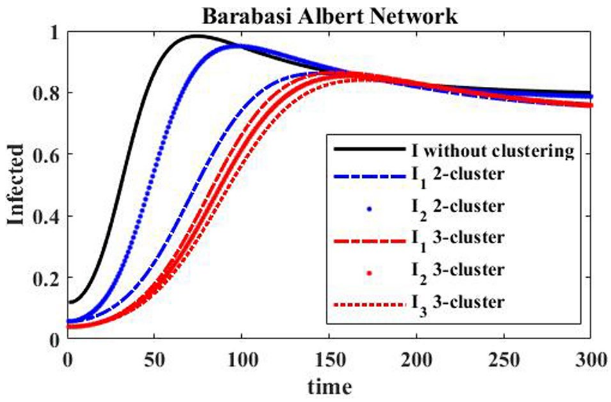


Fig. 9 Compare the infected groups in three models: SEIRS, two-cluster-SEIRS, and three-cluster-SEIRS on the hypothetical BA

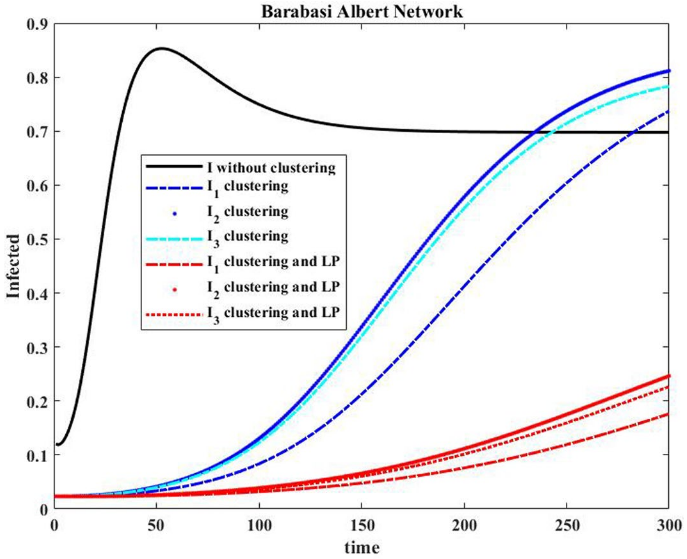


Fig. 10 Compare the infected groups in three models: SEIRS, clustered-SEIRS, and clustered-LPSEIRS on the hypothetical BA

visually demonstrates this trend, illustrating how the spread of malware in each cluster evolves in response to changes in the infection transmission rate and the values $\alpha_1 = 0 \cdot 009.\alpha_2 = 0 \cdot 005.\alpha_3 = 0 \cdot 005.\gamma_1 = 0 \cdot 004.\gamma_2 = 0 \cdot 003.\gamma_3 = 0 \cdot 003.\rho_{12} = \rho_{21} = \rho_{32} = \rho_{23} = \rho_{13} = \rho_{31} = 0 \cdot 005.\mu = 0 \cdot 001.\Lambda = 0 \cdot 001.\delta_1 = 0 \cdot 03.\delta_2 = 0 \cdot 02.\delta_3 = 0 \cdot 01$. Figure 13 demonstrates the density of infected nodes by changing the recovery rate in each cluster. Here the remaining parameter values are: $\alpha_1 = 0 \cdot 009.\alpha_2 = 0 \cdot 005.\alpha_3 = 0 \cdot 005.\beta_1 = 0 \cdot 09.\beta_2 = 0 \cdot 08.\beta_3 = 0 \cdot 07.$ $\rho_{12} = \rho_{21} = \rho_{32} = \rho_{23} = \rho_{13} = \rho_{31} = 0 \cdot 005.\mu = 0 \cdot 001.\Lambda = 0 \cdot 001.$ $\delta_1 = 0 \cdot 03.\delta_2 = 0 \cdot 02.\delta_3 = 0 \cdot 01.$ It is observed that a higher recovery rate correlates with a slower spread of malware.

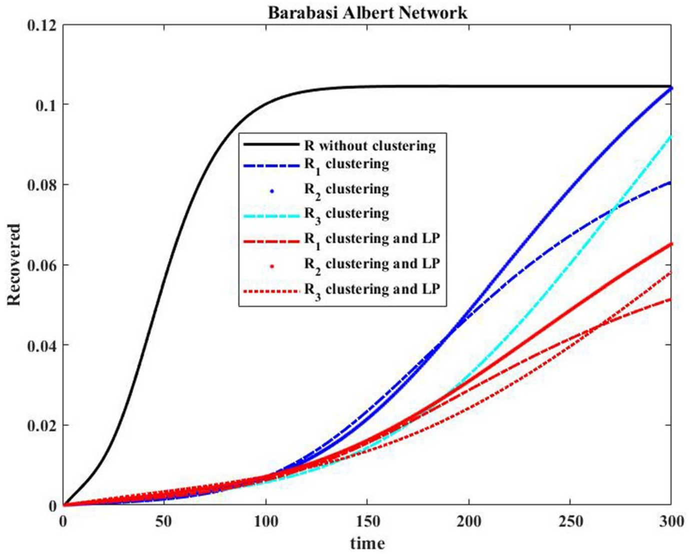


Fig. 11 Compare the recovered groups in three models: SEIRS, cluster-SEIRS, and cluster-LPSEIRS on the hypothetical BA

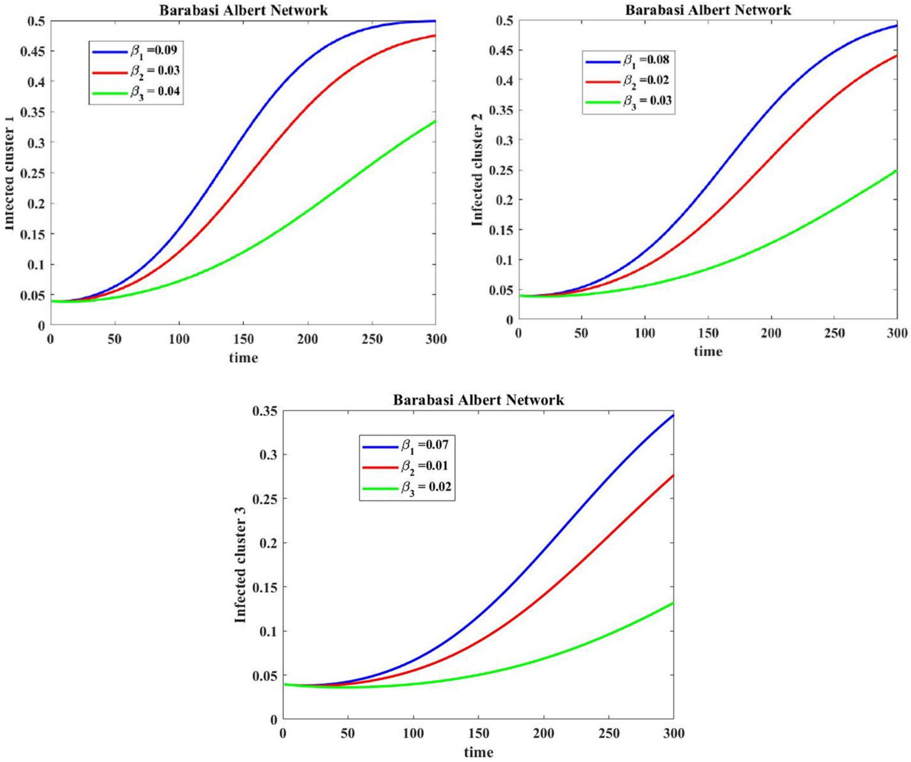


Fig. 12 The density of infected nodes in each cluster C_i that $i = 1..3$ on the hypothetical BA with changing of β

R_0 values were calculated for various β values in both the one-cluster and proposed BA network models. Clustering significantly reduces R_0. R_0 is a critical threshold for determining whether a malware epidemic will occur. If R_0 is less than 1, an epidemic is

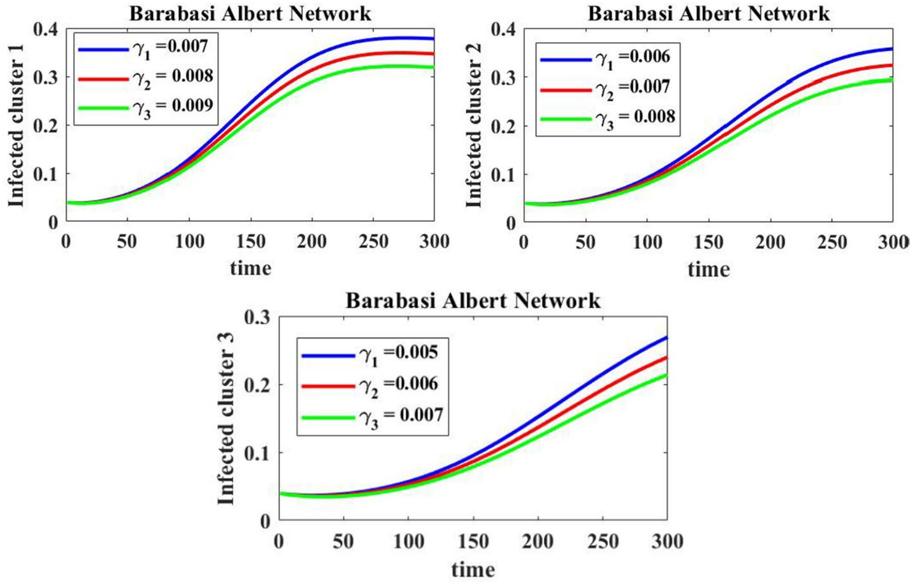


Fig. 13 The density of infected nodes in each cluster C_i that $i = 1..3$ on the hypothetical BA with changing of γ

Table 5 Table of values R_0 for SEIRS and cluster-LPSEIRS models on the hypothetical BA

	β	R_0	R_{0-1}	R_{0-2}	R_{0-3}
1	0.0050	0.6264	0.1289	0.0834	0.0693
2	0.0070	1.0022	0.2255	0.1459	0.1213
3	0.0090	1.3780	0.3222	0.2084	0.1733
4	0.0110	1.7539	0.4188	0.2709	0.2253
5	0.0130	2.1297	0.5155	0.3334	0.2773
6	0.0150	2.5055	0.6122	0.3959	0.3293
7	0.0170	2.8814	0.7088	0.4585	0.3813
8	0.0190	3.2572	0.8055	0.5210	0.4333
9	0.0210	3.6330	0.9021	0.5835	0.4853
10	0.0230	4.0088	0.9988	0.6460	0.5373

unlikely. However, if R_0 exceeds 1, an epidemic is more probable. As shown in Table 5, without clustering and link prediction, a malware epidemic occurs for β values greater than 0.005. However, with clustering and link prediction, the epidemic onset is delayed. Figure 14 illustrates how R_0 evolves as the infection rate (β) increases.

Numerical simulation on the datasets

This section presents simulation results using real-world networks: soc-dolphins, ia-infect-dublin, bn-macaque-rhesus-brain-2, and eco-everglades. Figure 15 visualizes these networks as simulated in Gephi 0.10.2 and includes graphs representing the relationship between the number of clusters and the eigenvalues of the Laplacian matrix. The optimal number of clusters is determined by the first significant gap in

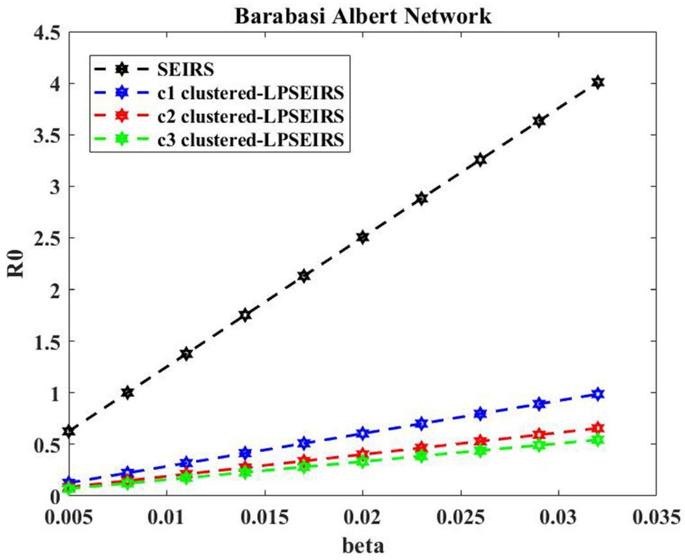


Fig. 14 Compare R_0 in cluster-LPSEIRS and SEIRS on the hypothetical BA

the cluster-eigenvalue graph. Table 6 lists the optimal cluster numbers for each network, which were used in the network simulations.

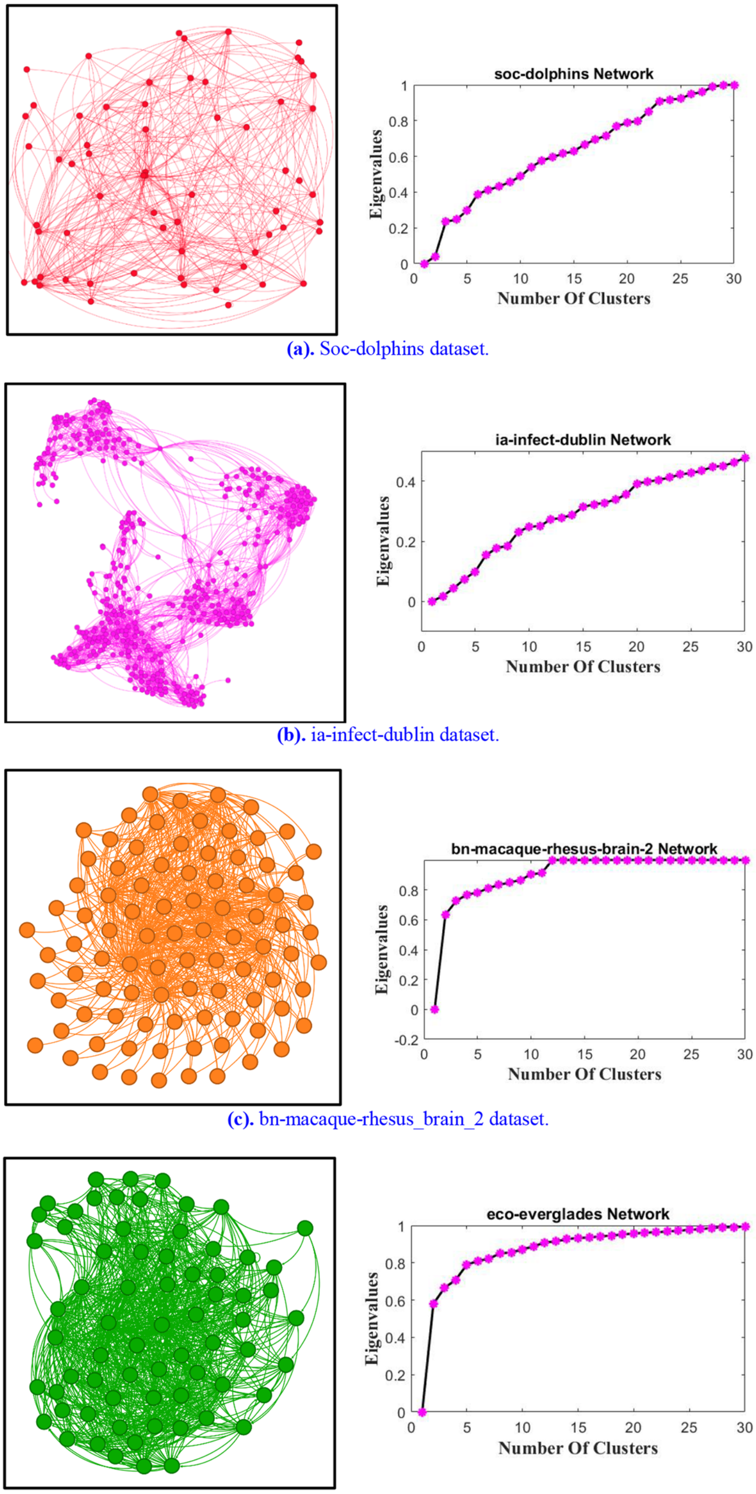
The simulations for the Soc-dolphins and ia-infect-dublin datasets were conducted using the optimal number of clusters determined in Table 6. Figure 16 compares the density of infected nodes in three models: SEIR, clustered-SEIRS, and clustered-LPSEIRS, using the Soc-dolphins dataset. Figure 17 presents a similar comparison for the ia-infect-dublin dataset. In both cases, the proposed model consistently outperforms the other models in controlling the spread of infection.

Tables 7 and 8 present the basic reproduction number (R_0) values for the Soc-dolphins and ia-infect-dublin datasets as the pollution transfer rate increases in the SEIRS and cluster-LPSEIRS models. As expected, the cluster-LPSEIRS model consistently exhibits lower R_0 values than the SEIR model. This indicates a reduced likelihood of a malware epidemic in the cluster-LPSEIRS model. Moreover, the proposed model experiences a delayed onset of malware epidemics compared to the SEIR model. Figures 18 and 19 visually represent these findings.

Since the optimal number of clusters for the bn-macaque-rhesus_brain_2 and eco-everglades datasets is one, which limits the ability to assess the impact of clustering, we intentionally used three clusters in our simulations. This allowed us to evaluate how clustering affects the proposed model's performance even in networks with a single optimal cluster.

Figure 20 compares the density of infected nodes in the SEIRS, cluster-SEIRS, and cluster-LPSEIRS models using the bn-macaque-rhesus_brain_2 and eco-everglades datasets. The results demonstrate that the cluster-LPSEIRS model consistently outperforms the other models in mitigating the spread of malware within the network.

Tables 9 and 10 present the R_0 values calculated for the SEIRS and clustered-SLPSEIRS models on the bn-macaque-rhesus_brain_2 and eco-everglades datasets as the malware



(a). Soc-dolphins dataset.

(b). ia-infect-dublin dataset.

(c). bn-macaque-rhesus_brain_2 dataset.

(d). eco-everglades dataset.

Fig. 15 Illustration of networks and depicting the relationship between eigenvalues and the number of clusters for each dataset

Table 6 Datasets and the optimal number of clusters

Dataset	Number of clusters
Soc-dolphins dataset	2
Ia-infect-dublin dataset	5
Bn-macaque-rhesus_brain_2 dataset	1
Eco-everglades dataset	1

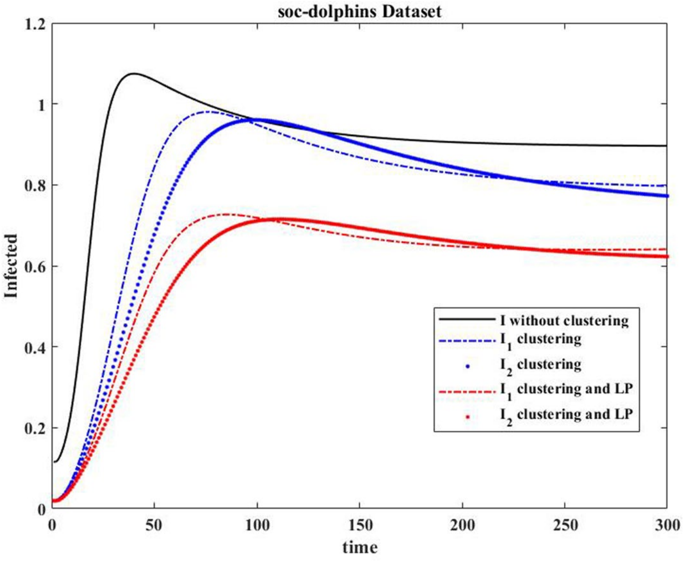


Fig. 16 Comparison of the infected node density in three models: SEIRS, cluster-SEIRS, and cluster-LPSEIRS on the Soc-dolphins dataset

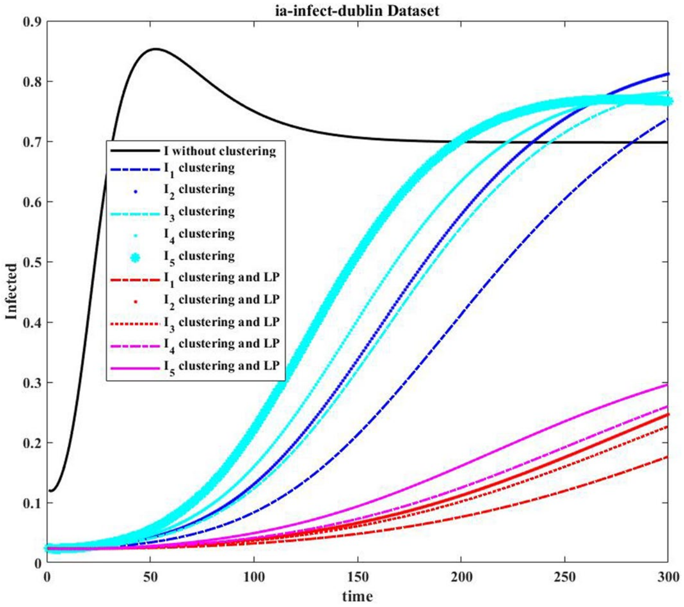


Fig. 17 Comparison of the infected node density in three models: SEIRS, cluster-SEIRS, and cluster-LPSEIRS on the ia-infect-dublin dataset

Table 7 Table of the values R_0 for SEIRS and cluster-LPSEIRS models on the Soc-dolphins dataset

	β	R_0	R_{0-1}	R_{0-2}
1	0.0050	0.3578	0.1217	0.0692
2	0.0070	1.7890	0.3921	0.3459
3	0.0090	3.2203	0.6624	0.6225
4	0.0110	4.6515	0.9328	0.8992
5	0.0130	6.0828	1.2032	1.1759
6	0.0150	7.5140	1.4736	1.4526
7	0.0170	8.9452	1.7440	1.7293
8	0.0190	10.3765	2.0143	2.0060
9	0.0210	11.8077	2.2847	2.2827
10	0.0230	13.2390	2.5551	2.5594

Table 8 Table of the values R_0 for SEIRS and cluster-LPSEIRS models on the ia-infect-dublin dataset

	β	R_0	R_{0-1}	R_{0-2}	R_{0-3}	R_{0-4}	R_{0-5}
1	0.0050	0.5380	0.3151	0.2459	0.4396	0.0824	0.1251
2	0.0070	11.2984	0.9453	0.9484	1.6957	1.7307	3.2526
3	0.0090	22.0587	1.5756	1.6509	2.9517	3.3790	6.3800
4	0.0110	32.8191	2.2058	2.3534	4.2078	5.0274	9.5075
5	0.0130	43.5795	2.8360	3.0560	5.4638	6.6757	12.6350
6	0.0150	54.3398	3.4662	3.7585	6.7199	8.3240	15.7625
7	0.0170	65.1002	4.0965	4.4610	7.9759	9.9723	18.8899
8	0.0190	75.8605	4.7267	5.1635	9.2320	11.6206	22.0174
9	0.0210	86.6209	5.3569	5.8660	10.4880	13.2689	25.1449
10	0.0230	97.3813	5.9871	6.5686	11.7441	14.9172	28.2723

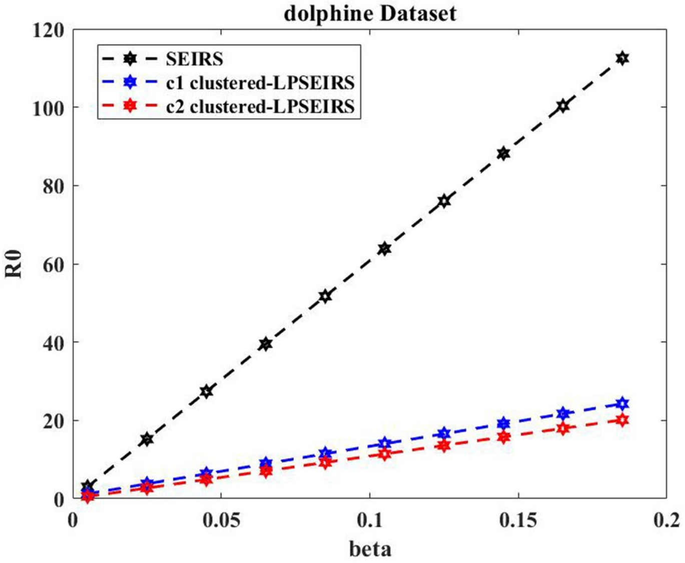


Fig. 18 Comparison of the values R_0 in the SEIRS and cluster-LPSEIRS on the Soc-dolphins dataset

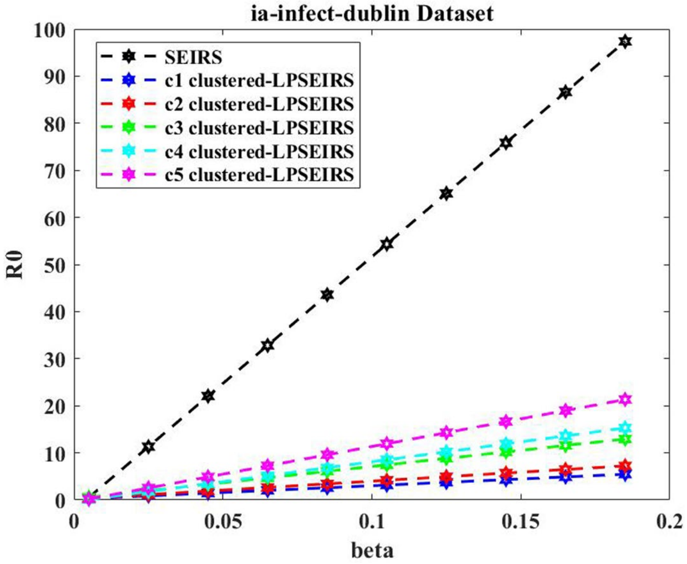


Fig. 19 Comparison of the values R_0 in SEIRS and cluster-LPSEIRS in the ia-infect-dublin dataset

transmission rate increases. These results are also visualized in Fig. 21. In both datasets, R_0 increases with rising malware transmission rates, but this increase is slower in the proposed model compared to the SEIR model. This indicates a delayed onset of malware epidemics in the proposed model.

Figure 22 analyzes the density of infected nodes in the cluster-SEIRS model across various datasets. The simulation was conducted with up to three clusters to assess the impact of clustering. The results indicate that increasing the number of clusters effectively reduces pollution within the network, emphasizing the efficacy of clustering in mitigating the spread of infection.

Conclusions

This paper investigates the dynamics of the LPSEIRS model in complex networks, incorporating clustering and link prediction. The network topology was generated using the Barabasi-Albert model with a power-law degree distribution. The optimal number of clusters was determined based on the Laplacian matrix eigenvalues.

The proposed model was analyzed mathematically to derive the basic reproduction ratio and equilibrium points. Simulations were conducted on both hypothetical Barabasi-Albert networks and real-world datasets. Three scenarios were considered: no clustering, clustering, and clustering with link prediction. The effects of varying malware transmission rates and recovery rates were examined.

By increasing the malware transmission rate, we calculated the basic reproduction number (R_0) and found that epidemics start more rapidly without clustering and link prediction. Simulation results demonstrate that c-cluster networks ($c > 1$) exhibit reduced malware spread compared to single-cluster networks. Moreover, R_0 is lower in c-cluster networks.

Clustering significantly contributes to mitigating malware spread in networks. Additionally, combining clustering with link prediction further reduces the spread of

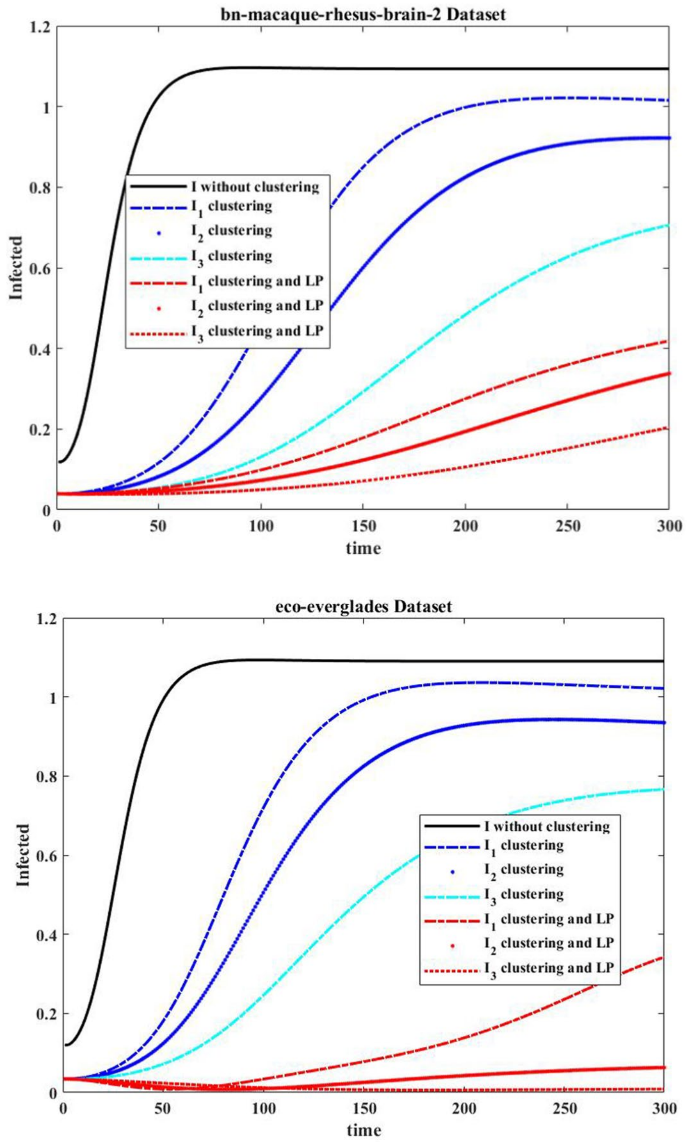


Fig. 20 Comparison of the infected node density in three models: SEIRS, cluster-SEIRS, and cluster-LPSEIRS on the bn-macaque-rhesus_brain_2 and eco-everglades datasets

Table 9 Table of values R_0 for SEIRS and cluster-LPSEIRS models on the bn-macaque-rhesus-brain-2 dataset

	β	R_0	R_{0-1}	R_{0-2}	R_{0-3}
1	0.0100	4.8418	0.4956	0.2958	0.0657
2	0.0130	6.2944	0.6442	0.3845	0.0854
3	0.0160	7.7469	0.7929	0.4733	0.1052
4	0.0190	9.1995	0.9416	0.5620	0.1249
5	0.0220	10.6520	1.0902	0.6507	0.1446
6	0.0250	12.1045	1.2389	0.7395	0.1643
7	0.0280	13.5571	1.3876	0.8282	0.1840
8	0.0310	15.0096	1.5362	0.9169	0.2038
9	0.0340	16.4622	1.6849	1.0057	0.2235
10	0.0370	17.9147	1.8336	1.0944	0.2432

Table 10 Table of values R_0 for SEIRS and cluster-LPSEIRS models on the eco-everglades dataset

	β	R_0	R_{0-1}	R_{0-2}	R_{0-3}
1	0.0100	0.4489	0.6618	0.5683	0.8975
2	0.0130	1.7956	0.7280	0.6251	0.9872
3	0.0160	3.1423	0.7941	0.6820	1.0770
4	0.0190	4.4890	0.8603	0.7388	1.1667
5	0.0220	5.8358	0.9265	0.7956	1.2565
6	0.0250	7.1825	0.9927	0.8524	1.3462
7	0.0280	8.5292	1.0588	0.9093	1.4360
8	0.0310	9.8759	1.1250	0.9661	1.5257
9	0.0340	11.2226	1.1912	1.0229	1.6155
10	0.0370	12.5693	1.2574	1.0798	1.7052

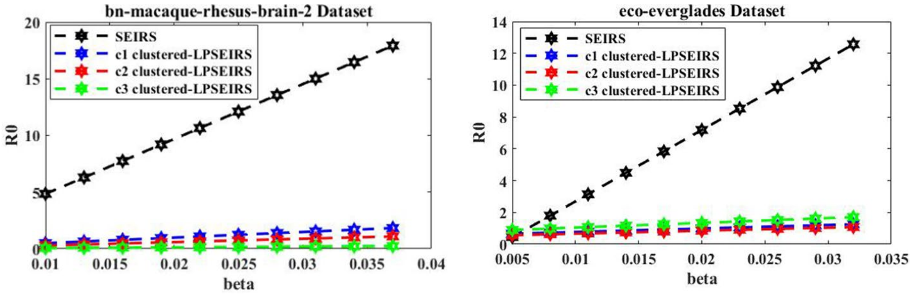


Fig. 21 Comparison of the values R_0 in SEIRS and cluster-LPSEIRS in the bn-macaque-rhesus-brain-2 and eco-everglades datasets

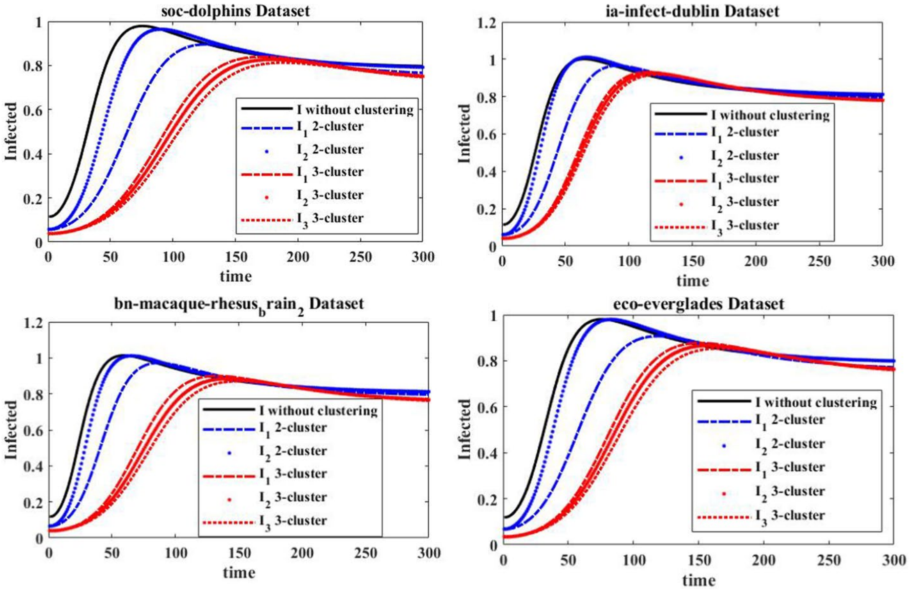


Fig. 22 Comparison of the infected nodes density in cluster-SEIRS model with c = 1..3 cluster on the datasets

malware. Comparative analysis of different clustering scenarios reveals that increasing the number of clusters leads to a decrease in malware propagation.

Author contributions
Conceptualization, E. Asadi and S.Hosseini; methodology, E. Asadi; software, E. Asadi; validation, S.Hosseini and E. Asadi; formal analysis, S.Hosseini and E. Asadi; investigation, S.Hosseini; resources, E. Asadi; data curation, S.Hosseini.; writing—original draft preparation, E. Asadi; writing—review and editing, S.Hosseini.; visualization, E. Asadi.; supervision, S.Hosseini.; project administration, S.Hosseini.

Funding
This research received no external funding.

Availability of data and materials
No datasets were generated or analysed during the current study.

Code availability
No codes are associated with this article.

Declarations

Ethics approval and consent to participate
This article does not contain any studies with human participants performed by any of the authors.

Competing interest
The authors declare no competing interests.

Received: 23 July 2024 Accepted: 25 September 2024
Published online: 01 October 2024

References

Aslan ÖA, Samet R (2020) A comprehensive review on malware detection approaches. IEEE Access 8:6249–6271
Boccaletti S et al (2006) Complex networks: Structure and dynamics. Phys Rep 424(4–5):175–308
Chen B-R, Cheng S-M, Mwangi MB (2022) A mobility-based epidemic model for IoT malware spread. IEEE Access 10:107929–107941
Daud NN et al (2020) Applications of link prediction in social networks: a review. J Netw Comput Appl 166:102716
del Rey AM (2015) Mathematical modeling of the propagation of malware: a review. Secur Commun Netw 8(15):2561–2579
Forouzandeh S et al (2023) A new method for recommendation based on embedding spectral clustering in heterogeneous networks (RESCHet). Expert Syst Appl 231:120699
Guillén JH, Del Rey AM, Encinas LH (2017) Study of the stability of a SEIRS model for computer worm propagation. Phys A 479:411–421
Kermack WO, McKendrick AG (1927) A contribution to the mathematical theory of epidemics. Proc R Soc Lond Ser A Contain Pap Math Phys Character 115(772):700–721
Kumar A et al (2020) Link prediction techniques, applications, and performance: a survey. Phys A 553:124289
Lahrouz A et al (2020) Global dynamics of an epidemic model with incomplete recovery in a complex network. J Franklin Inst 357(7):4414–4436
Li T et al (2021) An integrated cluster detection, optimization, and interpretation approach for financial data. IEEE Trans Cybern 52(12):13848–13861
Lü L, Zhou T (2011) Link prediction in complex networks: a survey. Physica A 390(6):1150–1170
Piqueira JRC, Cabrera MA, Batistela CM (2021) Malware propagation in clustered computer networks. Phys A 573:125958
Rafiee S, Salavati C, Abdollahpouri A (2020) CNDP: Link prediction based on common neighbors degree penalization. Phys A 539:122950
Severt M, Casado-Vara R, del Rey AM (2023) A Comparison of Monte Carlo-Based and PINN parameter estimation methods for malware identification in IoT networks. Technologies 11(5):133
Shen S et al (2019a) HSIRD: a model for characterizing dynamics of malware diffusion in heterogeneous WSNs. J Netw Comput Appl 146:102420
Shen S et al (2019b) SNIRD: Disclosing rules of malware spread in heterogeneous wireless sensor networks. IEEE Access 7:92881–92892
Shen S et al (2020) An epidemiology-based model for disclosing dynamics of malware propagation in heterogeneous and mobile WSNs. IEEE Access 8:43876–43887
Upadhyay RK, Iyengar SR (2013) Introduction to mathematical modeling and chaotic dynamics. CRC Press
Van den Driessche P (2017) Reproduction numbers of infectious disease models. Infect Dis Model 2(3):288–303
Van Der Hofstad R (2024) Random graphs and complex networks, vol 54. Cambridge University Press
Wang X et al (2023) Modeling, critical threshold, and lowest-cost patching strategy of malware propagation in heterogeneous IoT networks. IEEE Trans Inf Forensics Secur 18:3531–3545
Yadav P, Keshri AK (2022) The dynamics of SEIQR-V malware propagation model in IoT networks. In: 2022 international conference on IoT and blockchain technology (ICIBT). IEEE

Yuan H et al (2015) A distributed link prediction algorithm based on clustering in dynamic social networks. In: 2015 IEEE international conference on systems, man, and cybernetics. IEEE

Zhu X et al (2020) Modeling and analysis of malware propagation for cluster-based wireless sensor networks. In: 2020 IEEE 6th international conference on dependability in sensor, cloud and big data systems and application (DependSys). IEEE

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.