

The impact of the application of antivirus programs on the security of computer systems

Uticaj primene antivirusnih programa na bezbednost računarskih sistema

Ivan Jovanović^a, Milena Cvjetković^{b*}, Milovan Cvjetković^c

^a Aviation Academy, Belgrade, Serbia

^b College of Academic Studies "Dositej", Belgrade, Serbia

^c Technical College of Academic Studies, Belgrade, Serbia

Article info

Original scientific paper/ Originalan naučni rad

Received/ Rukopis je primljen:
23 January, 2024
Revised/ Korigovan:
17 April, 2024
Accepted/ Prihvaćen:
12 May, 2024

DOI:
<https://doi.org/10.5937/bizinfo2402011J>

UDC/ UDK:
004.056.54:004.78

Abstract

The paper presents a study that analyzes the application of different antivirus programs to reduce negative consequences on computer systems caused by malicious attacks. The research has shown the impact of antivirus programs on protecting computer systems from malicious attacks. The effect of antivirus programs was tested using neural networks. The following network activities were analyzed: payment for antivirus programs, sponsor advertising of antivirus programs, protection against viruses, system requirements, automatic updates, and technical support. The research results showed that the accuracy of predicting the intensity of consequences on computer systems was highest for small consequences. The network diagram showed that small and medium consequences were mainly caused, while large consequences were minimized. Analyzing the impact of each independent variable on reducing consequences on the computer system confirmed that the system configuration of the computer had the most significant impact. System requirements of the computer system affect the response rate of antivirus programs and the detection of different types of viruses.

Keywords: antivirus programs, malicious attacks, system requirements, computer system, neural network

Sažetak

U radu je prikazano istraživanje koje se bavi analizom primene različitih antivirusnih programa u cilju smanjenja negativnih posledica na računarske sisteme. Istraživanjem je prikazan uticaj primene antivirusnih programa u zaštiti računarskih sistema od zlonamernih napada. Pomoću neuronskih mreža ispitan je efekat primene antivirusnih programa. Analizirane su sledeće mrežne aktivnosti: plaćanje antivirusnog programa, reklame sponzora antivirusnog programa, zaštita od virusa, sisemski zahtevi, automatsko ažuriranje i tehnička podrška. Rezultati istraživanja su pokazali da je tačnost predviđana ishoda intenziteta posledica na računarski sistem najveća kod malih posledica. Mrežni dijagram je pokazao da su uglavnom prouzrokovane male i srednje posledice, dok su velike posledice svedene na minimum. Analizom uticaja svake nezavisne varijable na smanjenje posledica na računarski sistem, potvrđeno je da najveći uticaj ima sistemski konfiguracija računara. Sistemski zahevi računarskog sistema utiču na brzinu odziva antivirusnih programa i detektovanje različitih vrsta virusa.

Ključne reči: antivirusni programi, zlonamerni napadi, sistemski zahtevi, računarski sistem, neuronska mreža


1. Introduction

The development of information technology has led to more efficient protection of business information systems from cybercrime. The number of threats to information systems is constantly increasing, with numerous attacks recorded in different areas including education, healthcare, management, energy, finance, IT, telecommunications, and many others. According to the

Kaspersky Security Network, 1,416,295,227 network resources of organizations worldwide were blocked in 2020, while antivirus programs recognized 456,573,467 malicious attacks of unique URL addresses. Antivirus files discovered 87,941,334 unique malicious and potentially unwanted objects (Kaspersky Company, 2021). Computer systems in the current business world face different forms of malware or malicious attacks, which are very hostile in nature. The goal of such software

*Corresponding author

E-mail address: cvjetkovicm@gmail.com

This is an open access paper under the license 

is to disable the work of computer systems, and they can interfere with normal functioning with various hidden motives such as sabotage (obstruction), making money from computer system users, inability to realize business activities, and many others. The development of malware has made it difficult for them to be detected in computer systems as they have the ability to hide deep into the system. Antivirus programs play a key role in protecting computer systems from malware and achieving cyber security (The Economic Times, 2021).

Computer systems are threatened by various forms of malware. Adware is a type of malware that attacks a computer system in the form of on-screen advertisements. Spyware is another type of malware that secretly monitors a user's activity without their permission. A virus is a type of malware that connects to another program and inadvertently replicates itself through other computer programs. Trojan horses are very dangerous malware for a computer system, as they are used for financial information theft and as an instrument for installing other malware forms. Ransomware is a type of malware that locks the device and encrypts files, with the possibility of the user paying a ransom for access to their data. These attacks mostly endanger computer business systems, which are the focus of this study (Kaspersky Company, 2021).

The security of computer systems is one of the key areas of information technology. Preventing attacks and malicious activities in business information systems can prevent possible damages and losses and enable efficient work. Detecting attacks and frauds involves sample classification by identifying abnormal patterns from normalcy, using statistical classification, data mining, and machine learning methods (Raman et al., 2017; Song et al., 2017). Business information systems face numerous threats such as viruses, malware, service denial, and information theft. Package level analysis plays an important role in identifying malicious software threats and attacks that cannot be reported by antivirus software (Bagyalakshmi et al., 2018).

2. Literature review

According to Xiao et al. (2019), a classifier based on the Long Short-Term Memory (LSTM) language model is able to determine whether an attack on an information system is malicious with a high efficiency of 96.6% and a low false positive rate of 9.3%. Saxe and Berlin (2017) suggest the use of the eXpose neural network, which utilizes character-level embedding and convolutional neural networks, for detecting malicious attacks. Their research shows a high intrusion detection rate of 92% and a low false alert rate of 0.1%.

Althubiti et al. (2018) also employ the LSTM model for intrusion detection and report an accuracy of 85%. However, a subsequent study using the LSTM RNN model achieves even better results, with an accuracy of 97.54% and a precision of 98.95%. This top-to-bottom deep learning approach aims to address important cybersecurity challenges in an environment where arrays

are intentionally blurred to prevent clear function separation (Agarap, 2018).

Wang et al. (2017) are using the convolutional neural network for the classification of the malware traffic with the results of 99,41% of the classifier accuracy. According to the research the IPS and honeypot servers easily communicate with each other and monitor the networks while the content and the new patterns of the attacks are being added to the signature database. The evidence of the attacks is reducing the high level of the false attacks while the hybrid attack detection method with higher performances has been developed (Baykara & DAŞ, 2019).

For the detection of malicious attacks, passive detection technology using signature databases has been employed. An automated system has been developed to extract URLs from log files and send them for scanning by the malicious software system analysis. If a URL is found to be malicious, it is sent to the cache memory server to be blocked by the network (Bux et al., 2021). Research conducted by Gao et al. (2019) presents the STRIP model, which enables the detection of Trojan inputs and blockdoored models. This model is compatible with various alternatives of input-agnostic Trojan attacks and entropy manipulation attacks. In testing conducted by malware analysts (Raff et al., 2020), AutoYara was found to reduce the time spent building Yara rules by up to 44-86%, allowing analysts to focus on more advanced malware that current tools cannot handle.

The experimental results from Clements and Lao (2018) demonstrate that the proposed algorithm is capable of efficiently classifying the input trigger model as the specified class in the MNIST dataset by injecting hardware Trojans in approximately 0.03% of neurons hidden in the fifth layer of arbitrary 7-layer convolutional neural networks.

The TAN (Transaction Authentication Number) generator is a crucial model for protecting banking services and preventing cybercrime, as noted by Sardina et al. (2016). The banking sector is one of the most vulnerable areas of the economy. The application of artificial intelligence techniques could help solve various issues related to fraud and data leakage (Soni, 2019).

The research by Li et al. (2012) proposes an algorithm based on the deep migration learning model for the detection and extraction of intrusions in computer systems, which combines learning models with the time of intrusion. The results show that the algorithm has a greater detection efficiency for a shorter period of time. Another research by Švarc and Strnad (2021) shows a method that can be used when the volume of attacks on a computer system is high and the time period between actions is the same. The detection rate for simple attacks was 93.57% and for advanced attacks, it was 95.37%. Denial-of-Service (DDoS) attacks are one of the most effective attacks in terms of the damage inflicted and the challenges involved in preventing, detecting, and controlling them. Research has shown that artificial

intelligence can be used to prevent complicated DDoS attacks (Vuković, 2020).

The research results have shown that during the training of a neural system with a backdoor sample, 28% of the malicious Trojan type signatures were identified, and the worm sample was sensitive to this virus type during the neural network learning process. This research has confirmed the importance of using a malicious software identification system to preserve information security (Gavrylenko et al., 2018). Training neural network architectures with millions of files can offer efficient performance for classifying malicious attacks. One research study presented several binary and family architectures for such classification, and found that multi-task learning improves classification results for very low false positive rate values below 0.07%. The study suggests that a two-class, binary classification should use two hidden layers and multi-task learning, while a shallow multi-task network is more efficient for family classification (Huang & Stokes, 2016).

Another research study proposed a detection method for the MalNet malicious software using deep neural networks CNN and LSTM. MalNet was tested on a sample of 42,386 files and achieved an accuracy of 99.88% and TPR of 99.14% with an FPR of 0.1% (Yan et al., 2018). The study aimed to compare the effectiveness of different deep learning approaches, including RNN recurrent neural network, long-short-term memory LSTM, and other traditional machine learning classifiers. A study found that LSTM has the highest detection rate of malicious attacks compared to other deep learning approaches (Vinayakumar et al., 2018).

The goal of the research was to detect malicious attacks on a system by monitoring micro-architectural deviations. The proposed approach was found to be particularly efficient in detecting Rovhammer attacks that exploit DRAM interferences and Spectre attacks that exploit speculative execution and side channel vulnerabilities. Experimental results showed a detection accuracy of 0% total error for Rovhammer attacks using SVM and only 0.77% false positives and 0% false negatives for Spectre attacks using a multilayer Perceptron classifier (Li & Gaudiot, 2019). In another study, the performance of six classifiers was compared on a dataset of malicious software images. The ImageNet Large-Scale Visual Recognition Challenge model and three other CNN models were used for grayscale classification of the malicious software images.

The results showed that the Inception-V3 model outperformed the other models for classifying malicious software images (Bensaoud, 2020). The study explores the use of artificial neural networks and deep learning methods in cybersecurity analysis. The study suggests that a successful security model should incorporate deep learning techniques that are tailored to the specific data characteristics, and that a sophisticated learning algorithm requires training with relevant data and knowledge to enable intelligent decision-making (Sarker, 2021).

The research presented a classifier for identifying malicious attacks that outperforms modern models and achieves classification accuracies of 0.998, 0.911, and 0.997 using the Maling, Ember, and BIG2015 datasets, respectively (Ghouti & Imam, 2020).

3. The research methodology

The theoretical part of the research presented various methods and tools used for detecting and preventing malicious attacks. Previous research has not yielded significant results in analyzing the prevention of malicious attacks on computer systems using antivirus programs. This research highlights the importance of applying different antivirus programs to different computer system configurations and the possibility of protection against malicious attacks that can have various consequences.

The research goal was to determine if the MLP neural network could help the managers in the business organisations to predict the attack consequences on the computer systems they have been using in their businesses and to prevent those attacks. This research proves the following hypotheses:

- H1: The response speed of antivirus programs depends on the system requirements of the computer systems and reduces the effects of viruses on the computer system;
- H2: Effective protection of computer systems and reduction of the consequences of malicious attacks mostly depends on the system requirements of the computer.

The research methodology of this work is focused on the selection of computer systems with different configuration requirements and testing the response speed of antivirus programs when attacked by different types of viruses. The goal of the research is to determine which antivirus program provides the fastest and most effective protection of computer systems and the impossibility of locking or damaging files. In order to analyze the effectiveness of antivirus programs, a simulation of the attack of different types of viruses on computer systems with different configurations and characteristics was carried out. The behavior of computer systems and the response speed of antivirus programs was analyzed after opening files infected with different types of viruses. Based on this, the response speed of antivirus programs was measured using a shell script, which performs a network scan and searches for sensitive information. The shell script gave the time for which the antivirus is activated to attack the virus inserted into the computer system and block its malicious work. The speed of response is a key indicator of the effectiveness of the application of antivirus programs, as they enable the security of computer systems and data security. The negative impact of malicious attacks and the consequences they can have on the system and its data will depend on the application and speed of response of various antivirus programs and the very characteristics of computer systems.

Using a shell script, the speed of virus detection was tested using certain antiviruses. Each antivirus was tested with three different types of viruses. The recommended computer system configuration had the following specifications: Processor (CPU): Intel Core i5 (sixth generation or newer) or equivalent, Operating System: Microsoft Windows 10 Professional x64, RAM: 8 GB RAM, Hard Disk: 512 GB SSD or 1 TB HDD. The tested configuration had the following specifications: Operating System: Windows 10 Pro 64-bit, Processor (CPU): AMD Ryzen 3 2200G, RAM: 8 GB, Hard Disk: 3 TB Western Digital HDD, 1TB Seagate HDD. The system requirements for testing the antivirus programs were different. 360 Total Security antivirus was tested with the following system requirements: Operating system: Windows 10/8.1/8/7/Vista/XP (32-bit and 64-bit), Memory: 512 Mb, CPU: 1.6 GHz, Free disk space: 1 Gb.

AVG Antivirus was tested with the following system requirements: "Windows 10 except Mobile and IoT Edition (32 or 64-bit); Windows 8/8.1 except RT and Starter Edition (32 or 64-bit); Windows 7 SP2 or later, any Edition (32 or 64-bit), Windows fully compatible PC with Intel Pentium 4 / AMD Athlon 64 processor or above (must support SSE2 instructions); ARM-based devices are not supported. 1 GB RAM or above 2 GB free space on the hard disk. Internet connection to download, activate, and maintain application updates and antivirus database. Optimally standard screen resolution no less than 1024 x 768 pixels".

Panda Dome antivirus was tested on a computer system with the following characteristics: "Operating system: Windows 10 (32/64-bits); Windows 8/8.1 (32/64-bits); Windows 7 (32/64-bits); Windows Vista (32/64-bits); Windows XP 32-bits (SP3 or later); Processor: Pentium 300 MHz or faster. RAM: 256 MB. Hard disk: 240 MB free space. Browser: MS Internet Explorer 6.0 or later. NOTE: Panda VPN requires Windows 7 or later and .NET Framework 4.6 or later".

McAfee antivirus was tested on a computer system with the following system requirements: "Operating System

Windows 10 (Does not support Windows 10 in S mode); Windows 8.1, 8 and 7 SP1 fully patched (32 and 64-bit); Build 4.10.1 or higher: macOS 10.15 and above; Build 4.9.0.2: macOS 10.12 - 10.14; Smartphones & Tablets; Android 7 or later; Apple iOS 13 or later; Hardware 2 GB RAM for Windows 7 and above 500 MB free drive space; 1 GHz Processor".

Avira antivirus was tested on a computer system with the following characteristics: "Operating systems Microsoft Windows 7 Service Pack 1 or above with the newest updates, hotfixes, and service packs installed; RAM 2 GB of RAM or more; Disk Space Minimum 2 GB of free disk space (additional space for temporary and quarantine files needed); CPU type Intel Pentium 4 / AMD Athlon 64 Processor or faster (with support for the SSE2 instructions set) Browser: Internet Explorer 11 or newer".

Kaspersky antivirus was tested on a computer system with the following characteristics: "General requirements; 1500 MB of free disk space; Processor with SSE2 support; Internet connection (for installation and activation, participation in the Kaspersky Security Network, as well as database and program module updates); Microsoft Internet Explorer 8.0 or later; To access My Kaspersky, we recommend using Microsoft Internet Explorer 9.0 or later; Microsoft Windows Installer 4.5 or later; Microsoft .Net Framework 4 or later; Hypervisor protection is not supported on 32-bit operating systems. FAT32 file system is not supported. Requirements for operating systems. 1 GHz processor or faster. 1 GB of free RAM for 32-bit operating systems and 2 GB of free RAM for 64-bit operating systems".

The attack of three types of viruses - a standard virus, a crypto (Pedro) and a Trojan horse - was simulated on computer systems with the specified system requirements and antivirus programs. Different antivirus programs have shown different response times when it comes to these three types of viruses. Table 1 shows the time period of activation of antivirus programs during malicious attacks obtained using shell scripts.

Table 1. Response time of antivirus programs during malicious attacks on the computer system

Types of viruses	The types of antivirus programs					
	360 Total Security antivirus	AVG AntiVirus	Panda Dome antivirus	McAfee antivirus	Avira antivirus	Kaspersky antivirus
Standard viruses	1.926s	doesn't detected	0.075	11.236s	1.288	5.523s
Kripto (Pedro)	0.981s	12.982s	4.354s	17.168s	doesn't detected	2.931s
Trojan horse virus	doesn't detected	0.104s	1.591s	16.341s	doesn't detected	5.897s

Source: Author's calculation based on shell script

The analyzed viruses have different degrees of influence on the security of computer systems. In order to protect ourselves from the larger consequences of malicious attacks, it is important to quickly detect these viruses and prevent them from causing further damage to computer systems. By simulating virus attacks and testing antivirus programs on computer systems with different system requirements, it was concluded that some antivirus programs react faster than others, especially when it comes to crypto viruses that have significant

consequences for the security of computer systems. Panda Dome and 360 Total Antivirus were found to have the fastest response times for detecting the viruses tested, while Avira failed to detect the two types of viruses analyzed. Based on these results, we can confirm hypothesis 1, which is that the response time of antivirus programs depends on the system requirements of computer systems and reduces the effects of viruses on computer systems. This study used predictors/variables as inputs to construct an Artificial Neural Network (ANN)

model capable of predicting the level of damage a virus can cause to a computer system. ANN consists of three layers and uses supervised learning algorithms to classify input data into specific output categories.

The research used dependent and independent variables. Variables are defined according to the characteristics of computer systems and antivirus programs. Six independent variables were analyzed. The independent variables are the payment of the antivirus program and the opportunity to use the advertisements of the free virus sponsor. These variables indicate whether a free antivirus provides exactly the same protection as a paid virus. The values of these two variables are 0 and 1, where 0 is a negative response, while 1 is a positive response. Among the variables analyzed was the possibility of protection against all viruses, the values of which are expressed on a scale from 1 to 5, where 1 indicates very weak viruses, while 5 indicates extremely strong viruses. The intensity of the virus depends on the type of virus, where we can only talk about the effect of the virus on the slow operation of the computer, until the files on the computer system are locked or damaged.

The system requirements of the computer were analyzed through the possibility of applying different capacities of RAM memory that the computer system can have, so this variable was analyzed through the following values: 1 - 1GB, 2 - 2 GB, 3 - 4 GB, 4 - 8 GB and 5 - 16 GB of RAM. The variables automatic update of the antivirus program and technical support are expressed through the two values 0 and 1, where 0 indicates the impossibility of their application, while 1 indicates the affirmative answer of their implementation. The dependent variable indicates the consequences that viruses can have on the computer system and it is expressed through three values. A value of 1 indicates minor consequences that the computer suffers from a virus attack, and these consequences are usually caused by a standard virus that slows down the computer. The medium intensity of consequences represents a value of 2 and expresses the consequences of a virus such as a Trojan horse that disrupts the operation of the entire system, but there is a possibility to remove the consequences. The major consequences brought by the virus have a value of 3 and they are characteristic of a crypto virus (Pedro) that locks all files on the computer, the possibility of restoring files is minimal, while there is the possibility of paying a ransom for accessing data, but with a high possibility of fraud.

The research was conducted in a virtual environment, by simulating the attack of three different types of viruses on computer systems with different system requirements. The research sample collected data related to 150 computer systems used in secondary and higher education. The multi-layer preceptor MLP module of IBM Statistics 25 was used to analyze the collected data, which was the basis for building the neural network model and testing its accuracy.

For the neural network construction model, its training capabilities are defined, which include four parameters shown in graph 1. The purpose of the Lambda parameter is to control whether the Hessian matrix is negatively defined. Based on the Sigma parameter, the magnitudes of weight changes affecting the Hessian score are controlled via the derivative of the first-order error function (Zacharis, 2016). The value of the Lambda parameter is set to 0.0000005, while the value of the Sigma parameter is 0.00005.

The rules for stopping the neural network training were defined by the following parameters (Zacharis, 2016):

- "Maximum steps without a decrease in error 5;
- Maximum training time 15 min;
- Minimum relative change in training error 0.0001;
- Minimum relative change in training error ratio 0.001".

3.1. The research results

The Table 2 represents the descriptive statistics of the analyzed variables, where the total amount of the business computer systems covered by the research is shown. There were listed the minimal, maximal and medium values of the analyzed variables as well as the standard deviation.

In the Table 3 were shown the data set information that are used for the ANN model building.

Table 3. Case Processing Summary

		N	Percent
Sample	Training	108	72.0%
	Testing	42	28.0%
Valid		150	100.0%
Excluded		0	
Total		150	

Source: Author's calculation based on SPSS 25

Table 2. The descriptive statistics of the analyzed variables

Variables	N	Min	Max	Medium value	Standard deviation
Payment	150	0	1	.81	.391
The possibility of using advertisements	150	0	1	.45	.499
Protection against all types of viruses	150	1	5	3.85	.763
System requirements	150	1	5	2.37	1.138
Automatic update	150	0	1	.76	.429
Technical support	150	0	1	.70	.460
Consequences on the system	150	1	3	1.89	.906

Source: Author's calculation based on SPSS 25

In the Table 4 is shown the neurons number in every layer and six independent analyzed variables.

Table 4. Network information

Network information		
Input Layer	Covariates	1 Payment 2 The possibility of using advertisements 3 Protection against all types of viruses 4 System requirements 5 Automatic update 6 Technical support
	Number of Units ^a	6
	Rescaling Method for Covariates	Standardized
	Hidden Layer(s)	1
	Number of Units in Hidden Layer 1 ^a	3
	Activation Function	Hyperbolic tangent
Output Layer	Dependent Variables	1 Consequences on the system
	Number of Units	3
	Activation Function	Softmax
	Error Function	Cross-entropy

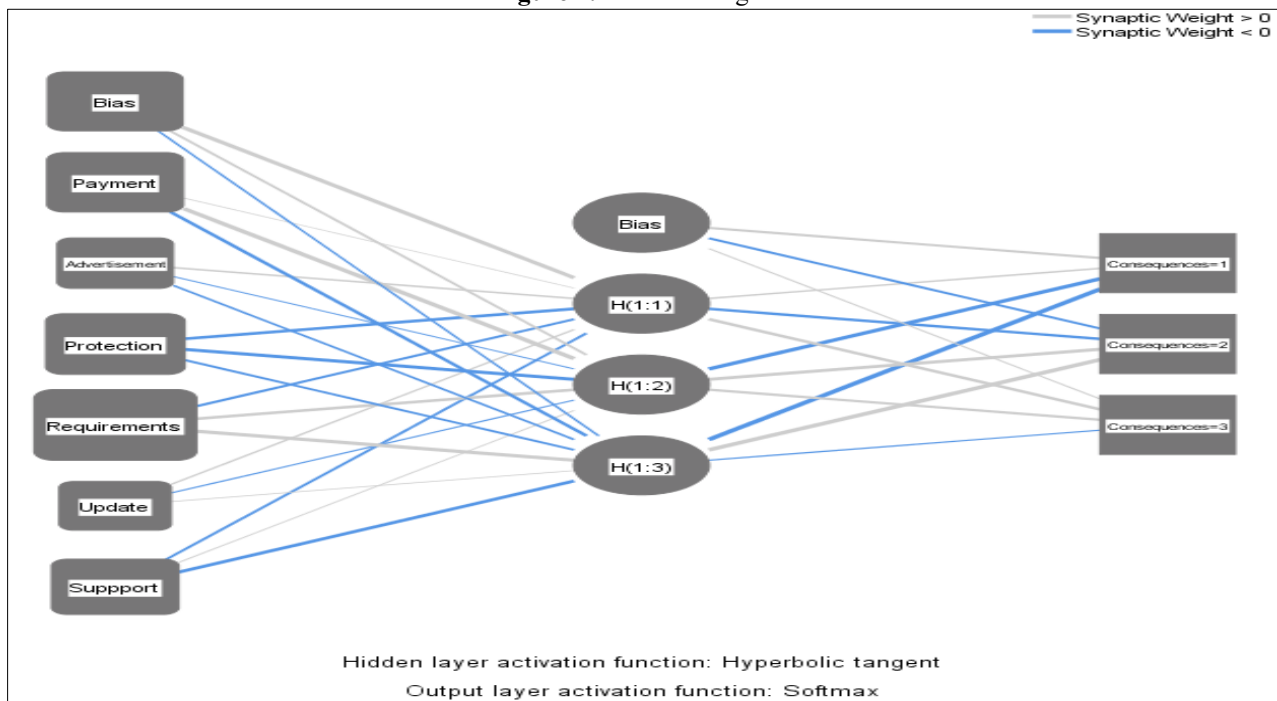
a. Excluding the bias unit

Source: Author's calculation based on SPSS 25

Based on the network information shown in the graph, it can be seen that the automatic architecture selection has selected 1 hidden layer and 3 nodes in the hidden layer, while the output layer includes 3 nodes.

The hidden layer of the activation function is the hyperbolic tangent. The SoftMax function was used for the output layer. Cross entropy was used as the error function.

Figure 2. Network diagram



Source: Author's calculation based on SPSS 25

According to the network diagram (figure 2), three different outcomes can be observed - small, medium and large consequences on the computer system, due to the use of 6 network activities: payment, advertising possibility, virus protection, system requirements, automatic adjustment and technical support. The network diagram shows that there are 6 input nodes, 3 hidden layers and 3 output nodes that indicate the intensity of the consequences on the computer system due to the activity of the virus. Table 5 shows the results related to the training and testing results and the retention sample.

Table 5. Model summary

Model summary		
Training	Cross Entropy Error	18.392
	Percent Incorrect Predictions	22.7%
	Stopping Rule Used	5 consecutive step(s) with no decrease in error ^a
	Training Time	0:00:00.06
	Cross Entropy Error	7.682
Testing	Percent Incorrect Predictions	21.4%
	Dependent Variable: Consequences on the system	

a. Error computations are based on the testing sample.

Source: Author's calculation based on SPSS 25

The cross-entropy error gives the results for the sample as well as for the test. The value of this error is 18,392, which indicates the power of the model to predict the outcome of the intensity of consequences on the computer system. When it comes to training data, the percentage of incorrect predictions is 22.7%, while the reported percentage of training data is 21.4%. The learning process is repeated until the sample reaches 5 consecutive steps without error reduction.

Table 6. Confusion matrix

Sample	Observed	Predicted			Percent Correct
		Small	Medium	Large	
Training	Small	41	0	10	80.4%
	Medium	4	14	3	66.7%
	Large	10	4	22	61.1%
	Overall Percent	50.9%	16.7%	32.4%	71.3%
Testing	Small	17	0	2	89.5%
	Medium	2	3	0	60.0%
	Large	5	0	13	72.2%
	Overall Percent	57.1%	7.1%	35.7%	78.6%

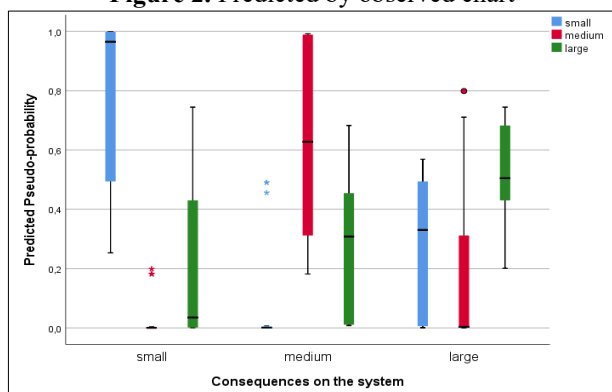
Dependent Variable: Consequences on the system

Source: Author's calculation based on SPSS 25

Table 6 shows the classification results for the categorical variable outcome and the intensity of the consequences on the computer system, broken down by individual outcomes and in total. The predicted outcome is defined as a success if the probability is higher than 0,5. Out of the total number of training cases, 71,3% of them were accurately classified, while this percentage was 78,6% for the training-related cases. When it comes to the outcomes of the intensity of the consequences on the computer system, the prediction accuracy is highest for small consequences.

Figure 2 shows a graph that displays the predicted pseudo-probabilities. This graph shows the outcomes of the intensity of the consequences that classify certain pseudo-probabilities based on a full set of points. The achieved values above 0.5 indicate accurate predictions.

Figure 2. Predicted by observed chart



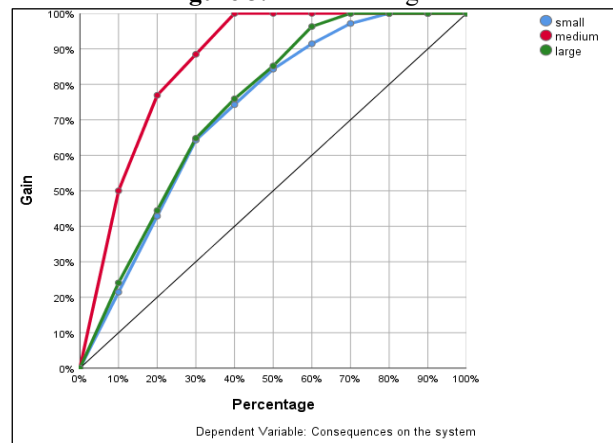
Source: Author's calculation based on SPSS 25

The following graph, Figure 3, presents the exact data classifications that were obtained by the ANN model according to the given classification, which could occur

by chance, that is, without using the neural network model. The efficiency measure of the classification model has been calculated as the percentage of accurate predictions obtained by the model, relative to the exact predictions obtained without the model, which is the baseline on the graph.

The farther the curve is from the baseline on the graph, the higher the gain. The higher gain indicates better performance. The diagram above is a visual aid in the performance validation of the classification model. Based on the values obtained in the gain diagram, the benefit factor of the stated model has been calculated. In this case, the 80% increase for the fault category is $80\%/40\% = 2$.

Figure 3. Cumulative gains



Source: Author's calculation based on SPSS 25

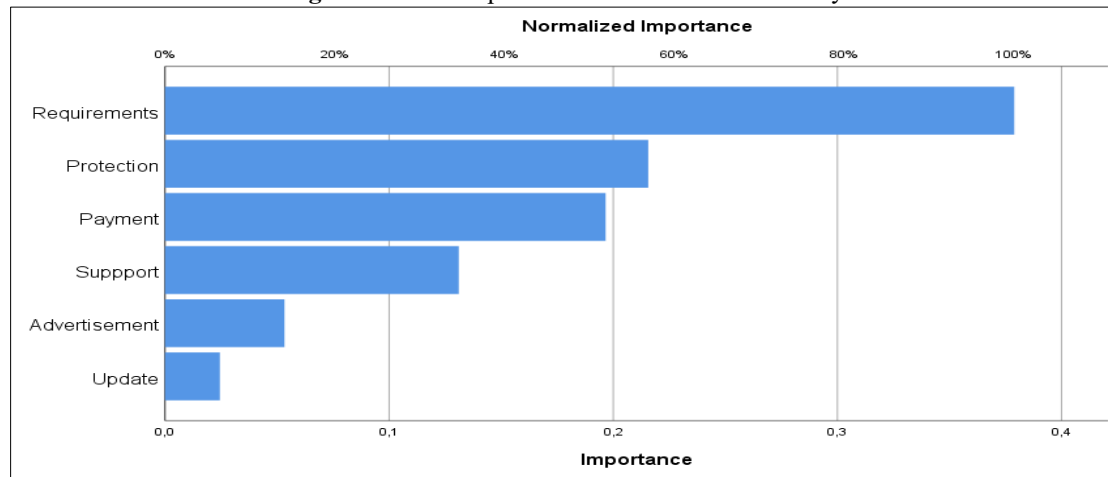
The influence of each independent variable on the intensity of consequences on computer systems has been examined. Table 7 and Figure 4 show the results of the influence of the independent variables on the dependent variable of the intensity of the consequence.

Table 7. Independent Variable Importance

	Importance	Normalized Importance
Payment	.197	51.9%
The possibility of using advertisements	.053	14.1%
Protection against all types of viruses	.216	56.9%
System requirements	.379	100.0%
Automatic update	.024	6.5%
Technical support	.131	34.6%

Source: Author's calculation based on SPSS 25

Based on the presented results of the influence of each independent variable on the outcome of consequences on computer systems, it has been determined that computer system requirements have the greatest influence. A slightly less significant effect has been shown by the variable of protection against all types of viruses. Therefore, efficient protection of computer information systems and reduction of malicious consequences can be achieved through appropriate computer system requirements and antivirus programs that provide protection against all types of viruses.

Figure 4. The independent variables action intensity

Source: Author's calculation based on SPSS 25

The variable that has shown the greatest impact on the protection of computer systems is system requirements. In other words, computer systems that have higher memory capacity and speed will respond more efficiently and quickly to malicious attacks. A somewhat weaker effect was shown by variable protection against all viruses. Effectiveness in the protection of computer systems will be ensured when the antivirus provides protection from viruses that slow down the computer, through those viruses that disrupt the operation of the entire computer system, to a virus that locks all files on the computer and makes it much more difficult to unlock it again. Also, the variable payment for an antivirus program had an impact, which only indicates that paid antivirus programs provide more effective protection than those that can be found for free on the Internet.

4. Conclusions

Choosing an adequate antivirus program provides protection against popular and currently known viruses that adversely affect the operating system and the entire information system. These antivirus programs immediately block the application of malicious programs in places where they appear most often, such as the startup directory and the system registry. In addition to blocking installation, all malicious files are automatically moved to quarantine even if the user did not run the file. This process occurs during a real-time system scan, and after the virus is quarantined, the user is given the option to delete the virus. Antivirus programs are tested on the average computer configuration used in information systems. Research results were obtained using shell scripts and neural networks. The research confirmed that the system requirements of the computer system have the greatest impact on preventing malicious attacks and their consequences. In addition, the response time of antivirus programs to malicious attacks is conditioned by the characteristics of the computer system. Based on these results, the research hypotheses were confirmed. Future research directions could be focused on security analysis using peer-to-peer technology in some closed information systems.

References

- Agarap, A. F. M. (2018, February). A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data. In *Proceedings of the 2018 10th international conference on machine learning and computing*, 26-30. <https://doi.org/10.1145/3195106.3195117>
- Althubiti, S. A., Jones, E. M., & Roy, K. (2018, November). LSTM for anomaly-based network intrusion detection. In *2018 28th International telecommunication networks and applications conference IEEE. (ITNAC)*, 1-3.
- Bagyalakshmi, G., Rajkumar, G., Arunkumar, N., Easwaran, M., Narasimhan, K., Elamaran, V., ... & Ramirez-Gonzalez, G. (2018). Network vulnerability analysis on brain signal/image databases using Nmap and Wireshark tools. *Ieee Access*, 6, 57144-57151. <http://doi.org/10.1109/ACCESS.2018.2872775>
- Baykara, M., & DAŞ, R. (2019). SoftSwitch: A centralized honeypot-based security approach using software-defined switching for secure management of VLAN networks. *Turkish Journal of Electrical Engineering and Computer Sciences*, 27(5), 3309-3325. <http://doi.org/10.3906/elk-1812-86>
- Bensaoud, A., Abudawaood, N., & Kalita, J. (2020). Classifying malware images with convolutional neural network models. *International Journal of Network Security*, 22(6), 1022-1031. [http://doi.org/10.6633/IJNS.202011_22\(6\).17](http://doi.org/10.6633/IJNS.202011_22(6).17)
- Bux, K., Yousaf, M., Jalbani, A. H., & Batool, K. (2021). Detection of malicious servers for preventing client-side attacks. *Mehran University Research Journal of Engineering & Technology*, 40(1), 230-240. <http://doi.org/10.22581/muet1982.2101.20>
- Clements, J., & Lao, Y. (2018). Hardware trojan attacks on neural networks. *arXiv preprint arXiv:1806.05768*.
- Ghouthi, L., & Imam, M. (2020). Malware classification using compact image features and multiclass support vector machines. *IET Information Security*, 14(4), 419-429. <https://doi.org/10.1049/iet-ifs.2019.0189>
- Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D. C., & Nepal, S. (2019, December). Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, 113-125. <http://doi.org/10.22581/muet1982.2101.20>

- Gavrylenko, S., Babenko, O., & Ignatova, E. (2018, April). Development of the disable software reporting system on the basis of the neural network. *Journal of Physics: Conference Series*, 998(1), 012009. <https://doi.org/10.1088/1742-6596/998/1/012009>
- Huang, W., & Stokes, J. W. (2016, July). MtNet: a multi-task neural network for dynamic malware classification. In *International conference on detection of intrusions and malware, and vulnerability assessment*. Springer, Cham. 399-418.
- Kaspersky Company (2021, May 25). *Securelist by Kaspersky*. <https://securelist.com/it-threat-evolution-q3-2020-non-mobile-statistics/99404/>
- Kaspersky Company (2021, May 30). *Malware & Computer Virus Facts & FAQs*. <https://www.kaspersky.com/resource-center/threats/computer-viruses-and-malware-facts-and-faqs>
- Li, D., Deng, L., Lee, M., & Wang, H. (2019). IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning. *International journal of information management*, 49, 533-545. <https://doi.org/10.1016/j.ijinfomgt.2019.04.006>
- Li, C., & Gaudiot, J. L. (2019, July). Detecting malicious attacks exploiting hardware vulnerabilities using performance counters. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)* 1, 588-597. IEEE. <https://doi.org/10.1109/COMPSAC.2019.00090>
- Raff, E., Zak, R., Lopez Munoz, G., Fleming, W., Anderson, H. S., Filar, B., ... & Holt, J. (2020, November). Automatic YARA rule generation using biclustering. In *Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security*, 71-82. <https://doi.org/10.1145/3411508.3421372>
- Raman, M. G., Somu, N., Kirthivasan, K., & Sriram, V. S. (2017). A hypergraph and arithmetic residue-based probabilistic neural network for classification in intrusion detection systems. *Neural Networks*, 92, 89-97. <https://doi.org/10.1016/j.neunet.2017.01.012>
- Sardina, J., Olkhovskii, A., & Lowell, R. P. (2018). U.S. Patent No. 10,025,710, Washington, DC: U.S. Patent and Trademark Office.
- Sarker, I. H. (2021). Deep cybersecurity: a comprehensive overview from neural network and deep learning perspective. *SN Computer Science*, 2(3), 1-16. <https://doi.org/10.1007/s42979-021-00535-6>
- Saxe, J., & Berlin, K. (2017). eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys. *arXiv preprint arXiv:1702.08568*.
- Song, Q., Zheng, Y. J., Xue, Y., Sheng, W. G., & Zhao, M. R. (2017). An evolutionary deep neural network for predicting morbidity of gastrointestinal infections by food contamination. *Neurocomputing*, 226, 16-22. <https://doi.org/10.1016/j.neucom.2016.11.018>
- Soni, V. D. (2019). Role of Artificial Intelligence in Combating Cyber Threats in Banking. *International Engineering Journal for Research & Development*, 4(1), 7-7.
- Švarc, L., & Strnad, P. (2021). Automated Computer Attacks Detection in University Environment. *Acta Informatica Pragensia*, 10(1), 75-84. <http://doi.org/10.18267/j.aip.147>
- The Economic Times (2021, June 3). *What is 'Computer Virus'*. <https://economictimes.indiatimes.com/definition/computer%20virus>
- Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2018). Detecting malicious domain names using deep learning approaches at scale. *Journal of Intelligent & Fuzzy Systems*, 34(3), 1355-1367.
- Vuković, I (2020). Application of artificial intelligence in detection of DDoS attacks, Archibald Reiss Days, 10.
- Wang, W., Zhu, M., Zeng, X., Ye, X., & Sheng, Y. (2017, January). Malware traffic classification using convolutional neural network for representation learning. In *2017 International conference on information networking (ICOIN)*, 712-717. IEEE. <http://doi.org/10.1109/ICOIN.2017.7899588>
- Xiao, X., Zhang, S., Mercaldo, F., Hu, G., & Sangaiah, A. K. (2019). Android malware detection based on system call sequences and LSTM. *Multimedia Tools and Applications*, 78(4), 3979-3999. <https://doi.org/10.1007/s11042-017-5104-0>
- Yan, J., Qi, Y., & Rao, Q. (2018). Detecting malware with an ensemble method based on deep neural network. *Security and Communication Networks*, 7247095. <https://doi.org/10.1155/2018/7247095>
- Zacharis, N. Z. (2016). Predicting student academic performance in blended learning using Artificial Neural Networks. *International Journal of Artificial Intelligence and Applications*, 7(5), 17-29.

