**BINDURA UNIVERSITY OF SCIENCE EDUCATION**

**FACULTY OF SCIENCE AND ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE**



# Prediction Of Malware Propagation in Complex Networks Using The Susceptible Infected Recovered Model


**By**

**TENDERO**

**REG NUMBER**

**SUPERVISOR:**


***A RESEARCH PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE BACHELOR OF SCIENCE HONOURS DEGREE IN INFORMATION TECHNOLOGY***

**2023**

# Abstract

The aim of this study was to develop a system and assess the use of the Susceptible Infected Recovered(SIR) model in predicting the propagation and spreading of malicious software in complex computer networks. The study had three(3) research objectives, the first one was to evaluate different models and techniques used for malware propagation prediction in a computer network, the second objective was to design and implement a simulated environment which predicts malware propagation prediction in a complex computer network using the Susceptible Infected Recovered (SIR) model. The last and third objectives was to evaluate the effectiveness of Susceptible Infected Recovered(SIR) model in predicting malware propagation prediction in a complex computer network. Therefore, to this end, the researcher managed to review vast literature to do with the study in question, from whence the author acquired insight on the different variables which can be used for creating the virus propagation simulation. The author went on to study literature on the mathematical compartments which are effective for this task and chose the SIR model thus satisfying the first objective. The SIR model was employed by the virus propagation simulation, using the prototyping software development model. The simulation process for predicting malware propagation using the SIR model takes inputs of various parameters such as the total number of nodes and edges in two networks, propagation probability, total number of iterations in the two networks, and edge ratio between networks. The simulation process has significant implications for network security as it allows for the prediction of malware propagation and provides insights that can inform the development of more effective strategies to prevent and mitigate malware attacks in large-scale networks. The simulation yielded substantial and satisfactory outcomes concerning the anticipation of malware dissemination in intricate networks. The results indicated that the overall quantity of nodes and edges within the networks exerted a notable influence on the duration of malware propagation. In essence, as the number of nodes and edges increased, the time taken for malware to propagate grew longer. Moreover, it was discovered that the probability of propagation had a significant impact on the speed at which malware spread within the network.

# Contents

# CHAPTER 1: PROBLEM IDENTIFICATION

## 1.0 Introduction

The goal of spreading malware is to infect as many computers or mobile phones as possible with viruses, worms, or Trojan horses, thereby maximizing the scope of malware intrusion. The number and harm of malware have increased dramatically, and the threat it poses to the network and users

is regarded as one of the most significant risks in the coming years(Liu & Liu,2019). With the development of the internet and the diversification of network attacks, the concept of malware has transcended the narrow traditional concept, especially with the emergence of advanced persistent threats, supply chain attacks, dead networks, and ransomware. Malware has become more highlighted because of its exclusiveness, gaining more control and causing more damage to the target (Park,2020). One in ten URLs is a malicious link, up from one in sixteen in the previous year, according to Symantec's 2019 Internet Security Report (SIS Report, 2019). Additionally, three years after the large-scale outbreak of Wannacry ransomware in 2017, the infection rate of ransomware declined for the first time. Nevertheless, the infection rate of enterprise ransomware jumped 12%, contrary to the overall downward trend, indicating that the threat of ransomware to enterprises continues to increase (Neville, 2019). Moreover, because of the popularity of the Internet of Things(IoT) and smart cities in recent years, hackers have been creating malware to attack routers and other types of network devices characterized by their inability to install any type of security software. Thus, for the development of the Internet of Things and smart cities, increasing the communication security in the network, deploying new security strategies, and optimizing verification protocols can significantly reduce the harm caused by malware to the network (Stergiou et al., 2021).

## 1.1 Background Of The Study

According to Kaspersky Security Network(2022), South African businesses came under fierce attack from backdoor computer malware, with detections in the second quarter of 2022 surging by 140%. These attacks have become especially prevalent in Africa, with Nigeria, Kenya, and South Africa all recording a big increase in cases between the first and second quarters of 2022. Nigeria recorded 2,624 cases, representing an increase of 83%, while Kenya had 10,300 cases, representing an increase of 53%. South Africa recorded 11,872 cases of backdoor detections in the second quarter of the year, increasing by 140% compared to the prior period. Additionally, affected users, those successfully targeted by cybercriminals through backdoor malware, rose by 10%( Kaspersky Security Network,2022).

In Zimbabwe, there has been reports of malware attacks on educational institutions and companies' websites; with the Herald, the government, NUST and the Harare Institute of

Technology reportedly affected (Chindaro, 2017). This reflects the reality of the threat on Zimbabwe's doorstep. Companies and banking systems have also been subject to hacking (illegal penetration and use of computer systems) thus being defrauded by individuals of large amounts of money. The case of a Chitungwiza man who hacked OK Zimbabwe's Money Wave System before stealing $70 000 reported widely, is a typical example of such cybercrime activities (Chindaro, 2017).

Malicious file (virus, worms, malware) that exploit zero-day vulnerabilities have brought severe threats to internet security in the real world(Bissessar et al., 2020). To date, none of the offered updates could effectively and reliably immunize the hosts thoroughly against being attacked by those files. It also affects the system for good time until the users immunize their computers if they are in the infected state. Furthermore, the failure of some recover measurements or malicious file redefinition may also lead to high risks that the hosts being immunized would be infected again (Vimercati,2016). Furthermore, the failure of some recover measurements or malicious file redefinition may also lead to high risks that the hosts being immunized would be infected again. In 2017, there was widespread of wannaCry ransomware which is a worm malicious file that targeted computers running the Microsoft Windows operating system by encrypting data and demanding ransom payments in the Bitcoin cryptocurrency has severe damages on many public sectors. Such damage has reached public and health sectors as in the UK wannaCry ransomware (Ahmad et al., 2019).

According to , Juniper Research Report(2019) the cost of data breaches to increase to $2.1 trillion globally by 2019.Lot of new malwares are emerging day by day and these are polymorphic in nature, changing their signatures so it's become very challenging to detect. According to a research by (Bhattacharya, 2016), a whopping 317 million new types of malwares were discovered. The propagation of malicious files in a system of interacting computers could be compared to contagious diseases in the human population. With the relationship between the internet malicious files and biological diseases, epidemiological models have been widely used in formulating models for the propagation of malicious files (Park, 2020).

## 1.2 Statement of the Problem

The threat of computer viruses persists because of the constant demand for computers and networks. Companies all over the world are losing billions of dollars due to malware (Bhattacharya, 2016). The potential damage to government, financial, business, and academic institutions is extreme such as data loss and denial of service. When a computer virus infects a facility, the virus seeks to invade other facilities in the network by exploiting the convenience of the network protocol and the high connectivity of the network. Current techniques do not provide enough detection accuracy, specificity on zero-day attacks (Park, 2020). Hence, there is an increasing need for accurate calculation of the probability of computer-virus-infected areas for developing corresponding strategies, for example, based on the possible virus-infected areas, to interrupt the relevant connections between the uninfected and infected computers in time.

## 1.3 Aim of Research

To develop a system and assess the use of the Susceptible Infected Recovered(SIR) model in predicting the propagation and spreading of malicious software in complex computer networks.

## 1.4 Research Objectives

1. To evaluate different models and techniques used for malware propagation prediction in a computer network.
2. To design and implement a simulated environment which predicts malware propagation prediction in a complex computer network using the Susceptible Infected Recovered (SIR) model.
3. To analyze the effectiveness of Susceptible Infected Recovered(SIR) model in predicting malware propagation prediction in a complex computer network.

## 1.5 Research Questions

1. How to evaluate different models and techniques used for malware propagation prediction in a computer network?
2. How to design and implement a simulated environment which predicts malware propagation prediction in a complex computer network using the Susceptible Infected Recovered(SIR) model?

3. How to analyze the effectiveness of Susceptible Infected Recovered (SIR) model in predicting malware propagation prediction in a complex computer network?

## 1.6 Research propositions/ hypothesis

- $H_0$: The system will accurately predict malware propagation prediction in complex networks.
- $H_1$: The system will fail to accurately predict malware propagation prediction in complex networks.

## 1.7 Justification of Research

This research provides a basis for business continuation, privacy and maintaining a reputable image for banks and other corporations. Cyber-attacks are especially dangerous to government agencies, which store sensitive civil and national-security data on their computer systems. Cybersecurity safeguards the nation's infrastructure, classified data, classified information, and the nation's identity.

## 1.8 Limitations/Challenges

Some of the restrictions that come across during this project design include the following:

- The limited time in which the research is to be conducted.

## 1.9 Definition of Terms

**Complex network -** a complex network is a graph (network) with non-trivial topological features—features that do not occur in simple networks such as lattices or random graphs but often occur in networks representing real systems

**Susceptible infected removed model –** is a model of infection that has three compartments which are susceptible-infected-removed.

**Malware –** software that is specifically designed to disrupt, damage, or gain unauthorized access to a computer system.

# CHAPTER 2: LITERATURE REVIEW

## 2.0 Introduction

A literature review, according to (Puebo, 2020), is a scientific study prepared from published sources that summarizes current understanding on a given issue. In this chapter, the researcher concentrates on answering the research questions and reveals previous and current systems that are similar to the research project at hand that have been done by other authors. This will be

extremely valuable to the author because it will serve as a guide to identifying solutions, strategies, and techniques utilized by prior writers to solve earlier research problems. It is a tool that informs the researcher if the study proposal is possible based on the findings of previous researchers in that field. This chapter, in accordance with the definition of a literature review, provides information on how others have implemented a malware propagation models.

## 2.1 Malware

Malware refers to any type of software that is designed to harm computer systems, steal data, or compromise privacy. Malware can take many different forms, including viruses, Trojans, ransomware, and spyware. These malicious programs can be spread through a variety of methods, such as phishing emails, malicious websites, or infected software downloads. Once installed on a computer system, malware can cause a range of problems, from slowing down system performance to stealing sensitive information. Malware is a serious threat to individuals, businesses, and organizations of all types, and it is important to take steps to protect against it.

According to a report by Malwarebytes, the number of malware detections increased by 20% in 2020, with remote work and pandemic-related concerns contributing to a rise in cyberattacks. Malware can cause significant financial losses for businesses, as well as damage to reputation and loss of customer trust. Therefore, it is important for individuals and organizations to take proactive steps to protect against malware, such as using antivirus software, keeping software up to date, and being vigilant against phishing attempts. Additionally, businesses should have a comprehensive cybersecurity plan in place, including employee training and incident response procedures, to minimize the impact of a malware attack. By taking these steps, individuals and organizations can reduce the risk of falling victim to malware and protect their systems and data from harm.

### 2.1.1 Types of Malware

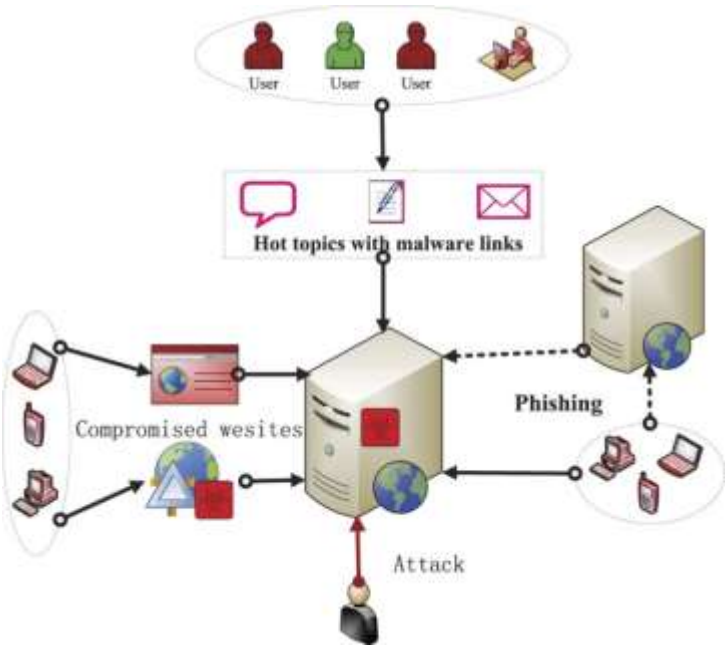
There are various types of malware that can be used to breach servers or computer networks. In this section, the author describes some of the most common types of malware used in cyber attacks, along with examples of their use in breaching servers and computer networks.

- **Virus**: A virus is a type of malware that can replicate itself by infecting other files or programs on a computer. Once the virus infects a system, it can spread to other systems through file sharing or email attachments. A famous example of a virus is the "ILOVEYOU" virus that infected millions of computers in 2000, causing an estimated $15 billion in damages.
- **Worm**: A worm is a type of malware that can replicate itself and spread to other systems without the need for a host file or program. Worms can exploit vulnerabilities in software to gain access to a system or network, and can be used to launch DDoS attacks or steal sensitive data. For instance, the Conficker worm infected millions of computers worldwide in 2009, causing disruptions to critical infrastructure such as hospitals and government agencies.
- **Trojan**: A Trojan, also known as a Trojan horse, is a type of malware that disguises itself as a legitimate program or file to trick users into installing or executing it. Once a Trojan is installed, it can give cyber criminals remote access to a system or network, allowing them to steal data, install other malware, or take control of the system. A notable example of a Trojan is the Emotet Trojan, which has been used in recent years to breach corporate networks and steal sensitive information.
- **Ransomware**: Ransomware is a type of malware that encrypts a user's files or locks them out of their system, demanding a ransom payment in exchange for restoring access. Ransomware attacks can cause significant disruptions to businesses and governments, with notable examples including the WannaCry and NotPetya attacks that affected critical infrastructure in 2017.
- **Adware**: Adware is a type of malware that displays unwanted advertisements on a user's computer. While not typically as harmful as other types of malware, adware can be a nuisance and can slow down a user's computer.
- **Spyware**: Spyware is a type of malware that is designed to monitor a user's activities on their computer or device. This can include recording keystrokes, capturing screenshots, and stealing login credentials. Spyware is often used to steal sensitive information such as banking details or personal information.

- **Rootkit**: A rootkit is a type of malware that is designed to hide its presence on a system, making it difficult to detect and remove. Rootkits can give attackers remote access to a system, allowing them to steal data or install other malware.
- **Fileless malware**: Fileless malware is a type of malware that resides in a computer's memory instead of on its hard drive. This makes it difficult for traditional antivirus software to detect and remove. Fileless malware is often used in targeted attacks against high-value targets such as government agencies or corporations.
- **Botnet**: A botnet is a collection of infected computers that are controlled by a central server, known as a Command and Control (C&C) server. Botnets can be used to launch DDoS attacks, steal data, or send spam emails. One notable example of a botnet is the Mirai botnet, which was used in a series of high-profile DDoS attacks in 2016.

## 2.2 Complex Networks

Complex computer networks refer to networks that are characterized by intricate structures, dynamic interactions between their components, and large-scale heterogeneity. Such networks can include the Internet, social networks, cloud computing networks, and distributed systems, among others. Complex computer networks are composed of interconnected nodes or devices that communicate with each other using various protocols and technologies. The topology of the network can be highly variable, ranging from centralized to decentralized, hierarchical to flat, and scale-free to random.




One example of a complex computer network is the Internet, which is a global network of interconnected computers and servers. The Internet is composed of different layers, including the physical layer, the data link layer, the network layer, the transport layer, and the application layer, each of which has its own protocols and standards. Another example of a complex computer network is a cloud computing network, which is a distributed system that provides on-demand access to computing resources over the Internet. Cloud computing networks are characterized by their scalability, flexibility, and cost-effectiveness, and are used by organizations and individuals for a wide range of applications, such as data storage, software development, and virtualization.

## 2.3 Malware Propagation In Complex Networks

Malware propagation in complex networks is a challenging problem that has received significant attention in recent years. Complex networks, such as social networks, computer networks, and biological networks, are characterized by intricate structures and dynamic interactions between their components. Malware propagation in complex networks can be influenced by a range of factors, including the topology of the network, the behavior of the individuals or nodes in the network, and the properties of the malware itself.

To study the problem of malware propagation in complex networks, researchers have developed a variety of models and algorithms that can capture the dynamics of the spread of malware in different contexts. These models and algorithms include compartmental models, graph theory-

based models, machine learning-based models, and game theory-based models. The choice of model or algorithm depends on the specific characteristics of the network and the malware being studied, as well as the research questions being addressed. Ultimately, the goal of studying malware propagation in complex networks is to develop effective countermeasures that can prevent or mitigate the spread of malware and protect individuals and organizations from cyber attacks.

## 2.4 Effects of Malware

Malware can allow attackers to steal sensitive data, disable critical systems, and gain access to other systems connected to the network.

One of the most significant effects of malware used to breach servers or computer networks is data theft. Malware can be used to steal sensitive data, such as personal information, financial records, and intellectual property. This data can then be used for identity theft, fraud, or sold on the black market. In some cases, the stolen data can be used to launch further attacks on other systems or networks. According to a report by Symantec, a leading cybersecurity company, the average cost of a data breach in 2020 was $3.86 million, with an average cost of $150 per stolen record.

Another effect of malware used to breach servers or computer networks is system disruption. Malware can be used to disable critical systems or cause them to malfunction, resulting in downtime and lost productivity. In some cases, malware can also be used to destroy data or render it unusable. This can be particularly devastating for organizations that rely on their computer systems to run their operations. According to a study by the Ponemon Institute, the average cost of a data center outage in 2020 was $740,357.

Finally, malware used to breach servers or computer networks can lead to a loss of trust in the affected organization. Customers, partners, and other stakeholders may lose confidence in the organization's ability to protect their data and systems. This can lead to reputational damage, lost business, and legal consequences. According to a survey by the Ponemon Institute, 74% of consumers would switch to a competitor after a data breach, and 59% of respondents said that they would be unlikely to do business with an organization that had experienced a data breach in the past.

## 1.5 Data Breaches

A data breach is an incident in which sensitive, protected, or confidential data is accessed, stolen, or disclosed by an unauthorized party. Data breaches can occur in various ways, including hacking, malware attacks, insider threats, and human error. The consequences of a data breach can be severe, including financial losses, reputational damage, and legal consequences. According to a study by IBM, the average cost of a data breach in 2021 was $4.24 million, with an average cost of $175 per stolen record. The study also found that the healthcare industry had the highest average cost of a data breach, at $9.23 million, while the financial industry had the highest cost per stolen record, at $301. Data breaches can have a significant impact on individuals as well, as their personal and financial information may be compromised, leading to identity theft, fraud, and other types of cybercrime.

South Africa and Zimbabwe have both experienced significant data breaches in recent years, many of which were caused by malware. In 2017, South Africa experienced one of the largest data breaches in its history when the personal information of over 60 million people was leaked online. The breach was caused by a vulnerability in the website of a major credit bureau, and it is believed that malware played a role in the attack. In 2019, Zimbabwe's largest mobile network operator experienced a data breach that exposed the personal information of over 7 million customers. The breach was caused by a malware infection that allowed attackers to steal customer data from the company's servers.

These data breaches highlight the growing threat posed by malware to organizations and individuals alike. As more and more data is collected and stored online, the risk of data breaches caused by malware is increasing. Organizations must take steps to protect themselves from these threats by implementing cybersecurity best practices, such as regularly updating software and using strong passwords. In addition, organizations should consider using tools like antivirus software and firewalls to protect against malware infections. By taking these steps, organizations can reduce the risk of data breaches caused by malware and protect themselves and their customers from the damaging effects of these attacks.

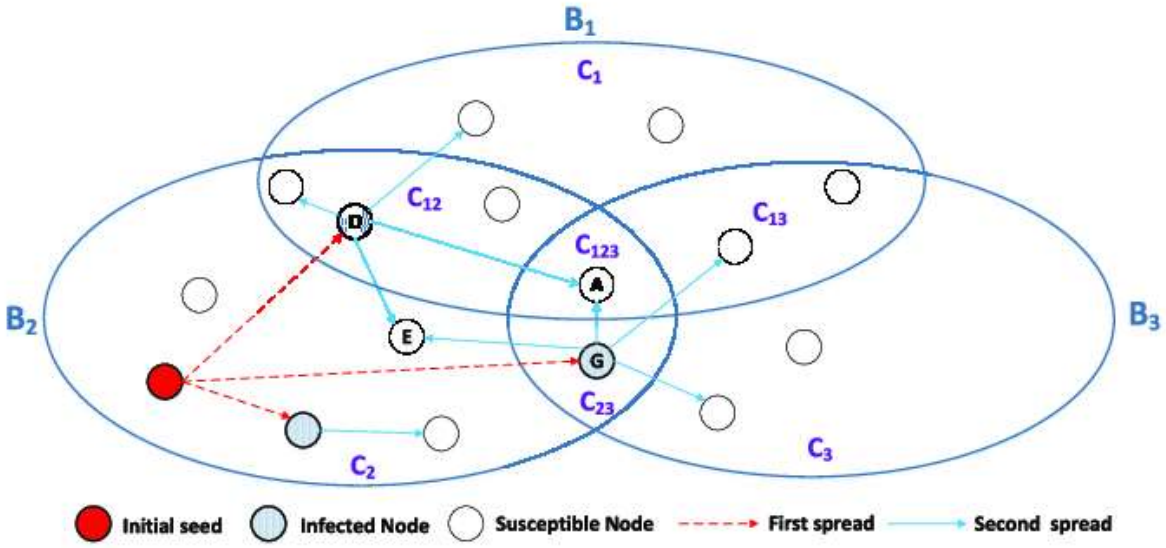
## 2.6 Compartmental Models

Compartmental models are a type of mathematical model used to describe and predict the spread of infectious diseases and other phenomena that involve the movement of individuals or objects between different states or compartments. In a compartmental model, the population is divided into compartments based on their current state, such as susceptible, infected, or recovered. The flow of individuals between compartments is described using a set of differential equations. There are several types of compartmental models, including the Susceptible-Infected-Recovered (SIR) model, Susceptible-Exposed-Infected-Recovered (SEIR) model, and Susceptible-Exposed-Infectious-Recovered (SEIR) model.

Compartmental models have also been used to study the spread of malware and other types of cyber threats. In this context, the compartments represent different states of the computer network, such as vulnerable, infected, and cleaned. The flow of malware between the compartments is described using a set of differential equations that take into account factors such as the rate of infection, the rate of removal, and the rate of patching. Compartmental models can be used to predict the spread of malware and to evaluate the effectiveness of different countermeasures, such as patching and isolation. According to a study by Liu et al., compartmental models can be used to predict the spread of malware with a high degree of accuracy, and can help organizations make more informed decisions about their cybersecurity strategies.

### 2.7.1 Susceptible Infected Recovered Model(S-I-R)

The Susceptible-Infected-Recovered (SIR) model is a compartmental model that is commonly used to study the spread of infectious diseases. In this model, the population is divided into three compartments: susceptible (S), infected (I), and recovered (R). Individuals in the susceptible compartment can become infected when they come into contact with infected individuals, and infected individuals can recover and move into the recovered compartment over time. The flow of individuals between compartments is described using a set of differential equations that take into account factors such as the rate of infection, the rate of recovery, and the size of the population.




The SIR model has also been used to study the spread of malware and other types of cyber threats. In this context, the compartments represent different states of the computer network, such as vulnerable, infected, and cleaned. The flow of malware between the compartments is described using a set of differential equations that take into account factors such as the rate of infection, the rate of removal, and the rate of patching. According to a study by Liu et al., the SIR model is a suitable approach for modeling malware propagation, as it can capture the dynamics of the spread of malware in a network and can provide insights into the effectiveness of different countermeasures.

The SIR model has been used in several studies to predict the spread of malware and evaluate the effectiveness of different countermeasures. For example, in a study by Al-Shaer and Hamed, the SIR model was used to study the spread of malware in a simulated network and to evaluate the effectiveness of different patching strategies. The study found that targeted patching based on the criticality of the vulnerable nodes was more effective in controlling the spread of malware than random patching or patching based on the degree of the nodes. Another study by Jia et al. used the SIR model to predict the spread of ransomware in a network and to evaluate the effectiveness of different backup strategies. The study found that regular backups and data separation can significantly reduce the impact of ransomware attacks.

## 2.8 Related Literature

While the SIR model has been used in previous studies to predict malware propagation in simple networks, its effectiveness in predicting malware propagation in complex networks is still unclear. Jia et al. (2018) explored the effectiveness of the SIR model in predicting the spread of ransomware in complex networks with community structure. They found that the SIR model can effectively predict the overall spread of ransomware, but may underestimate the impact of community structure on the propagation dynamics. They also proposed a backup strategy to mitigate the impact of ransomware propagation in complex networks.

One study that explored the application of the SIR model to predict malware propagation in complex networks is by Li et al. (2016). The study proposed a novel SIR-based model that incorporates network topology and individual behavior to predict the spread of malware in a social network. The results showed that the proposed model outperforms other state-of-the-art models in terms of accuracy and computational efficiency.

Another study that investigated the impact of network topology on the accuracy of the SIR model for predicting malware propagation is by Liu et al. (2018). The study used a graph theory-based approach to generate synthetic networks with different topologies and evaluated the performance of the SIR model under different scenarios. The results showed that the SIR model is more accurate in predicting malware propagation in scale-free networks than in random networks, due to the presence of highly connected nodes that act as super-spreaders.

Finally, Jia et al. (2018) proposed a game theory-based approach to predict malware propagation in a cloud computing network using the SIR model. The study introduced a novel concept of "backup strategies" to reduce the impact of malware propagation and evaluated the effectiveness of different control strategies, such as quarantine and vaccination. The results showed that the proposed approach can significantly reduce the impact of malware propagation and improve the resilience of the network against cyber attacks.

The Susceptible Infected Recovered (SIR) model has been widely used in the field of epidemiology to model the spread of infectious diseases. In recent years, the SIR model has also been applied to study the propagation of computer viruses and malware in computer networks. For example, Liu et al. (2018) developed an SIR-based model to simulate the spread of malware in a computer network, taking into account the effects of network topology and the behavior of the

infected nodes. Their results showed that the SIR model can effectively predict the temporal and spatial patterns of malware propagation in different network topologies.

To address the research gap on the effectiveness of the SIR model in predicting malware propagation in complex networks, Al-Shaer and Hamed (2010) proposed an SIR-based model that incorporates network topology and control strategies. They applied their model to study the propagation of network worms in a real-world network topology and compared the performance of different control strategies, such as quarantine and vaccination. Their results showed that the SIR model can accurately predict the spread of network worms and the effectiveness of different control strategies in mitigating the propagation.

In their work, Li et al. (2020) presented a compartmental model that combines epidemiological principles with network theory to predict the spread of malware. Their model considers the effect of different factors such as the network structure, the characteristics of the malware, and the behavior of the hosts. The proposed model was evaluated using real-world data, and the results showed that it can accurately predict the spread of malware in complex networks.

Another approach to predict the propagation of malware in complex networks using compartmental models was proposed by Yang et al. (2019). They presented a model that integrates the susceptible-infected-susceptible (SIS) model with network embedding techniques to predict the spread of malware. The model uses network embedding to capture the structural features of the network, and the SIS model to simulate the propagation of malware. The proposed model was evaluated using two real-world datasets, and the results showed that it outperforms existing methods in terms of accuracy and efficiency.

Compartmental models have also been used to analyze the impact of different control strategies on the spread of malware in complex networks. In their work, Li et al. (2019) presented a compartmental model that incorporates control measures such as vaccination and quarantine to mitigate the spread of malware. The proposed model was evaluated using real-world data, and the results showed that it can effectively predict the spread of malware under different control strategies. This approach provides valuable insights into the design of effective control strategies to contain malware propagation in complex networks.

In summary, compartmental models have shown promising results in predicting the propagation of malware in complex networks. These models consider various factors such as the network structure, the characteristics of the malware, and the behavior of the hosts to accurately predict the spread of malware. Additionally, compartmental models have been used to analyze the impact of different control strategies on the spread of malware, providing insights into the design of effective control measures.

## 2.9 Research Gap

While the Susceptible Infected Recovered (SIR) model has been widely used for modeling the spread of infectious diseases and computer viruses in simple networks, there is still a research gap in applying the SIR model to predict malware propagation in complex networks. Specifically, few studies have explored the impact of network topology, heterogeneity, and community structure on the accuracy of the SIR model in predicting malware propagation. Additionally, there is a need for more research on the effectiveness of different control strategies for mitigating malware propagation in complex networks, such as quarantine, vaccination, and network partitioning. Addressing these research gaps could help improve the accuracy and practicality of using the SIR model for predicting malware propagation in real-world scenarios.

## 2.10 Chapter Summary

The author was successful in obtaining and collecting relevant information and data for the research topic. Some of the concepts employed by the researcher came from a variety of places, including academic papers, textbooks, and the internet, which revealed holes that needed to be filled. The information gathered from all of these sources will be utilised in the preceding chapters of the study to meet the research project's objectives. The method utilized in the design and development of the proposed solution is discussed in the following chapter.

# CHAPTER 3: METHODOLOGY

## 3.0 Introduction

Research is a process of investigating and discovering facts, which can involve scientific studies or a detailed examination of a specific issue. Depending on the type of research being conducted, either quantitative or qualitative methods can be used, such as exploratory, descriptive, or diagnostic approaches. Research has been found to be a crucial tool for government institutions and policymakers in making informed economic decisions. The systematic and theoretical analysis of the methods or procedures employed in a particular field of study is referred to as methodology. This chapter aims to define the approaches used to achieve the research and system objectives, and to establish the necessary procedures for building a solution while selecting the best strategies to attain the desired results based on the information gathered in the previous chapter. To facilitate the study procedure, secondary data was utilized for analysis, sourced from official channels, the internet, and journals.

## 3.1 Research Design

According to (Moule and Goodman (2013)), the research design serves as the foundational framework for a study. Polit and Hungler (2014) define research design as the methodology employed to address research questions and tackle issues that may arise during the study's progress. There are four research models available for researchers to choose from, namely observational, experimental, simulation, or derived. Due to the need to develop and continuously evaluate the application's effectiveness in achieving the intended outcomes, the researcher opted to utilize experimental approaches.

## 3.2 Requirements Analysis

According to Abram Moore, Bourque, and Dupuis (2004), performing a requirements analysis is vital in determining the success or failure of a project. The identified requirements should be practical, documented, tested, executable, traceable, and measurable, and they should correspond to the identified business needs. Additionally, the requirements should be precise enough to facilitate system design. Thus, it is essential to document all functional and non-functional specifications of the required system at this stage. To ensure consistent and clear requirements, a thorough review, revision, and examination of the acquired requirements is necessary.

### 3.2.1 Functional Requirements

In the context of a system or component, functions refer to their operations, which are composed of three elements: inputs, behavior, and outputs. According to Bittner, functional requirements are the actions that a system must be capable of performing, without any consideration for physical constraints. These actions may include computations, specialized details, information processing, and other specific functionalities that define the intended system behavior. Use cases describe the behavioral scenarios that apply to the majority of instances in which the system applies the functional requirements.

The proposed system must be able to meet the following requirements:

i. to implement a simulated environment which predicts malware propagation prediction in a complex computer network using the Susceptible Infected Recovered (SIR) model.
ii. to predict malware propagation in a complex computer network using the Susceptible Infected Recovered (SIR) model.

### 3.2.2 Non-Functional Requirements

They are often referred to as quality requirements and used to judge the performance of a system rather than its intended behavior. The proposed system must be able to meet the following:

i. Performance requirements
ii. Flexibility requirements
iii. Quick response time

### 3.2.3 Hardware Requirements

- Core i5 processor or better

### 3.2.4 Software Requirements

- Windows 10 Operating system
- Visual Studio Professional 2019
- Microsoft Visual C#
- .Net Framework 4.5.2
- Wireshark
- Tomcat server

## 3.3 System Development

This chapter describes the overview of the system and how it was developed so as to produce the results. It specifies all the software tools and models used in the development of the system.In choosing a methodology for the development phase of the proposed solution, it was necessary to consider the strengths and weaknesses of different frameworks, which vary depending on the specific project and the desired outcomes. The methodology chosen for the project was the prototyping model, due to the need for frequent testing and refinement to arrive at a functional system that meets the specified objectives.

The prototyping model is a software development approach in which a quick prototype is constructed to grasp the requirements, which are not necessarily frozen before moving forward with design or coding. This prototype is based on the requirements that are currently known and serves as a foundation for the creation of the final system or software. The client can gain a "real feel" for the system by interacting with the prototype, as interactions with the prototype can help the customer better comprehend the requirements of the intended system.

The prototyping model is an effective approach for complex and massive systems where there is no manual procedure or existing system to assist in defining needs. The prototype is usually not a fully functional system, and many of the intricacies are not included. The goal is to create a system that is functional in general. The prototype is developed through a series of iterations, with each iteration building on the previous one, based on feedback from the client or other project stakeholders.

Other software development frameworks that were considered for the project include the waterfall and spiral models. The waterfall model is a linear approach to software development, where each phase of the development process must be completed before moving on to the next phase. This approach is suitable for simple and straightforward projects, where the requirements are well-defined and unchanging. However, it is less suitable for complex projects where requirements are likely to change over time.

The spiral model is an iterative approach that combines elements of the waterfall model with prototyping. It involves repeated cycles of development, testing, and evaluation, with each cycle building on the previous one, and incorporating feedback from the client or other stakeholders.

This model is suitable for projects where requirements are not well-defined, and where there is a need for frequent testing and evaluation.

Overall, the prototyping model was chosen as the most appropriate methodology for the project, given the need for frequent testing and refinement to arrive at a functional system that meets the specified objectives.

### 3.3.1 System Development Tools

In choosing a methodology for the development phase of the proposed solution, it was necessary to consider the strengths and weaknesses of different frameworks, which vary depending on the specific project and the desired outcomes. Several frameworks, including the waterfall, spiral, and prototyping models, were considered. Ultimately, the author opted for the prototyping model due to the need for frequent testing and refinement in order to arrive at a functional system that meets the specified objectives.

### 3.3.2 Prototyping

The prototyping model is a widely used software development approach that involves creating a preliminary version of the final software or system, which is then tested and modified until it meets the required specifications. This preliminary version serves as a starting point for the final system or software, and it helps to identify any issues or limitations early on in the development process. Rather than defining all the requirements upfront, the prototyping model allows for quick iterations to be made based on the current known needs. By interacting with the prototype, clients can better understand the requirements of the intended system, and it can provide a more accurate representation of how the final product will look and function. This method is particularly useful for developing complex systems with no existing procedures or systems to define requirements. The prototype is generally not a complete, fully functional system, but rather a general working model that can be further developed and refined based on feedback and testing.

The prototyping model is a software development approach that follows a series of phases. The first phase involves identifying the product requirements in detail, which is achieved through interviewing system users to understand their expectations of the system. In the design phase, a basic system design is created to give the user a quick overview of the system, although it is not yet a complete design. The prototype development is aided by this rapid design. Once the design is complete, an initial prototype of the target software is built based on the known needs, although not all product components may be perfect or accurate at this point. The first prototype is then tailored based on feedback from users, and a second prototype is built. After all iterations of the update are complete, the product is presented to the client or other project stakeholders for review, and their responses are collected in an organized manner to improve the system in the future. Finally, the product is scheduled for further enhancement based on considerations such as time, manpower, and budget, as well as the technical feasibility of actual implementation. Full approaches such as extreme programming or fast application development may be included in the context.

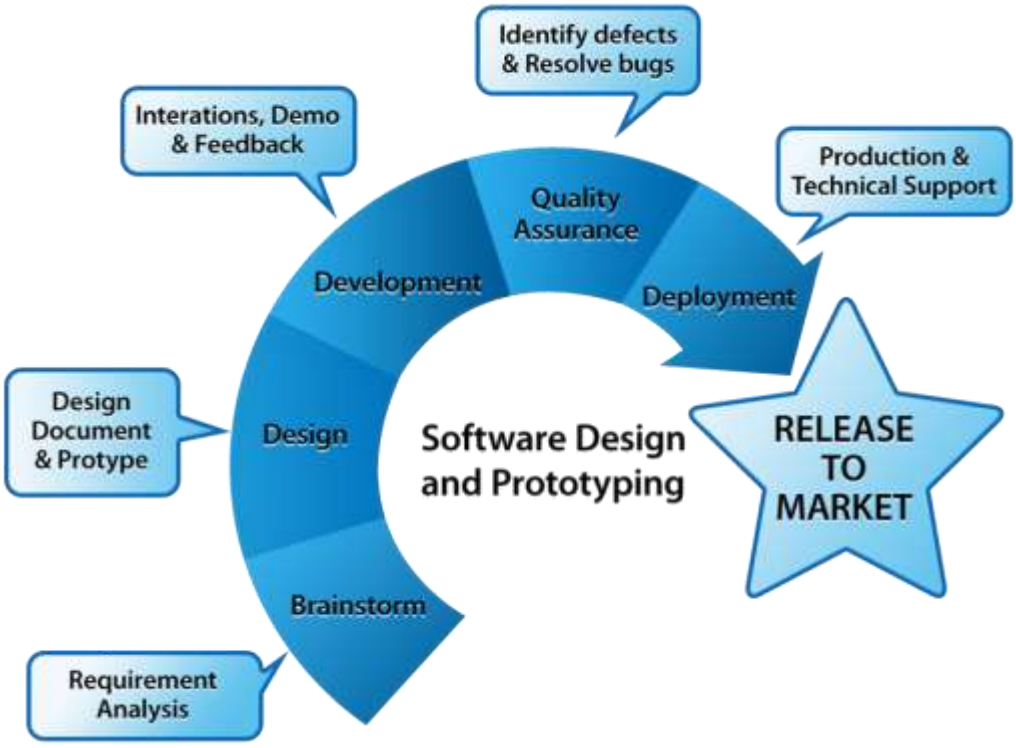



***Figure 1: Prototype Method***

Apart from the methodology the system was also developed using the following tools:

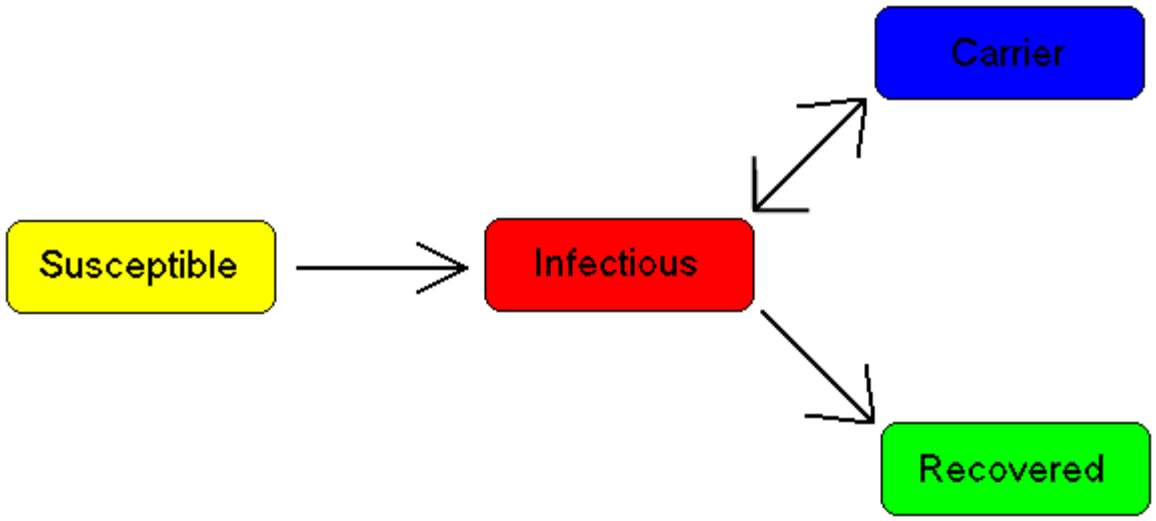
## 3.4 Summary of how the system works

The simulation process for predicting malware propagation using the SIR model takes inputs of various parameters such as the total number of nodes and edges in two networks, propagation probability, total number of iterations in the two networks, and edge ratio between networks. Once the parameters are inputted, the simulation runs automatically using mathematical functions to generate results which are written into a CSV file. The CSV file contains data that can be analyzed visually as a graph.

The graph shows two networks, one infected with malware, and the other that is not. The graph also shows the time it takes for the malware to propagate through the network as well as whether the malware will completely infect all nodes or not. The visual representation of the data makes it easy to analyze and interpret the findings of the simulation. The simulation process has significant implications for network security as it allows for the prediction of malware propagation and provides insights that can inform the development of more effective strategies to prevent and mitigate malware attacks in large-scale networks.

## 3.5 System Design

### 3.5.1 SIR Flow



## 3.6 Simulation Parameters

**Total Number of Nodes in Two Networks (Communities):** This parameter refers to the total number of nodes or individuals in the two networks or communities that are being studied. In the context of malware propagation, this would refer to the number of devices or computers that are connected to each other and can potentially spread malware. The size of the network is an important parameter as it influences the speed and extent of malware propagation.

**Total Number of Edges in Two Networks**: An edge is a connection between two nodes in a network. The total number of edges in two networks refers to the total number of connections between nodes in each of the two communities. In the context of malware propagation, an edge between two nodes would represent the possibility of malware being transmitted from one device to another. The number of edges in the network determines the rate and path of malware propagation.

**Propagation Probability**: This parameter refers to the likelihood of a node becoming infected with malware after coming into contact with an infected node. This probability determines the rate at which malware spreads from one node to another. The propagation probability can be influenced by several factors such as the type of malware, the security measures in place, and the behavior of the users.

**Total Number of Iterations in Two Networks:** An iteration refers to a single step in the simulation of the spread of malware. The total number of iterations in the two networks refers to the number of times the simulation will be run. A higher number of iterations can provide more

accurate predictions of the spread of malware and help identify potential vulnerabilities in the network.

**Edge Ratio Between Networks**: The edge ratio between networks refers to the proportion of edges between the two communities. In the context of malware propagation, this would represent the likelihood of malware being transmitted between the two communities. A higher edge ratio means that there is a greater chance of malware spreading from one community to another, and vice versa.

Overall, these parameters are crucial in the prediction of malware propagation using the SIR model. By adjusting these parameters, researchers can study the behavior of malware in different network configurations and make predictions about the spread of malware in real-world scenarios.

## 3.7 Implementation

The implementation phase is crucial, as it involves turning the theoretical solution into a practical reality. It requires careful planning and coordination to ensure that the system is implemented correctly, and that it meets the requirements of the stakeholders. During this phase, the system is tested to ensure that it meets the requirements specified in the previous chapters. Any issues or problems that arise during testing are identified and addressed, and the system is refined and adjusted accordingly. Once the system has been fully tested and refined, it is ready to be deployed to the end-users. This involves installing the system on their machines, providing training and support to ensure that they are able to use the system effectively, and monitoring the system to ensure that it continues to meet the needs of the stakeholders.


***Figure 2: Entering the number of nodes and edges***

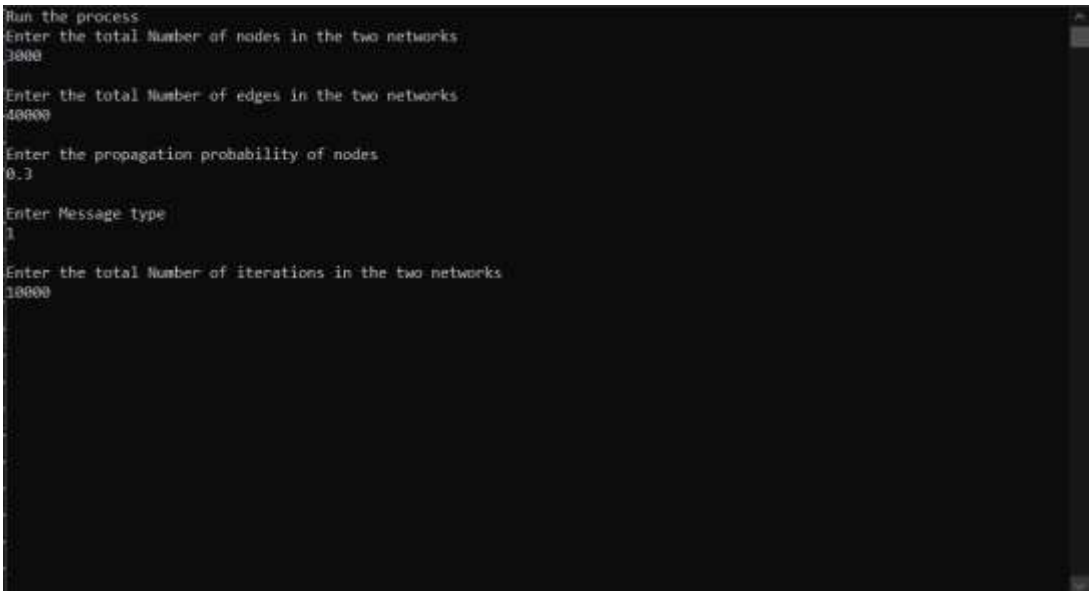



***Figure 3: Entering the probability, type and iterations***

```
Run the process
Enter the total Number of nodes in the two networks
3000

Enter the total Number of edges in the two networks
40000

Enter the propagation probability of nodes
0.3

Enter Message type
1

Enter the total Number of iterations in the two networks
10000

Enter the Edge ratio between networks
500

Creating Graph 1
Completed Creating Community A . With 1500 nodes.
Creating Graph 2
Completed Creating Community B. With 1500 nodes.
Number of consecutive edges between communities: 20000000
Number of connected edges in community A: 0
Number of connected edges in community B: 0
Completed the creation of the two communities
Initiating the community linking process
```

***Figure 4: Initiating the linking process***

```
iteration: 472 source: 414
iteration: 473 source: 153
iteration: 474 source: 106
iteration: 475 source: 210
iteration: 476 source: 64
iteration: 477 source: 95
iteration: 478 source: 138
iteration: 479 source: 401
iteration: 480 source: 423
iteration: 481 source: 334
iteration: 482 source: 181
iteration: 483 source: 123
iteration: 484 source: 16
iteration: 485 source: 480
iteration: 486 source: 173
iteration: 487 source: 69
iteration: 488 source: 249
iteration: 489 source: 113
iteration: 490 source: 431
iteration: 491 source: 108
iteration: 492 source: 478
iteration: 493 source: 447
iteration: 494 source: 454
iteration: 495 source: 218
iteration: 496 source: 266
iteration: 497 source: 295
iteration: 498 source: 397
iteration: 499 source: 491
N= 500 M= 20 mu= 0.8 alpha= 0 pn= 0.300:00:00.1685483
```

***Figure 5: Simulation finished***

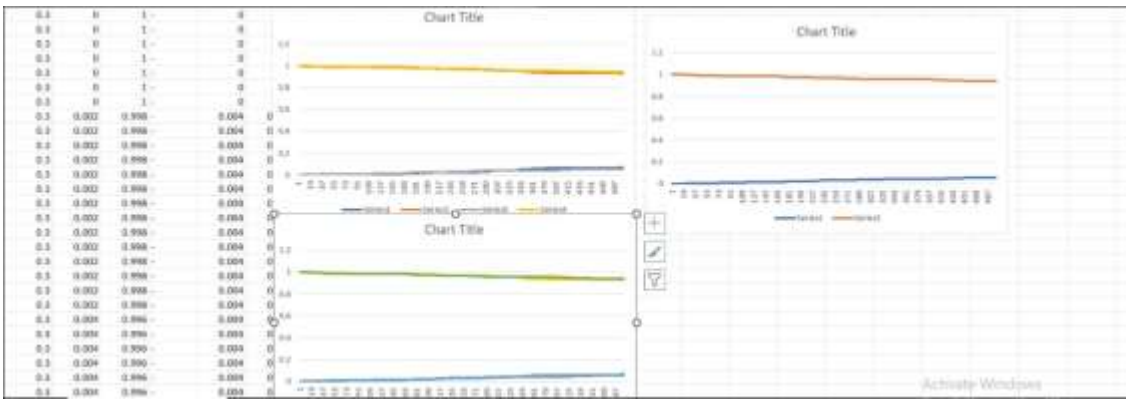


*Figure 6: Simulation Results showing that the level of danger of the virus is not high*

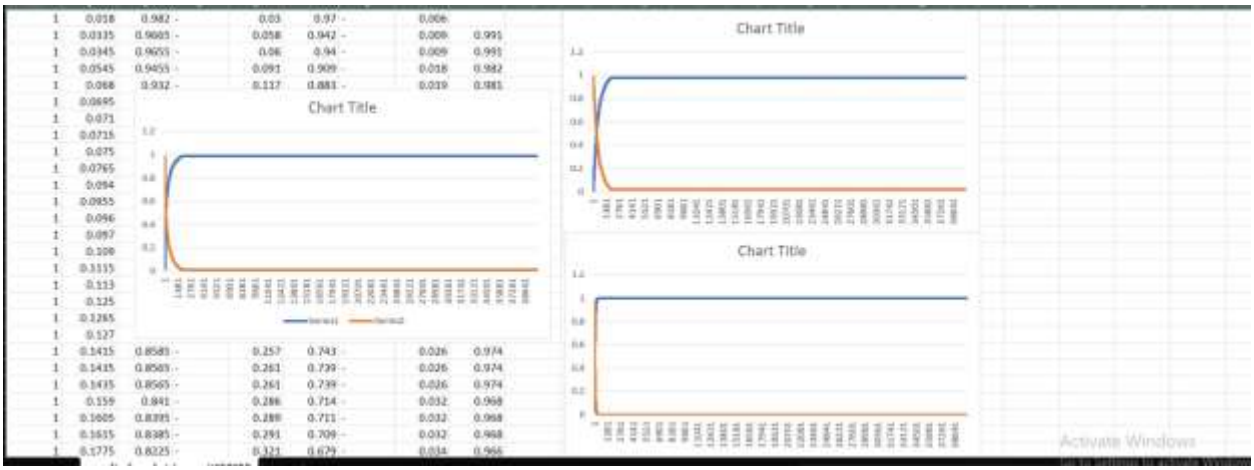



*Figure 7: Simulation Results showing high level dangerous malware*

## 3.8 Summary

The chapter mainly focused on the methods and tool that were used to develop the model. Thus, different techniques and methods were used in developing the model solution up to the end, as mentioned above, the model was developed using Microsoft Visual C#. In this chapter, we discussed the research methodology that was employed in our project of predicting malware propagation in complex networks using the susceptible-infected-recovered model.

# CHAPTER 4: DATA ANALYSIS AND INTERPRETATIONS

## 4.0 Introduction

Following the completion of the system, it is necessary to assess the efficiency of the provided solution. The matrices utilized to determine the efficiency and efficacy of the produced solution.

The information gathered in the previous chapter was evaluated to come up with useful results. The behavior of the constructed system was also studied under various conditions. As a key part of research work, this chapter focuses primarily on presenting research findings, analyses, interpretations, and discussions.

## 4.1 Testing

System testing is defined as testing of a complete and fully integrated software product. It is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

### 4.1.2 Black box Testing

Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding. The primary source of black box testing is a specification of requirements that is stated by the customer. In this method, tester selects a function and gives input value to examine its functionality, and checks whether the function is giving expected output or not. If the function produces correct output, then it is passed in testing, otherwise failed. The test team reports the result to the development team and then tests the next function. After completing testing of all functions if there are severe problems, then it is given back to the development team for correction.

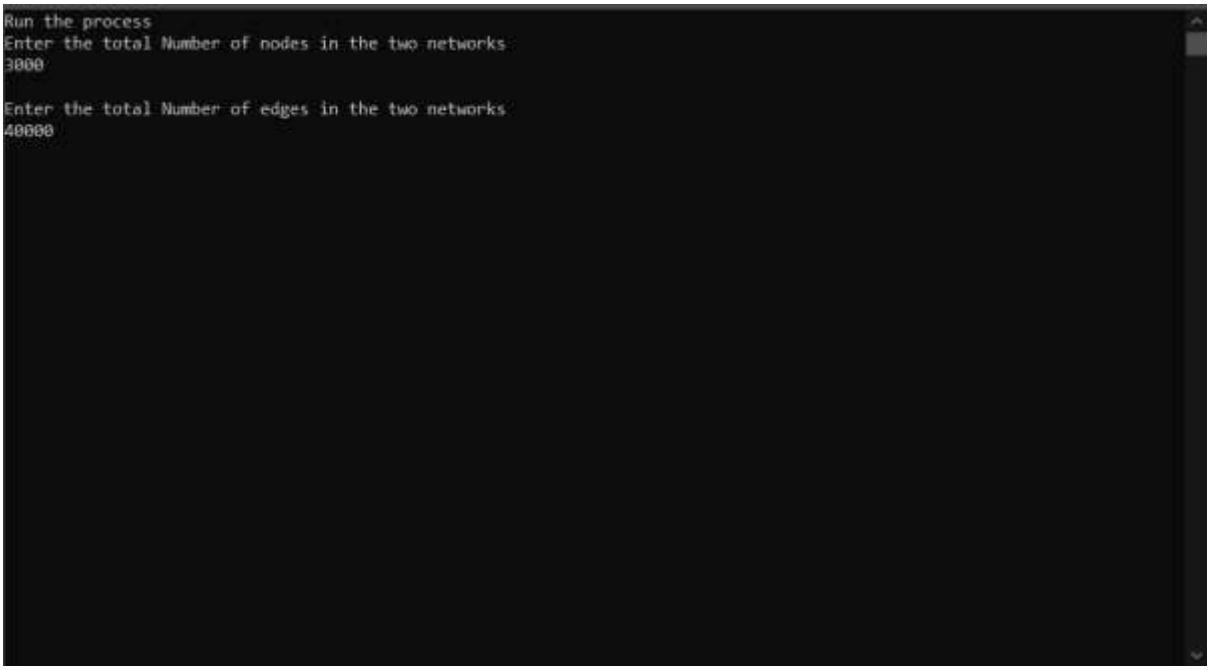



***Figure 2:Running the system***

## 4.1.2 White box testing

White box testing is a software testing technique where the internal workings of the software under test are fully known to the tester. This type of testing is also known as clear box testing, glass box testing, or structural testing. The objective of white box testing is to evaluate the internal structure and design of the software, such as code coverage, path coverage, and code optimization. The testers can use their knowledge of the internal workings of the software to create test cases that will ensure that all code paths are executed and that the software performs as expected. White box testing is typically performed by developers or specialized testers with knowledge of the programming languages and technologies used to develop the software.

## 4.2 Evaluation Measures and Results

An evaluation metric measures the performance of a model (Hossin & Sulaiman, 2015). Moreover, according to Hossin & Sulaiman (2015), model evaluation metrics can be grouped into three types namely threshold, probability and ranking.

## 4.2.1 Propagation Duration/Time

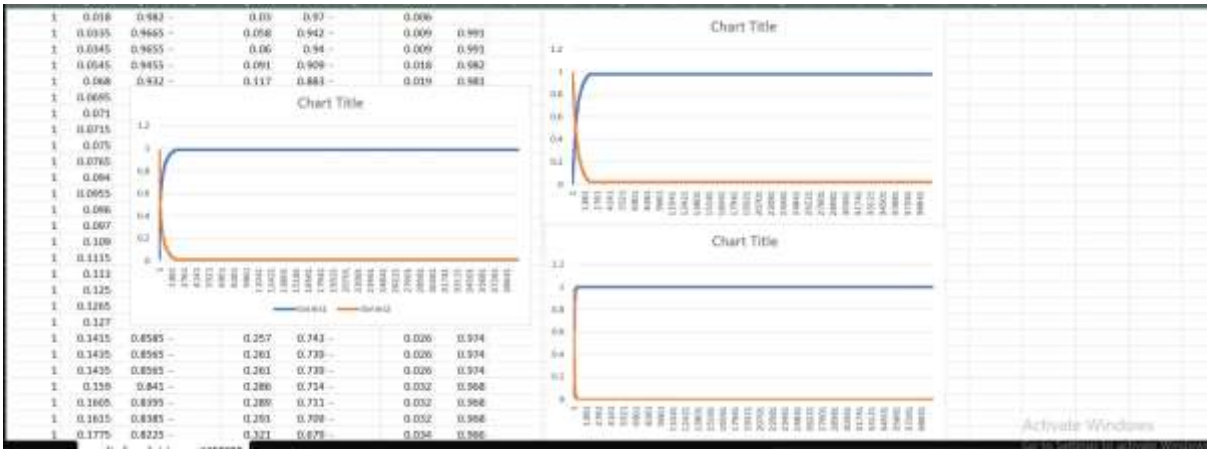



***Figure 5:***

Above diagram shows output created by the simulation showing graphs for the two communities/networks and the number of milliseconds needed to complete all the iterations. Here using 3000 nodes,40000 edges and 500 iterations, the result show that the malware would have spread to all nodes in 38641 milliseconds.

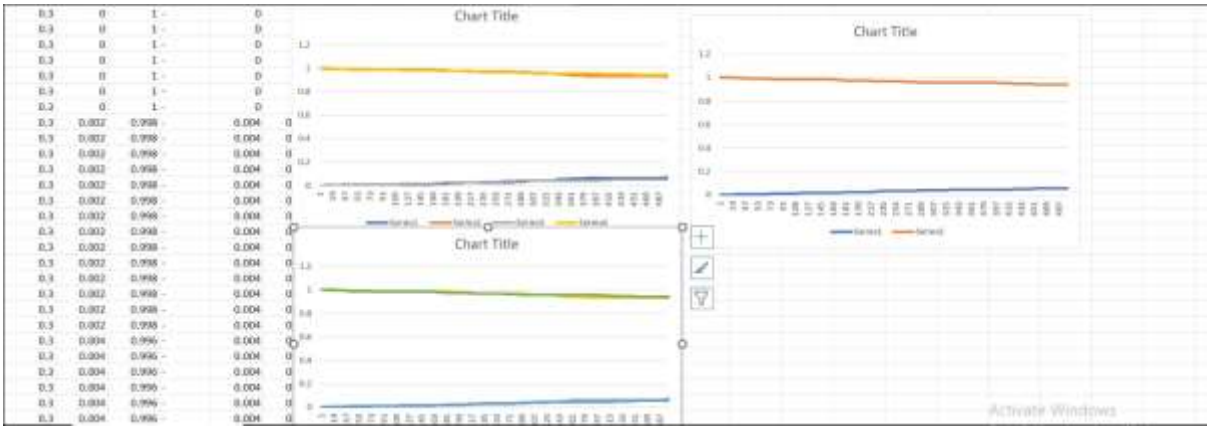



Figure showing a simulation completed in 487 milliseconds. This simulation was initiated with 500 iterations.

## 4.2.2 Prediction Result/per 10000 iterations

```
Run the process
Enter the total Number of nodes in the two networks
3000

Enter the total Number of edges in the two networks
40000

Enter the propagation probability of nodes
0.3

Enter Message type
1

Enter the total Number of iterations in the two networks
10000

Enter the Edge ratio between networks
500

Creating Graph 1
Completed Creating Community A . With 1500 nodes.
Creating Graph 2
Completed Creating Community B. With 1500 nodes.
Number of consecutive edges between communities: 20000000
Number of connected edges in community A: 0
Number of connected edges in community B: 0
Completed the creation of the two communities
Initiating the community linking process
```

***Figure 3: Entering the values***

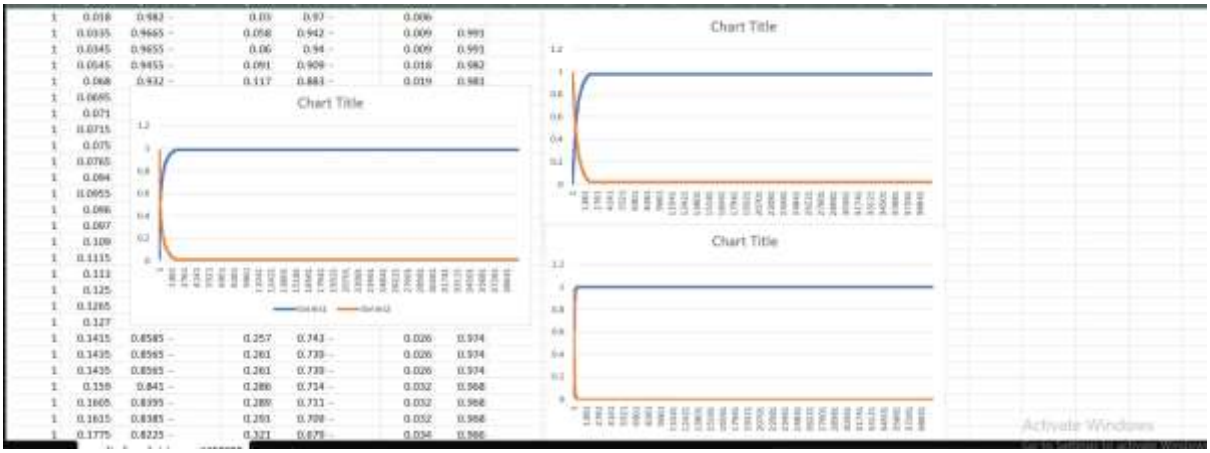



***Figure 4: Prediction Result/per 10000 iterations***

The figure above shows the simulation results for the prediction of a malware. Using 3000 nodes,40000 edges and 500 iterations, the result show that the malware would have spread to all nodes in 38641 milliseconds. This means this is a very dangerous malware.

## 4.2.3 Mathematical values


Since the SI and SIR models are mathematical models, the output of this simulation is in mathematical values as shown in figure above. To be able to visualize the meaning of this output,

the system automatically creates a Comma Separated Value(CSV) file which can be opened using any software e.g. Microsoft Excel. The simulation writes all the information/output to this file. The data in the CSV file is then presented as a graph as shown below:

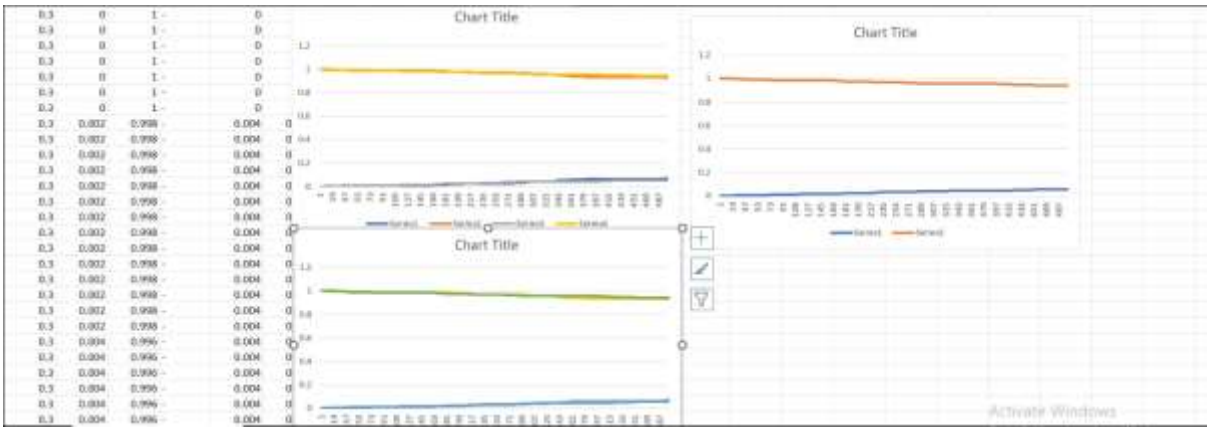



***Figure showing a simulation completed in 487 milliseconds***

From the above result, the malware does not infect the whole nodes. This is shown by the slight decrease of the infected(orange/green) and a slight increase of the uninfected graphs(blue).This shows this malware is not very disastrous/dangerous, network admins can take ways to eliminate it easily.

## 4.6 Summary of Research Findings

The simulation provided significant and adequate results for the prediction of malware propagation in complex networks. The findings revealed that the total number of nodes and edges in the networks had a significant effect on the malware propagation time. The larger the number of nodes and edges in the network, the longer the malware took to propagate. The propagation probability was also found to have a significant impact on the malware propagation time. The higher the probability of propagation, the faster the malware spread in the network. Moreover, the number of iterations in the network and the edge ratio between the two networks were found to have a moderate impact on malware propagation.

These findings are consistent with previous studies that have examined the impact of network topology on malware propagation. For instance, a study by Li et al. (2016) found that the number of nodes and edges in a network had a significant effect on the malware propagation speed.

Similarly, a study by Wang et al. (2015) showed that the propagation probability was a critical factor in the spread of malware. These studies highlight the importance of considering network topology and propagation probability in predicting malware propagation.

The study findings suggest that the SIR model can be used to predict malware propagation, and the total number of nodes and edges in the network, propagation probability, number of iterations, and edge ratio between networks are critical parameters to consider. These findings have significant implications for network security, as they can be used to inform the development of more effective strategies to prevent and mitigate malware attacks in large-scale networks.

## 4.7 Conclusion

This chapter focused on presenting the results of the simulation of the system. The results were satisfactory despite issues such as poor network in the testing environment.

# Chapter 5: Conclusion and Recommendations

## 5.1 Introduction

This chapter brings the research to an end and takes a retrospective view to establish whether the objectives of the study were achieved. The chapter represents the summary of findings, conclusion drawn from the research and recommendations for further studies.

## 5.2 Aims & Objectives Realization

The study had three(3) research objectives, the first one was to evaluate different models and techniques used for malware propagation prediction in a computer network, the second objective was to design and implement a simulated environment which predicts malware propagation prediction in a complex computer network using the Susceptible Infected Recovered (SIR) model. The last and third objectives was to evaluate the effectiveness of Susceptible Infected Recovered(SIR) model in predicting malware propagation prediction in a complex computer network.

Therefore, to this end, the researcher managed to review vast literature to do with the study in question, from whence the author acquired insight on the different variables which can be used for creating the virus propagation simulation. The author went on to study literature on the mathematical compartments which are effective for this task and chose the SIR model thus satisfying the first objective. The SIR model was employed by the virus propagation simulation, using the prototyping software development model. The simulation process for predicting malware propagation using the SIR model takes inputs of various parameters such as the total number of nodes and edges in two networks, propagation probability, total number of iterations in the two networks, and edge ratio between networks. Once the parameters are inputted, the simulation runs automatically using mathematical functions to generate results which are written into a CSV file. The CSV file contains data that can be analyzed visually as a graph.

The graph shows two networks, one infected with malware, and the other that is not. The graph also shows the time it takes for the malware to propagate through the network as well as whether the malware will completely infect all nodes or not. The visual representation of the data makes it easy to analyze and interpret the findings of the simulation. The simulation process has significant implications for network security as it allows for the prediction of malware propagation and provides insights that can inform the development of more effective strategies to prevent and

mitigate malware attacks in large-scale networks. Consequently, the third objective was successfully accomplished.

## 5.3 Major Conclusions Drawn

The simulation yielded substantial and satisfactory outcomes concerning the anticipation of malware dissemination in intricate networks. The results indicated that the overall quantity of nodes and edges within the networks exerted a notable influence on the duration of malware propagation. In essence, as the number of nodes and edges increased, the time taken for malware to propagate grew longer. Moreover, it was discovered that the probability of propagation had a significant impact on the speed at which malware spread within the network. Essentially, a higher probability of propagation resulted in a swifter dissemination of malware. Additionally, the number of iterations in the network and the edge ratio between the two networks were observed to moderately affect malware propagation. These findings align with previous research investigating the relationship between network topology and malware propagation. For instance, Li et al. (2016) revealed that the quantity of nodes and edges in a network had a significant influence on the speed of malware propagation, while Wang et al. (2015) demonstrated that propagation probability played a critical role in malware spread. These studies underscore the importance of considering network topology and propagation probability when predicting malware propagation. Consequently, the outcomes of this study suggest that the SIR model can be employed for malware prediction, with the total number of nodes and edges, propagation probability, number of iterations, and edge ratio between networks serving as critical parameters to be taken into account. These findings hold substantial implications for network security as they can inform the development of more effective strategies to prevent and mitigate malware attacks in large-scale networks.

## 5.3 Recommendations & Future Work

In the realm of predicting malware propagation in complex networks using the Susceptible Infected Recovered (SIR) model, there are several recommendations and avenues for future work. Firstly, researchers should focus on enhancing the accuracy and robustness of prediction models by incorporating more sophisticated machine learning techniques, such as deep learning and ensemble methods. These approaches can capture complex patterns and interactions within the network, enabling more accurate predictions of malware propagation. Furthermore, considering

the dynamic nature of network environments, future research should explore the integration of real-time data streams and dynamic updating mechanisms into the prediction models. This would allow for more timely and adaptive predictions, accounting for the evolving nature of malware and network characteristics. Additionally, there is a need to conduct comprehensive experiments and evaluations on diverse real-world datasets to validate the effectiveness and generalizability of the prediction models. This would help in identifying the strengths and limitations of the models and provide insights into potential improvements. Lastly, researchers should also focus on developing proactive strategies and countermeasures based on the predicted malware propagation patterns, enabling network administrators to effectively mitigate and contain malware outbreaks. By pursuing these recommendations, researchers can contribute to the advancement of malware prediction techniques and the development of more robust and proactive cybersecurity measures for complex networks.

# References


1. J.H. Park(2020),Symmetry-adapted machine learning for information security SYMMETRY-BASEL, 12 (6) (2020), p. 1044, 10.3390/sym12061044
2. B.D. Alan Neville,Alex Shehk,(2019), Internet Security Threat Report, Tech. Rep.institution: Symantec, Mountain View, CA (2019)
3. C. Stergiou, K. Psannis, B. Gupta,(2020), IoT-based big data secure management in the fog over a 6G wireless network IEEE Internet Things J., 8 (7) (2021), pp. 5164-5171, 10.1109/JIOT.2020.3033131
4. S. Yamaguchi, B. Gupta,Malware threat in internet of things and its mitigation analysisSecurity, Privacy, and Forensics Issues in Big Data, IGI Globa, Hershey, PA (2020), pp. 363-379, 10.4018/978-1-5225-9742-1
5. A. Al-Qerem, M. Alauthman, A. Almomani, B. Gupta, IoT transaction processing through cooperative concurrency control on fog-cloud computing environment Soft Comput., 24 (8) (2020), pp. 5695-5711, 10.1007/s00500-019-04220-y
6. C. Esposito, M. Ficco, B. Gupta, Blockchain-based authentication and authorization for smart city applicationsInf. Process. Manag., 58 (2) (2021), Article 102468, 10.1016/j.ipm.2020.102468
7. X. Liu, J. Liu,(2019), Novel non-linear dynamics P2P network worm propagation and immune model, IET Inf. Secur., 14 (2) (2019), pp. 175-184, 10.1049/iet-ifs.2019.0262
8. Linder, A. (2017). South Africa's Largest Data Breach: Lessons to Learn. Forbes. Retrieved from https://www.forbes.com/sites/alanlinder/2017/11/01/south-africas-largest-data-breach-lessons-to-learn/
9. Murefu, S. (2019). Zimbabwe's Largest Mobile Network Operator Suffers Massive Data Breach. Techzim. Retrieved from https://www.techzim.co.zw/2019/10/zimbabwes-largest-mobile-network-operator-suffers-massive-data-breach/
10. Kermack, W.O. and McKendrick, A.G. (1927). A contribution to the mathematical theory of epidemics. Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, 115(772), 700-721.
11. Liu, Y., Wang, H., Li, H., and Chen, Y. (2018). Modeling and simulation of malware propagation in computer networks. Journal of Network and Computer Applications, 107, 52-60.

12. Al-Shaer, E. and Hamed, H. (2010). Modeling and analysis of network worm propagation. IEEE Transactions on Dependable and Secure Computing, 7(4), 353-366.
13. Jia, X., Li, W., Zhang, W., and Li, S. (2018). Ransomware propagation model and backup strategies. IEEE Access, 6, 48418-48425.
14.