Modeling the detection process of polymorphic malware based on the Lotka-Volterra model*

Maksym Chaikovskyi[1,*,†], Inna Chaikovska[1,†], Tomas Sochor[2,†], Inna Martyniuk [1,†] and Oleksii Lyhun[1,†]

[1] *Khmelnytskyi National University, Instytuts'ka Str. 11, 29000, Khmelnytskyi, Ukraine*
[2] *Prigo University, Havirov, Czech Republic*

Abstract

The article proposes the use of the Lotka-Volterra model ("predator-prey" model) for modeling the process of detecting polymorphic malware. It is proposed to consider α as the probability that the number of polymorphic viruses will increase; β - the probability that polymorphic viruses of different levels of complexity will be detected using the selected methods, technologies and tools; γ - the probability that some of the selected methods, technologies and tools will not be effective in detecting polymorphic viruses of different levels of complexity as a result of the appearance of new varieties; δ - the probability that polymorphic viruses of different levels of complexity will require the complex use of selected methods, technologies and tools, as well as the latest approaches; x - quantitative measurement of polymorphic viruses at time t; y is a quantitative measure of the available technologies, methods and tools for detecting polymorphic viruses at time t. The influence of input indicators on the maximum rate of spread and detection of polymorphic viruses in its fluctuating process was studied. This approach confirms the feasibility of using a set of methods to detect polymorphic malware: string search algorithms, intelligent data analysis, sandbox analysis, machine learning, the method of developing structural functions, probabilistic logical networks.

Keywords

polymorphic malware, detection probability of polymorphic malware, Lotka-Volterra model

1. Introduction

The use of tools and techniques to detect polymorphic malware can be compared to the classic predator-prey model. The Lotka-Volterra model ("predator-prey" model) describes a population consisting of two species that interact with each other. Victims die out at a rate equal to the number of encounters between predators and prey, which is proportional to the size of both populations. Predators reproduce at a rate that is proportional to the amount of prey eaten by the predators. The system of equations that describes such a population is called the Lotka-Volterra model. According to the conditions of the model, the victims eat the plants, and the predators eat the victims. We will use this model to simulate the process of detecting polymorphic malware. Polymorphic viruses in a computer system will act as a "victim", tools and methods for detecting polymorphic malware will act as a "predator".

AdvAIT-2024: 1st International Workshop on Advanced Applied Information Technologies, December 5, 2024, Khmelnytskyi, Ukraine - Zilina, Slovakia

* Corresponding author.
† These authors contributed equally.
max.chaikovskyi@gmail.com (M. Chaikovskyi); inna.chaikovska@gmail.com (I. Chaikovska); tomas.sochor@osu.cz (T. Sochor); inmartunyk@ukr.net (I. Martyniuk); oleksii.lyhun@gmail.com (O. Lyhun)
0000-0002-9596-6697 (M. Chaikovskyi); 0000-0001-7482-1010 (I. Chaikovska); 0000-0002-1704-1883 (T. Sochor); 0009-0007-7751-8974 (I. Martyniuk); 0009-0004-5727-5096 (O. Lyhun)

2. Literature review

The problem of detecting malicious software is quite relevant and a significant amount of research by scientists is devoted to it.

The work [1] reflects a comprehensive modern review of research on the malware detection model.

The paper [2] proposes an intelligent agent system for detecting DDoS attacks using automatic feature selection and selection.

In [3], it is stated that detection of malicious traffic in computer systems and improvement of security of computer networks is possible using the results of analysis and detection of malicious programs using machine learning algorithms to calculate the difference in correlation symmetry.

The use of machine learning is also proposed in [15].

The study [4] proposed an approach that takes advantage of the deep transfer methodology and includes a fine-tuning method and various combination strategies to improve detection and classification performance without the need to develop training models from scratch.

In [6], malicious software is detected with the help of convolutional neural networks (CNN), in [9] with the help of machine learning algorithms.

The work [10] compares the methods of detecting malicious programs based on static, dynamic and hybrid analysis.

In work [12] proposes a new systematic approach to identifying modern malware using dynamic deep learning-based methods combined with heuristic approaches to classify and detect five modern malware families: adware, Radware, rootkit, SMS malware, and ransomware.

The work [13] proposes an integrated framework for implementing IoT with blockchain technology to guarantee high level of security and validation process based on the integration between consensus algorithms of blockchain (PBFT and Tangle).

In [14], a conceptual model of multi-computer systems was developed, which is designed to ensure the functioning of anti-virus baits and traps for detecting malicious programs.

In works [17, 18] proposed a novel detection approach by generating structural features through computing a stream of byte chunks using compression ratio, entropy, Jaccard similarity coefficient and Chi-square statistic test.

The paper [20] presents an approach to the detection of metamorphic viruses based on the analysis of its obfuscation features.

In [21], the K-NN algorithm was used to detect malicious software. In [22], a support vector machine (SVM) model was used to detect malicious software. Dynamic Malware Analysis with Reinforcement Learning was carried out in [24].

In [25], a method for determining the effectiveness of a distributed system for detecting anomalous manifestations is proposed. In work [26], a method for detecting unknown metamorphic viruses is proposed, which is based on the analysis of potentially suspicious behavior of programs on the host, and in work [27] - a method for detecting metamorphic viruses, based on the search for equivalent functional blocks.

The paper [28] addresses the challenges associated with App-DDoS detection and presents a highly effective and adaptable solution for detecting various types of App-DDoS attacks.

So, you can see quite a wide selection of methods for detecting malware. One of these models is also the Lotka-Volterra model ("predator-prey" model) [5, 23], which has found wide use in various areas of our life: in space research [7], biology [8, 11], in many in the fields of engineering [16], medicine [19], security assessment of cyber-physical systems [29]. However, the use of this model for researching the process of identifying polymorphic software is quite appropriate and relevant, which is why this study is devoted to it.

3. Methodology

Consider the classic Lotka-Voltaire model and its adaptation to the process of detecting polymorphic malware.

3.1. The classic Lotka-Volterra model

In general, the model of interspecific competition looks as follows:

$$\begin{cases} \frac{dx}{dt} = (\alpha - \beta y)x \\ \frac{dy}{dt} = (-\gamma + \delta x)y \end{cases} \quad (1)$$

where x is the number of victims; y is the number of predators; t – time; $\alpha, \beta, \gamma, \delta$ are coefficients that reflect the interaction between species.

3.2. Adaptation of the model to study the process of detection of polymorphic malware

In the case of adaptation of the model to simulate the polymorphic malware detection process, α, β, γ, δ can display the following: α is the probability that the number of polymorphic viruses will increase; β is the probability that polymorphic viruses of different levels of complexity will be detected using the selected methods, technologies and tools; γ is the probability that some of the selected methods, technologies and tools will not be effective in detecting polymorphic viruses of different levels of complexity as a result of the appearance of new varieties; δ is the probability that polymorphic viruses of different levels of complexity will require the complex use of selected methods, technologies and tools, as well as the latest approaches; x - quantitative measurement of polymorphic viruses at time t; y is a quantitative measure of the available technologies, methods and tools for detecting polymorphic viruses at time t.

It immediately follows from the system that if there are no polymorphic viruses (x = 0), then the number of necessary methods, technologies and tools for their detection will decrease exponentially with a certain initial coefficient (γ according to formula 1).

$$\dot{y} = -\gamma \cdot y \rightarrow y = C_1 \cdot e^{-\gamma \cdot t}, C_1 \epsilon R., \quad (2)$$

A similar situation is obtained in the complete absence of methods, technologies and tools for detecting polymorphic viruses (y = 0):

$$E\dot{x} = \alpha \cdot x \rightarrow x = C_2 \cdot e^{\alpha \cdot t}, C_2 \epsilon R., \quad (3)$$

This equation (3) is sometimes called the Malthus equation.

Therefore, the growth of polymorphic viruses is exponential with a certain, predetermined constant (α).

It is worth noting that the Lotka-Volterra model makes several assumptions:

1. There is a constant appearance of polymorphic viruses.
2. Polymorphic viruses, as well as their detection technologies, are in the computer system.
3. Only the presence of polymorphic viruses and their detection technologies in the computer system is taken into account.

Let's find special points possessed by the system:

$$E\begin{cases}(\alpha-\beta y)x=0\\(-\gamma+\delta x)y=0\end{cases}\rightarrow\begin{cases}\alpha x=\beta xy\\\gamma y=\delta xy\end{cases}\rightarrow\begin{cases}y(0)=\dfrac{\alpha}{\beta}\\x(0)=\dfrac{\gamma}{\delta}.\end{cases} \quad (4)$$

It is clear that when x (0) = 0, y (0) = 0, the special point will be precisely (0, 0), but this case is not interesting, because at the zero moment of time there are no polymorphic viruses and technologies for their detection and, logically , no longer appear.

Much more interesting things happen in the nonzero case. Depending on the initial parameters, a special point will change - such a number of viruses and their detection technologies, when both indicators remain unchanged and balanced.

If the initial condition does not fall into a special point, the phase curves will be located around it, forming an infinite cyclic oscillation, which was exactly what Lotka and Volterra were talking about. That is, the number of polymorphic viruses will grow, and the number of effective methods for their detection will fall, then vice versa, and so on for an unlimited amount of time (within reasonable limits, of course).

3.3. Stages of the proposed integrated approach to detection, analysis and classification of polymorphic malware

This approach is the second stage in the proposed comprehensive approach to detection, analysis and classification of polymorphic malware (Figure 1).

4. Experiments

Consider the implementation of the "predator-prey" model for modeling the process of detecting polymorphic malware using the Lotka-Volterra equation solver [30].

The following scale is used to denote x and y parameters (table 1). The β indicator was formed on the basis of previous studies on the effectiveness of the complex use of the above methods for detecting polymorphic malware.

4.1. Experiment 1

Experiment 1 (2 methods were used to detect polymorphic viruses) involves the following input parameters (Figure 2, 3): α=0.2; β=0.3 (2 methods were used to detect polymorphic viruses); γ=0.7; δ=0.3; x=1; y=1; max_time = 100 (seconds); t = 1.

Table 1
Point Scale for Input Parameters

Ball scale	x	y	β
to 1	Polymorphic viruses of the 1st level of complexity	1 method used (string search algorithm)	0.1
2	Polymorphic viruses of the 2nd and lower levels of complexity	2 methods were used (string search algorithm + 1 of the methods (intelligent data analysis, sandbox analysis, machine learning, structural function development method)	0.3
3	Polymorphic viruses of the 3rd and lower levels of complexity	3 methods were used (string search algorithm + 2 of the methods (intelligent data analysis, sandbox analysis, machine learning, structural function development method)	0.4

4	Polymorphic viruses of the 4th and lower levels of complexity	4 methods were used (string search algorithm + 3 methods (intelligent data analysis, sandbox analysis, machine learning, structural function development method)	0.5
5	Polymorphic viruses of the 5th and lower levels of complexity	5 methods were used (row search algorithm, intelligent data analysis, sandbox analysis, machine learning, structural function development method)	0.6
6 or more	Polymorphic viruses of the 6th and lower levels of complexity	6 or more methods are used (string search algorithm, intelligent data analysis, sandbox analysis, machine learning, structural function development method, probabilistic logic networks)	0.9

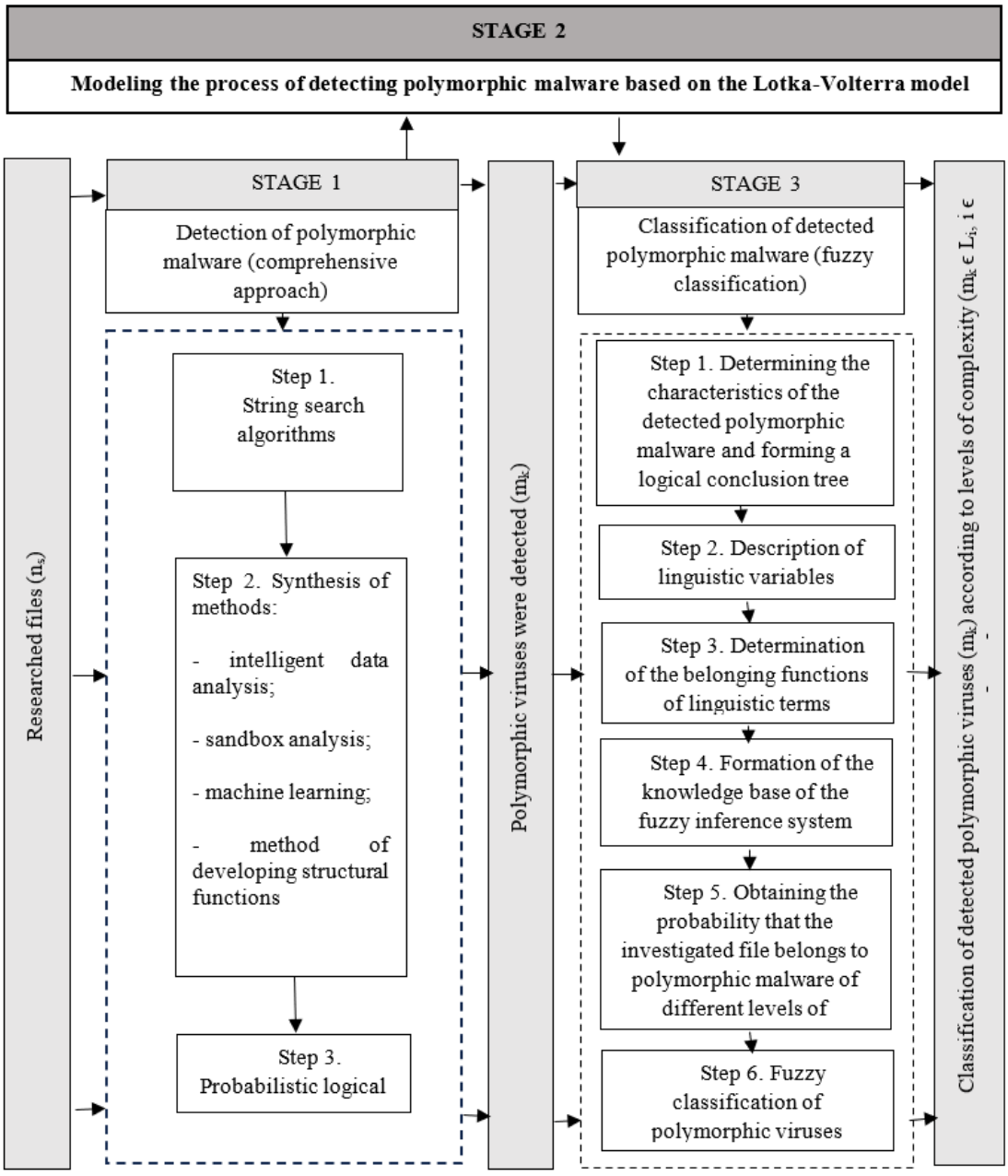

Figure 1: A comprehensive approach to detection, analysis and classification of malicious software.

It can be seen in Figures 2, 3 that the process is oscillatory. With the same initial values of the number of polymorphic viruses and methods of their detection on a point scale at the level of 1 point. Under these input values, the number of polymorphic viruses increases, and the number and

efficiency of polymorphic virus detection methods decreases. When the value of y reaches β = 0.3, partial detection of polymorphic viruses occurs and their number begins to decrease.

The decrease in the number of polymorphic viruses after a certain time begins to be affected by y, and the number of polymorphic viruses reaches the value (in point expression) γ/δ=0.7/0.3=2.33, the number of methods used to detect polymorphic malware also begins to decrease along with by reducing the number of polymorphic viruses. The decrease in the number of polymorphic viruses and methods of its detection decreases until y reaches the value α/β = 0.2/0.3=0.66. At this moment, the number of polymorphic viruses begins to increase, and after a certain period of time and methods of their detection. This process is constantly repeated with a certain period.

The periodicity of the process can be clearly observed in the pictures. The number of polymorphic viruses and their detection methods fluctuates around the values of x = 2.33, y = 0.66, respectively.

The periodicity of the process is well observed on the phase curve (x(t), y(t)), which is a closed line. The extreme left point of this curve is the point at which the number of polymorphic viruses reaches its minimum value, and the extreme right point - the maximum. Between these points, the number of effective detection methods first decreases to the lower point of the phase curve and then increases to the upper point of the phase curve. The phase curve covers the point x = 2.33 and y = 0.66. At this point, the system has a stationary state (dx/dt=0, dy/dt=0). If at the initial moment the system was at this point, then over time x(t) and y(t) will not change and will remain constant, in all other cases an oscillatory process will be observed. Based on these initial values, the maximum value of polymorphic malware detection methods (in terms of points) will be 2.33 points.

It can be seen that the selected virus detection methods are not effective and lead to the spread of viruses to the level of almost 5 points.

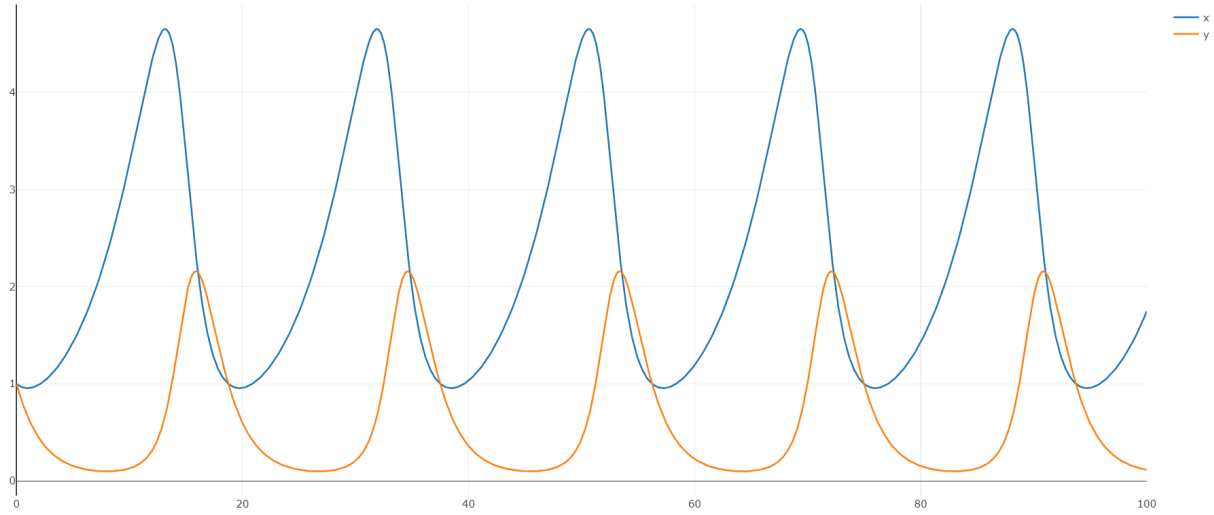

Figure 2: Temporal functions of the "predator-prey" system (x-axis – time, y-axis – point scale), experiment 1

4.2. Experiment 2

Experiment 2 (5 methods were used to detect polymorphic viruses) involves the following input parameters (Figure 4, 5): α=0.2; β=0.6 (5 methods were used to detect polymorphic viruses); γ=0.2; δ=0.3; x=1; y=1; max_time = 100 (seconds); t = 1.

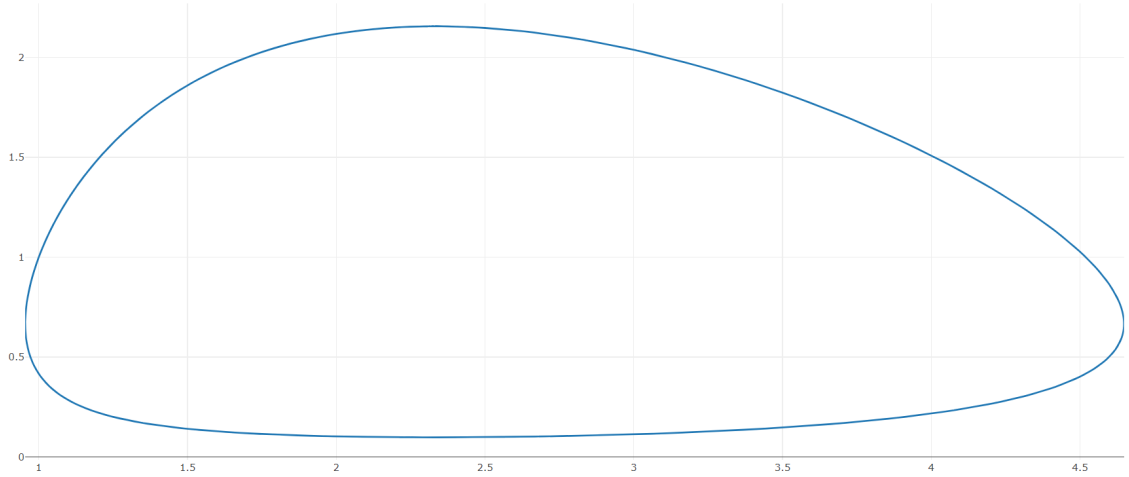

Figure 3: Phase portrait of the predator-prey system, experiment 1.

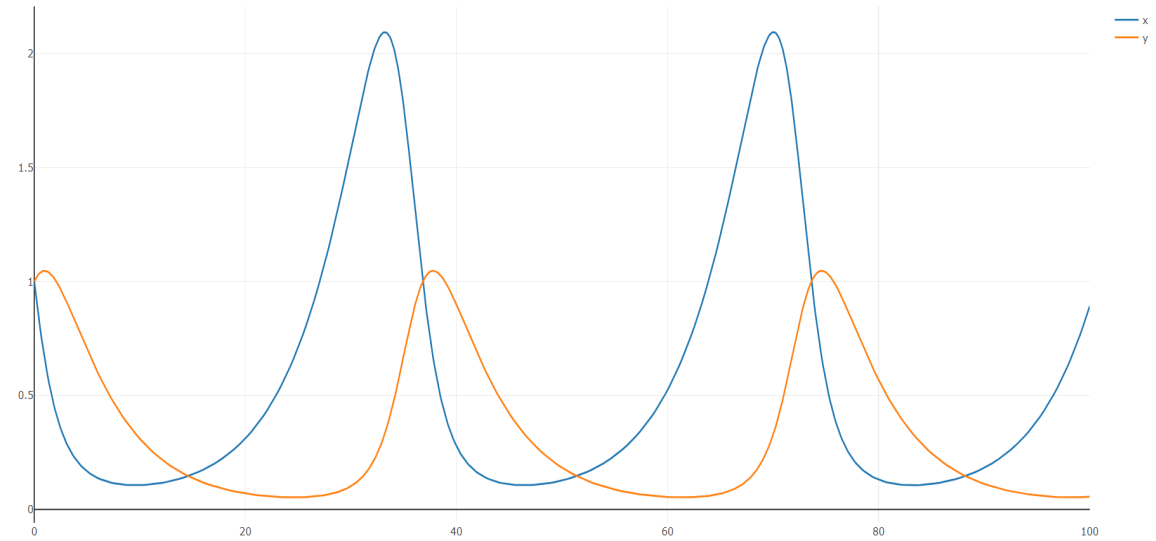

Figure 4: Temporal functions of the "predator-prey" system (x-axis – time, y-axis – point scale), experiment 2.

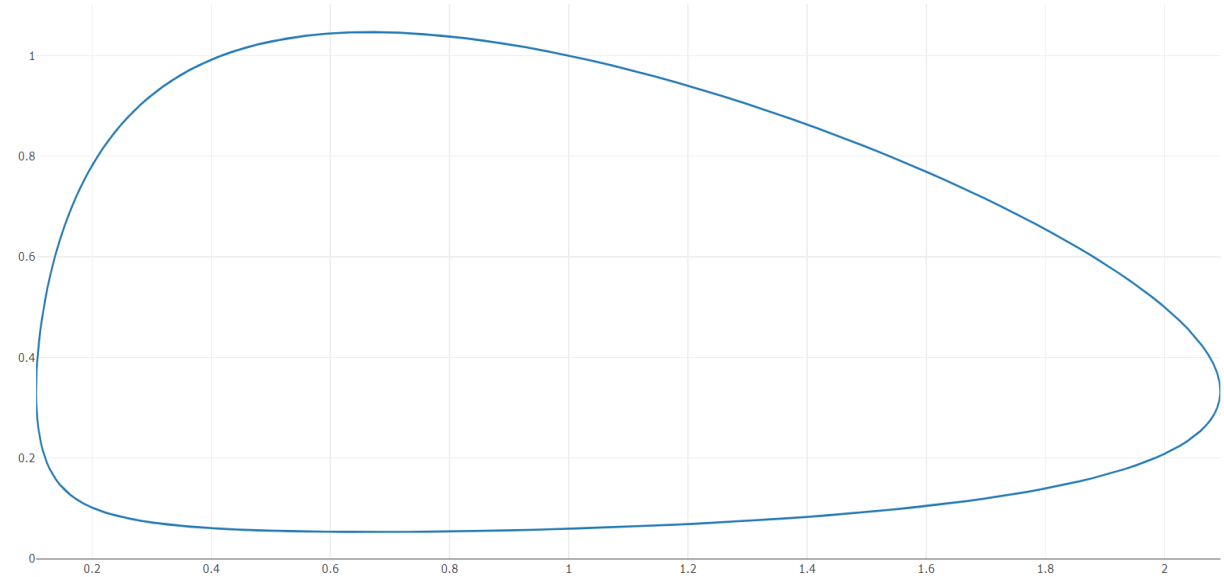

Figure 5: Phase portrait of the predator-prey system, experiment 2.

It can be seen that these virus detection methods (5) are effective and lead to a spread of viruses slightly more than 2 points.

4.3. Experiment 3

Experiment 3 (6 methods were used to detect polymorphic viruses) involves the following input parameters (Figure 6, 7): α=0.5; β=0.9 (6 methods were used to detect polymorphic viruses); ɣ=0.3; δ=0.7; x=1; y=1; max_time = 100 (seconds); t = 1. The selected virus detection methods (6) are effective and result in a virus spread of slightly more than 1 point.

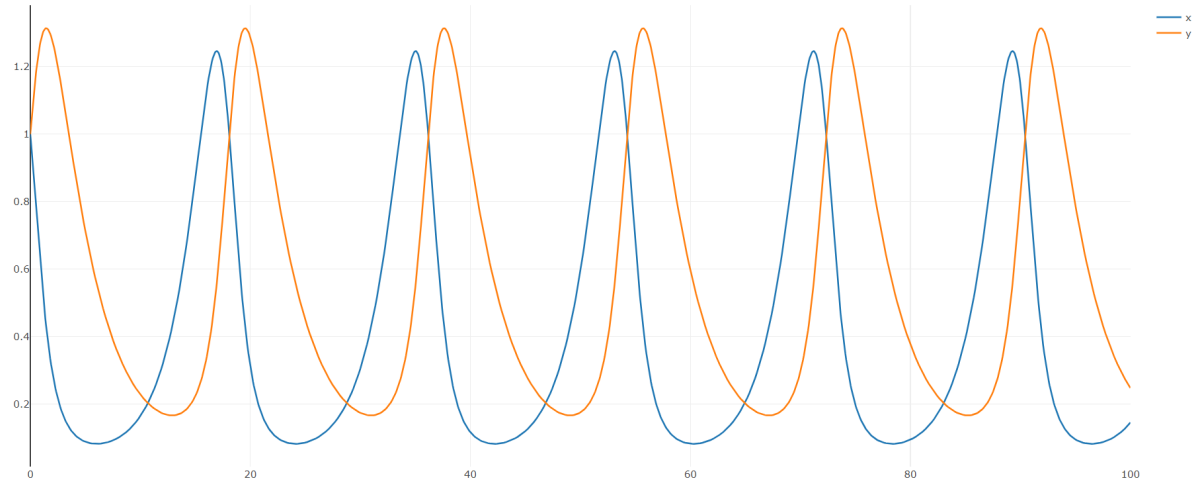

Figure 6: Temporal functions of the "predator-prey" system (x-axis – time, y-axis – point scale), experiment 3.

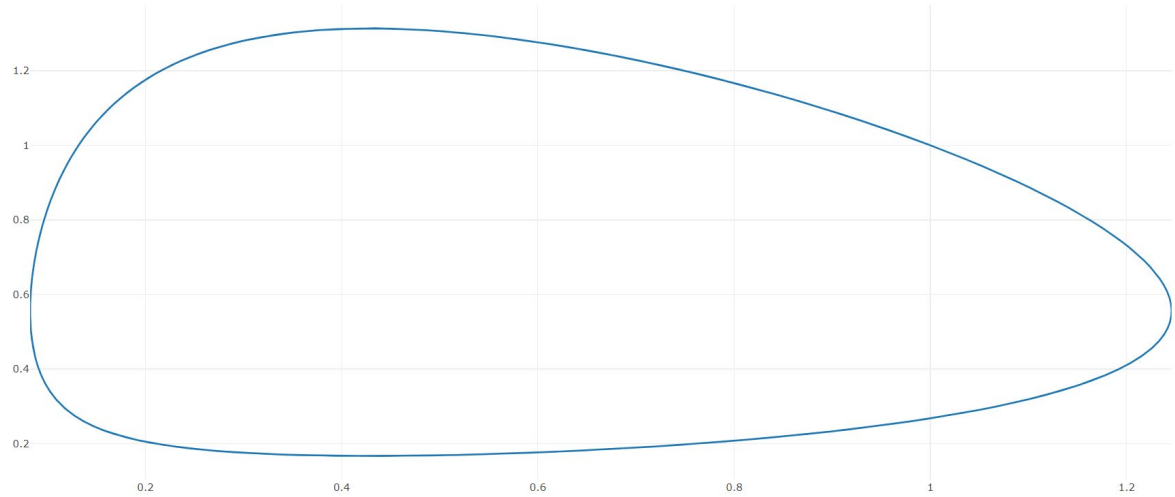

Figure 7: Phase portrait of the predator-prey system, experiment 3.

5. Conclusions

The study proposes the use of the Lotka-Volterra model for modeling the process of detecting polymorphic malware. It is proposed to consider α as the probability that the number of polymorphic viruses will increase; β - the probability that polymorphic viruses of different levels of complexity will be detected using the selected methods, technologies and tools; ɣ - the probability that some of the selected methods, technologies and tools will not be effective in detecting polymorphic viruses of different levels of complexity as a result of the appearance of new varieties; δ - the probability that polymorphic viruses of different levels of complexity will require the complex use of selected methods, technologies and tools, as well as the latest approaches; x - quantitative measurement of

polymorphic viruses at time t; y is a quantitative measure of the available technologies, methods and tools for detecting polymorphic viruses at time t. The influence of input indicators on the maximum rate of spread and detection of polymorphic viruses in its fluctuating process was studied. This approach confirms the feasibility of using a complex of 6 methods to detect polymorphic malware: string search algorithms, intelligent data analysis, sandbox analysis, machine learning, the method of developing structural functions, probabilistic logical networks.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to: grammar and spelling check; DeepL Translate in order to: some phrases translation into English. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] F. A. Aboaoja, A. Zainal, F. A. Ghaleb, B. A. S. Al-rimy, T. A. E. Eisa, A. A. H. Elnour, Malware Detection Issues, Challenges, and Future Directions: A Survey. Applied Sciences, 12(17), (2022). doi: 10.3390/app12178482

[2] R. Abu Bakar, X. Huang, M.S. Javed, S. Hussain, M.F. Majeed, An Intelligent Agent-Based Detection System for DDoS Attacks Using Automatic Feature Extraction and Selection. Sensors, 23, (2023), 3333. doi: 10.3390/s23063333

[3] M. S. Akhtar, T. Feng, Malware Analysis and Detection Using Machine Learning Algorithms. Symmetry (Basel), 14(11), (2022), 2304. doi: 10.3390/sym14112304

[4] S. B. Atitallah, M. Driss, I. Almomani, A novel detection and multi-classification approach for IoT-malware using random forest voting of finetuning convolutional neural networks. Sensors, 22(11), (2022) 4302. doi: 10.3390/s22114302

[5] B. Bonnard, J. Rouot, Feedback classification and optimal control with applications to the controlled Lotka–Volterra model. Optimization, (2024) 1–24. doi:10.1080/02331934.2024.2392209

[6] A. Chakraborty, K. Kriti, Yateendra, M.S. Bennet Praba, Polymorphic Malware Detection by Image Conversion Technique. International Journal of Engineering and Advanced Technology (IJEAT), 9(3), (2020) 2898-2903. doi: 10.35940/ijeat.B4999.029320

[7] Y. Chen, J. Ni, Y.C. Ong, Lotka–Volterra models for extraterrestrial self-replicating probes. The European Physical Journal Plus 137, 1109 (2022). doi:10.1140/epjp/s13360-022-03320-3

[8] M. Clenet, F. Massol, J. Najim, Equilibrium and surviving species in a large Lotka–Volterra system of differential equations. Journal of Mathematical Biology, 87 (2023), 13.

[9] R. Chiwariro, L. Pullagura, Malware Detection and Classification Using Machine Learning Algorithms, International Journal for Research in Applied Science & Engineering Technology, IJRASET, 11 (2023) 1727-1738. doi: 10.22214/ijraset.2023.55255

[10] A. Damodaran, F.D. Troia, C.A. Visaggio, T. H. Austin, M. Stamp, A comparison of static, dynamic, and hybrid analysis for malware detection, J Comput Virol Hack Tech 13 (2017) 1–12. doi: 10.1007/s11416-015-0261-z

[11] J. Davis, D. Olivença, S. Brown, E. Voit, Methods of quantifying interactions among populations using Lotka-Volterra models. Frontiers in Systems Biology, 2, (2022) 1021897. doi: 10.3389/fsysb.2022.1021897

[12] A. Djenna, A. Bouridane, S. Rubab, I.M. Marou, Artificial intelligence-based malware detection, analysis, and mitigation. Symmetry, 15(3), (2023), 677. doi: 10.3390/sym15030677

[13] O. Emam, H. Fahmy, M. Mamdouh, Securing IoT Systems using Blockchain Algorithms. Communications on Applied Electronics, 7(34), (2020) 10-17. doi:10.5120/cae2020652871.

[14] A. Kashtalian, S. Lysenko, O. Savenko, A. Nicheporuk, T. Sochor, V. Avsiyevych, Multi-computer malware detection systems with metamorphic functionality, Radioelectronic and Computer Systems 1 (2024) 152-175. doi: 10.32620/reks.2024.1.13.

[15] A. J. Kurian, A. Santhosh, M. Subin, Enhanced malware detection framework leveraging machine learning algorithms. International Research Journal of Modernization in Engineering Technology and Science 06(03) (2024) 3597-3603.
[16] Y. Lin, Q. Din, M. Rafaqat, A. A. Elsadany, Y. Zeng, Dynamics and Chaos Control for a Discrete-Time Lotka-Volterra Model. IEEE Access, 8, (2020) 126760-126775.
[17] Y. T. Ling, · N. F. M. Sani, · M. T. Abdullah, · N. A. W. A. Hamid, Metamorphic malware detection using structural features andnonnegative matrix factorization with hidden markov model, Journal of Computer Virology and Hacking Techniques 18 (2022)183–203.
[18] Y. T. Ling, N. F. M. Sani, M. T. Abdullah, N. A. W. A. Hamid, Structural Features with Nonnegative Matrix Factorization for Metamorphic Malware Detection, Computers & Security 104, 2 (2021) 102216. doi: 10.1016/j.cose.2021.102216
[19] A. Manikandan, Investigative Study of the Behavior of Lotka-Volterra Model of COVID-19. International Journal of Science and Research (IJSR), 10(11), (2021) 556-558.
[20] G. Markowsky, O. Savenko, S. Lysenko, A. Nicheporuk, The technique for metamorphic viruses' detection based on its obfuscation features analysis, CEUR-WS, 2104 (2018): 680–687.
[21] E. Masabo, et. al., Structural Feature Engineering approach for detecting polymorphic malware, in: Proceedings of the 15-th IEEE Intl Conf on Dependable, Autonomic and Secure Computing, 15-th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress, DASC/PiCom/DataCom/CyberSciTech, 2017, pp. 716-721.
[22] C. B. Nwagwu, O. E. Taylor, N. D. Nwiabu, A Model for Detection of Malwares on Edge Devices. International Journal Of Engineering And Computer Science, 13(07), (2024) 26274-26283.
[23] L. Poley, J. W. Baron, T. Galla, Generalized Lotka-Volterra model with hierarchical interactions. Physical Review E, 107, (2023) 024313. doi: 10.1103/PhysRevE.107.024313
[24] K. Potter, R. Shad, Dynamic Malware Analysis with Reinforcement Learning. Journal of Cyber Security, July 16, (2024). doi: 10.2139/ssrn.4897267
[25] B. Savenko, A. Kashtalian, A method for determining the effectiveness of a distributed system for detecting abnormal manifestations, Computer Systems and Information Technologies 2 (2022) 14–22. doi: 10.31891/csit-2022-2-2 In Ukrainian
[26] O. Savenko, S. Lysenko, A. Nicheporuk, B. Savenko, Approach for the Unknown Metamorphic Virus Detection, in: Proceedings of the 8-th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS, Bucharest, Romania, 2017, pp. 71–76. doi: 10.1109/IDAACS.2017.8095052
[27] O. Savenko, S. Lysenko, A. Nicheporuk, B. Savenko, Metamorphic Viruses' Detection Technique Based on the Equivalent Functional Block Search, CEUR-WS, 1844 (2017): 555–569.
[28] D. M. Sharif, H. Beitollahi, Detection of application-layer DDoS attacks using machine learning and genetic algorithms. Computers & Security, 135, (2023) 103511.
[29] S. Yevseiev, S. Pohasii, S. Milevskyi, O. Milov, Y. Melenti, I. Grod, D. Berestov, R. Fedorenko, O. Kurchenko, Development of a method for assessing the security of cyber-physical systems based on the Lotka–Volterra model. Eastern-European Journal of Enterprise Technologies, 5(9(113), (2021) 30–47. doi: 10.15587/1729-4061.2021.241638
[30] Lotka-Volterra equation solver. URL: https://fusion809.github.io/LotkaVolterra/