Issues in Information Systems
Volume 25, Issue 4, pp. 455-465, 2024

DOI: https://doi.org/10.48009/4_iis_2024_135

Analyzing machine learning algorithms for antivirus applications: a study on decision trees, support vector machines, and neural networks

Reese Martin, *Robert Morris University, rgmst170@mail.rmu.edu*
Robert Pava, *Robert Morris University, rjpst280@mail.rmu.edu*
Sushma Mishra, *Robert Morris University, mishra@rmu.edu*

Abstract

Machine Learning (ML) methods and tools are reshaping the cybersecurity landscape, enhancing the overall preparedness of organizations to ensure the confidentiality, integrity, and availability of processes and data. In the realm of cybersecurity, independent response, and detection, AI tools are being used extensively. This study delves into the effectiveness of three prominent machine learning algorithms – decision trees, support vector machines, and neural networks – in enhancing antivirus decision and response capabilities. Our study, which includes an extensive literature review on using ML techniques in the cybersecurity incident response and detection domain, has yielded significant findings. We have explored their effectiveness in locating and efficiently blocking incoming malware, and we discuss the implications of these findings and suggest future research directions.

Keywords: Machine Learning, Neural Network, Decision Tree, Support Vector Machine, content analysis, AI

Introduction

Artificial Intelligence (AI) has fundamentally changed how security threats are detected, analyzed, and mitigated. As cyber threats evolve and become increasingly persistent and dangerous, traditional cybersecurity solutions, such as antivirus, struggle to keep up with the protection they traditionally provided. With its advanced data processing and pattern recognition capabilities, AI offers a potent solution to these challenges. Antiviruses have been traditionally used as an essential security control on personal devices to protect an individual's sensitive information (Akhtar & Feng, 2023, p. 946). In the early days of the internet, this layer of security used the signature-matching method to determine malicious executables. The main problem with signature-based antivirus tools is that the individual would have to update the database so that it could understand the most recent malware. The push for a solution came to machine learning as it provided a reasonable response time to prevent viruses from gaining access to these devices. Machine learning has allowed a way to combat the exponential release of viruses and malware on the internet (Rhode et al., 2023). In this era of artificial intelligence and machine language-based tools, there is a demand to fine-tune the detection of such antiviruses because of the inconsistency of false positive detection and accuracy of malware.

This paper explores how AI is reshaping cybersecurity practices, specifically focusing on Machine Learning (ML) techniques such as decision trees, neural networks, and support vector machines (SVMs) in the intrusion detection domain. This paper aims to conduct a thorough investigation into machine learning and its prominent algorithms commonly employed in antivirus software detection solutions. The analysis will focus on three widely used algorithms: decision trees, support vector machines, and neural networks.

The impact of these algorithms will be analyzed through the accuracy of detection and the overall impact of the software it has with each algorithm. The study emphasizes the importance of examining personal devices to enhance security measures for individuals and organizations. Identifying these requirements is crucial to help better protect systems that hold sensitive data that individuals aim to safeguard. The choice of the best machine learning algorithm for antivirus applications depends on various factors, including the nature of the data, the specific requirements of the task, and the computational resources available. Each mentioned algorithm—decision trees, neural networks, and support vector machines (SVMs)—has strengths and weaknesses.

AI impacts cybersecurity practices in various ways. It has allowed automated threat detection, behavior analysis of network traffic, predictive analytics to define the baseline configuration and more adaptive defense mechanisms. This can allow for the creation of unparalleled capabilities while destroying cybersecurity defenses. (Camacho, 2024) Various AI techniques are available for cybersecurity; however, machine learning is beneficial for intrusion detection. With machine learning, antivirus software can significantly enhance its ability to detect and learn from the current malware and viruses that attempt to enter the protected system. This technology is not limited to known malware and viruses; it can also help detect potentially new malware and viruses that are unknown at that point. (SakthiMurugan, 2024) Identifying and preventing zero-day attacks, an area that often instills fear due to their unknown impact is a significant advantage of machine learning in antivirus software. This reassurance of the technology's effectiveness can help prevent system damage and instill confidence in its capabilities. Along with protecting individual systems, the protection of whole networks can also benefit from machine learning antivirus software. Even though this study is aimed to focus on individual systems protection, an organization can use the shield for layered security. (Zhang & Wang, 2024) For example, if a network has a firewall and antivirus software on the personal devices, the layered security can protect the system if viruses or malware gets past the firewall.

Continuing to develop and evolve with hackers is the only possible way to protect systems. Developing new ways to detect and protect systems would allow hackers to continue to collect and profit from individuals' personal information. Working with antivirus software, the ability and need for more secure software that can better detect and analyze malware. Our analysis of machine learning algorithms has concluded that decision trees, support vector machines, and neural networks can be used to help support antivirus software. Analyzing academic papers, we discussed the importance and use of each machine learning algorithm. We will also touch upon the drawbacks and future research that needs to be conducted.

The rest of this paper is organized as follows. The paper's methodology is presented after the introduction section, followed by a critical analysis of the extant research literature in decision trees, support vector machines, and neural networks for antiviruses. The discussion section presents what was learned and the implications of this critical content analysis study. The last section of the paper summarizes the study and discusses potential future research opportunities.

Methodology

The study employed a content analysis methodology, focusing on published papers in relevant research domains. A thorough literature review explored the various algorithms suitable for machine learning: decision trees, support vector machines, and neural networks. Search terms such as "anti-virus," "decision trees," "support vector machines," and "neural networks" were utilized to retrieve relevant studies from diverse research databases and websites, including link.springer.com, proquest.com, and IEEE, among other related domains. The main area that we analyzed was through our university's database subscriptions since we had paid access to researched peer review papers. We selected papers that covered different models using each of the proposed algorithms. The paper selection process involved analyzing the abstract first. Then if the abstract contained information that was relevant to our topic, we then reviewed the study to assess its importance. Fifty articles were reviewed for this study (some did not contribute to our understanding of the research context and are not referenced here). A detailed discussion of each algorithm reviewed in this study is presented below.

Results

Under the umbrella of ML techniques, we analyzed decision trees and their ability to detect malware by analyzing the first area of machine learning within antivirus software. We looked over multiple papers that discussed the importance and limitations of decision trees within antivirus software. The current process that decision trees are used within antivirus software is disassembling files, putting them into statistical features, and using a decision tree to classify if they include malicious activity. Working with decision trees has allowed malware detection to be more recognized in today's environment. Unfortunately, one of the significant drawbacks of using decision trees is the amount of time and resources required to disassemble the files.

Another aspect needed when analyzing detection methods is whether the analysis is static or dynamic (Bashari et al., 2011). Static analysis is when the program is scanned without executing, while dynamic analysis is when the program is being scanned as it is executing. Both methods are beneficial and have drawbacks when applicable (Palša et al., 2022). A paper that did a comprehensive analysis of machine learning algorithms concluded that decision trees have the highest accuracy of the chosen models at the rate of 91.2%. Along with accuracy, the other categories, such as precision, recall, and the F1-Score, all have above 90% in the metrics. Another benefit of decision trees is the robustness of noisy data, which allows for better analysis and detection of malware threats (Cobian, n.d.). The amount of data available today on the internet is exponentially increasing. This has allowed individuals to overflow networks with noise. Decision trees will limit the noise of the information. Through other papers discussing the decision trees, there were various accuracy percentages, with one being 81.58% (Rhode et al., 2021). These multiple numbers and calculations of accuracy show that there is a percentage of error to consider. The consensus is a positive impact of using decision trees as a benefit over Perimeter-based detection (Hossain et al., 2022).

As mentioned above, there are limitations to decision trees when detecting malware in antivirus software. Decision trees are a part of supervised learning algorithms. Without proper training data, the solution could result in overly complex and false positive results with detection. The time necessary to train the algorithm to detect and analyze malware correctly becomes much more challenging and time-consuming (Smith et al., 2022). Along with ensuring the correct data is implemented, the false negative rate, as it varies, remains too high to implement confidently within all devices. As one paper mentioned, the false negative rate is 17.3%. (Alqahtani, 2021). This shows the potential of using machine learning techniques such as decision trees to better protect individuals against malware. The other limiting factor is the time used in training and testing sets. This issue allows the organization much more time to implement these detection systems. However, as of right now, the technology is not up to par where the requirement is needed (Carlin et al., 2017). Data processing is too involved with the resources, and current technology's false negative rates are too high.

Support vector machines

The second machine learning algorithm that we researched support vector machines. We found ten papers discussing support vectors and their advantages and disadvantages. One such advantage of supporting vector machines is their ability to detect frequencies. It is possible to use SVMs to detect malicious radio signals by detecting whether the call is from the primary user or a malicious primary user (Cadena Muñoz et al., 2020). By performing primary user emulation (PUE), the attacker could cause a denial of service to direct users by making the network release the used channel. The SVM only had a 5% margin of error in detecting these fake users (Cadena Muñoz et al., 2020). Another prominent feature of SVMs is their reliance on previous knowledge. SVMs require a “training process” to develop what they seek (Faris et al., 2017). This would mean that there would need to be a database of known malicious information for the SVM to detect malware. It is also proposed that a multi-verse optimizer (MVO) be used to generate the parameters for an SVM so that it can be trained on what to detect (Faris et al., 2017). It has even been seen that Karush–Kuhn–Tucker (KKT) conditions can help identify malicious actors (Lu & Roychowdhury, 2007). An additional feature is predictability. In a sales forecast, SVMs were used to predict future trends for different stores (Di Pillo et al., 2016). This can be applied to applications as it can observe the behavior of an application, acknowledge abnormal behavior, and detect intrusion (Kim & Park, 2003).

There were also some flaws with this method that we came across in our research. For example, it has been found that it is possible to trick SVMs so that they miss malware (Biggio et al., 2014). If the SVM uses signature-based detection, it is possible to change the attack slightly so that the signature is different. The SVM would then not detect it because the signature does not match the one it was trained on. It is also true that attackers can implement machine learning into their attacks (Biggio et al., 2014). They could use machine learning to develop tactics to attack someone or something based on knowledge of their systems before the attack. Also, SVMs are at a disadvantage as they require data, and they base their detection on formatting (Ding et al., 2015). If the data is inaccurate to the attack, the SVM will likely not detect it. This can be seen in static analysis, where factors like computer crashes affected the percentage of false positives/negatives (Kakavand et al., 2018). Since the data was abruptly stopped, the detection was inaccurate.

Neural networks

The third type of machine learning used in antivirus detection we analyzed is neural networks. We examined ten papers to understand better, process, and determine the benefits and drawbacks of this type of machine learning method. This type of machine learning is mainly used to detect malware within memory addresses and instruction execution. This is because of the ease of updating the parameters, unlike other methods. An example of neural networks being a benefit is a paper that conducted a case study using 3,000 ransomware samples (Rhode et al., 2021). Their studies found that the model had a high malware detection rate of around 94%. However, the accuracy of this method of data drops to 67.7% when the model is trained with static data, which helps present the need for new ways of malware detection. Another benefit of this detection method is identifying specific types of malware, such as registry malware (Rhode et al., 2018). Even though changing the registry can be more challenging, using neural networks to detect and prevent damage is essential. A test conducted with the Windows 10 registry with malware detection neural networks only got around 34% accuracy, while other methods, such as the boosted tree, reached as high as 72% (Ali et al., 2019). Accurate neural networks can present a sound detection system for audio-visual domains. Through the learning model of neural networks, information can be analyzed as it is coming into the system, which allows for the protection of these domains that cannot be stopped from being investigated. Neural networks can also be used to analyze data sets, which would typically take more time because of the size of the data dumps (Pundir et al., 2020). Taking a version of malware detection named MRm-DLDet, it can analyze memory dumps and have an accuracy detection rate of 98.34%. The push for more accurate and faster data processing can be challenging because of the need and current methods of machine learning within malware detection (Liu et al., 2023).

One major drawback of neural networks is the requirement of the data set to train the data application. The required amount of data to train the neural networks remains high to gain the accuracy needed within the field. Another limitation neural networks pose is classification accuracy (Moussas & Andreatos, 2021). With neural networks having a classification accuracy that is too low, using this form of ML on its own can pose an issue with building the malware dataset from which to learn. With the ability of all these machine learning methods to protect networks and devices, the opposite can occur to attack systems (Catak et al., 2021). Neural networks are one of the machine learning algorithms that attackers use to bypass malware detection. A 178MB-AlexNet model can embed 36.9MB of malware with a 1% accuracy loss to a system. This poses a threat to keeping up the detection battle with malware when they use the same tools to become undetected (Wang et al., 2021). With all the information collected through analyzation the following table presents a summary of the findings for all three approaches.

Table 1: Summary of findings

Machine Learning Algorithms	Benefits	Drawbacks
Decision Trees	• Accuracy: 91.2% (Cobian, n.d.) • Analytics of noisy data (Rhode et al., 2021) • High-performance computing (Rhode et al., 2021) • Captures complex relationships in data and is less prone to overfitting (Zhang et al., 2020). • Handles both numerical and categorical data (Saad, 2018).	• Struggles with capturing intricate patterns in large datasets or handling highly dimensional data (C & Babu, 2016). • Time Needed to break down files (Bashari et al., 2011)
Support vector machines (SVM)	• Accuracy: 80% (Kakavand et al., 2018) • Detect information based on the training process (Faris et al., 2017), • Predict trends (Di Pillo et al., 2016) • Frequency detection (Cadena Muñoz et al., 2020), • Good with linear and non-linear data (Bhavsar & Panchal, 2012). • Observing the behavior of an application (Kim & Park, 2003)	• Effectiveness can be influenced by the choice of the kernel function (Lu et al., 2014). • It relies on a previous set of knowledge, possible to trick SVMs into missing malware, and data needs to be formatted (Biggio et al., 2014)
Neural Network	• Accuracy: 94% (Rhode et al., 2017) • Memory addresses, Instructions, and API calls (Rhode et al., 2021) • Detection of malware within voice/video transmissions (Pundir et al., 2020) • Automatically extracts relevant features from the data, making them suitable for tasks with intricate patterns (Notley & Magdon-Ismail, 2018). • Great in learning complex hierarchical features (Xue, 2018).	• Malware using NN to bypass security (Wang et al., 2021) • Training deep neural networks can be computationally intensive (Novikov et al., 2015). • Require large amounts of labeled data for training, and their complexity makes the interpretation hard (Novikov et al., 2015).

Discussion

Based on the papers we have researched thus far, it appears that neural networks would be the ideal machine-learning algorithm. Decision trees seem to have an instability factor that you must constantly account for (Mirzamomen & Kangavari, 2016). Machine learning takes the information that is available to it and compares it to the traffic that is coming into the network. Decision tree models are known to be unreliable since if training data is altered in any way, it can result in false readings or positives. This could cause downturns in work as analysts would need to remediate the training data and ensure that it fits the company's standards. Additionally, the decision tree algorithms provide for linear decisions where this needs to happen if something happens. Malware is constantly changing and evolving and may work differently than what was implemented. This would mean that any machine learning based on this linear algorithm could miss some threats if not implemented properly. Along with the ideal machine learning algorithm, there is a drawback that would require the analysis of the situation. We concluded from analyzing a limited number of papers with specific actions involved with the antivirus software. This is with a case-by-case discussion of the ideal machine learning algorithm that works with antivirus software.

Support vector machines have been seen to work better for predicting events. The algorithm sets parameters that look at many possible evaluations (Faris et al., 2017). Unfortunately, this method takes a long time to parse all the information. This means machine learning could take even longer to read through the data once it is compiled. Any downtime with this method could allow malware to spread to different computers on the network. All the observed tests had to implement additional parameters or other techniques for this algorithm to function correctly. Additionally, SVMs need the data to be formatted for reading. This would cause more downtime when it comes to processing data. Support vector machines would be inefficient compared to neural networks without the additional parameters or other techniques.

Neural networks allow for the detection of changing factors. This means that this method should be the best algorithm for detecting malware. They can see something within the set parameters or what is considered an “outlier” to the parameters (Faris et al., 2010). This can be used to find any anomalies in the environment that malware might have caused or any new file that entered the environment. The outlier can then be added to the parameters to look for it. With some external additions, neural networks can increase their accuracy rate. Due to our limited resources, we could not test these algorithms ourselves. All the data in this paper has been gathered externally through the databases provided in our methodology section. The following table describes the status of each machine learning algorithm being used within antivirus software.

Table 2: Status of ML within antivirus software

Machine Learning Algorithms	Current Status	Context of Antivirus
Decision Trees	It is commonly utilized but can become unsuitable for application when training data has been compromised or altered.	It relies on data to make determinations, but it may fail to detect viruses if the data is damaged or compromised.
Support vector machines (SVM)	The usage depends on the individual application, but it is not suggested to be used when training data is corrupted.	Uses data to make determinations and predict future issues with data. However, it could fail to detect viruses if the data provided is not formatted.
Neural Network	Usage is expanding. However, it is not advisable to use it when it is under attack from another neural network.	It can detect new malware variants and offers greater flexibility, as it can be trained like the other algorithms.

Implications

These results suggest that neural networks outperform the other described machine learning algorithms when detecting malware. Decision trees can detect malware at a slower pace and, just like support vector machines, require data to be trained on. However, there is a possibility that more than one algorithm will be used for malware detection. It has been found that "unclassifiable regions can be resolved by decision tree-based SVM for multi-classification" (Mulay et al., 2010, p.41). This would change the percentage that the algorithms could detect malware. Another possibility is the use of multiple decision trees or SVMs for detection. It has been tested that using two SVMs and a decision tree SVM makes it possible to detect and solve minor binary classification problems with the SVM (Liang et al., 2017). Concepts like this were not covered in this paper and can be researched further in the future.

On a practical note, Experimentation and empirical testing often determine the best approach for a particular environment (Sandnes & Eika, 2017). It is essential to assess the performance of different algorithms on an organization's specific dataset and choose the one that yields the best results regarding accuracy, precision, recall, and other relevant metrics. The choice might also depend on other considerations, such as interpretability, resource constraints, and real-time processing requirements. On a practical note, neural networks require massive computational resources and high-level expertise to train the models effectively, including the need for large, labeled datasets. This can be associated with the rising cost of implementing these techniques. It is suitable for use in a large organization. On the other hand, SVM and decision trees are more accessible to implement and interpret and often work well with smaller datasets and less computational power, making them useful for small and medium-sized organizations.

Conclusion

This paper comprehensively reviews machine learning's influence on antivirus detection and response. We analyzed three algorithms (decision trees, support vector machines, and neural networks) and how they effectively impact antivirus's ability to locate and adequately block incoming malware. Decision trees demonstrate effective malware detection with appropriate data yet face challenges in data processing and training model efficacy. Support vector machines utilize frequency detection, data, and predictions for malware detection, albeit with limitations if the data is inaccurate. Neural networks emerge as the most effective malware detection, boasting the ability to swiftly update search parameters, identify registry malware, and process data efficiently. Further research is warranted as this domain of machine learning continues to become more prevalent.

References

Akhtar, M. S., & Feng, T. (2023). Evaluation of Machine Learning Algorithms for Malware Detection. Sensors, 23(2), 946. https://doi.org/10.3390/s23020946

Ali, M., Shiaeles, S., Clarke, N., & Kontogeorgis, D. (2019). A proactive malicious software identification approach for digital forensic examiners. Journal of Information Security and Applications, 47, 139–155. https://doi.org/10.1016/j.jisa.2019.04.013

Alqahtani, M. A. (2021). Machine Learning Techniques for Malware Detection with Challenges and Future Directions. International Journal of Communication Networks and Information Security, 13(2), 258-270. https://reddog.rmu.edu/login?url=https://www.proquest.com/scholarly-journals/machine-learning-techniques-malware-detection/docview/2582833572/se-2

Bashari Rad, B., Masrom, M., Ibrahim, S., & Ibrahim, S. (2011). Morphed virus family classification is based on the statistical features of opcodes using a decision tree. Informatics Engineering and Information Science, 123–131. https://doi.org/10.1007/978-3-642-25327-0_11

Bhavsar, H., & Panchal, M. H. (2012, December). A Review on Support Vector Machine for Data Classification. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d683a971524a0d76382ce335321b4b8189bc8299

Biggio, B., Corona, I., Nelson, B., Rubinstein, B. I., Maiorca, D., Fumera, G., Giacinto, G., & Roli, F. (2014). Security evaluation of support vector machines in adversarial environments. Support Vector Machines Applications, 105–153. https://doi.org/10.1007/978-3-319-02300-7_4

C, K. K., & Babu, V. (2016). A survey on issues of decision tree and non-decision tree algorithms. International Journal of Artificial Intelligence and Applications for Smart Devices, 4(1), 9–32. https://www.researchgate.net/publication/306047758_A_Survey_on_Issues_of_Decision_Tree_and_Non-Decision_Tree_Algorithms

Cadena Muñoz, E., Pedraza Martínez, L. F., & Ortiz Triviño, J. E. (2020). Detection of malicious primary user emulation based on a support vector machine for a mobile cognitive radio network using software-defined radio. Electronics, 9(8), 1282. https://doi.org/10.3390/electronics9081282

Camacho , N. G. . (2024). The Role of AI in Cybersecurity: Addressing Threats in the Digital Age. Journal of Artificial Intelligence General Science (JAIGS) ISSN:3006-4023, 3(1), 143–154. https://doi.org/10.60087/jaigs.v3i1.75

Carlin, D., Cowan, A., O'Kane, P., & Sezer, S. (2017). The effects of traditional anti-virus labels on malware detection using dynamic runtime opcodes. IEEE Access, 5, 17742–17752. https://doi.org/10.1109/access.2017.2749538

Catak, F. O., Ahmed, J., Sahinbas, K., & Khand, Z. H. (2021). Data augmentation-based malware detection using Convolutional Neural Networks. PeerJ Computer Science, 7. https://doi.org/10.7717/peerj-cs.346

Cobian, D. (n.d.). Comprehensive Efficiency Analysis of Machine Learning Algorithms for Developing Hardware-Based Cybersecurity Countermeasures. https://arxiv.org/pdf/2201.07654.pdf

Di Pillo, G., Latorre, V., Lucidi, S., & Procacci, E. (2016). An application of support vector machines to sales forecasting under promotions. 4OR, 14(3), 309–325. https://doi.org/10.1007/s10288-016-0316-0

Ding, S., Zhu, Z., & Zhang, X. (2015). An overview of semi-supervised support Vector Machine. Neural Computing and Applications, 28(5), 969–978. https://doi.org/10.1007/s00521-015-2113-7

Faris, H., Hassonah, M. A., Al-Zoubi, A. M., Mirjalili, S., & Aljarah, I. (2017). A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture. Neural Computing and Applications, 30(8), 2355–2369. https://doi.org/10.1007/s00521-016-2818-2

Golovko, V., Bezobrazov, S., Kachurka, P., & Vaitsekhovich, L. (2010). Neural Network and artificial immune systems for malware and network intrusion detection. Advances in Machine Learning II, 485–513. https://doi.org/10.1007/978-3-642-05179-1_23

Hossain, Md. S., Hasan, N., Samad, M. A., Shakhawat, H. Md., Karmoker, J., Ahmed, F., Fuad, K. F., & Choi, K. (2022). Android ransomware detection from traffic analysis using metaheuristic feature selection. IEEE Access, 10, 128754–128763. https://doi.org/10.1109/access.2022.3227579

Kakavand, M., Dabbagh, M., & Dehghantanha, A. (2018). Application of machine learning algorithms for Android Malware detection. Proceedings of the 2018 International Conference on Computational Intelligence and Intelligent Systems. https://doi.org/10.1145/3293475.3293489

Kim, D. S., & Park, J. S. (2003). Network-based intrusion detection with Support Vector Machines. Information Networking, 747–756. https://doi.org/10.1007/978-3-540-45235-5_73

Liang, D., Lu, C., & Jin, H. (2017). Soft multimedia anomaly detection based on neural network and optimization-driven support Vector Machine. Multimedia Tools and Applications, 78(4), 4131–4154. https://doi.org/10.1007/s11042-017-5352-z

Liu, J., Feng, Y., Liu, X., Zhao, J., & Liu, Q. (2023). MRM-DLDet: A memory-resident malware detection framework based on memory forensics and Deep Neural Network. Cybersecurity, 6(1). https://doi.org/10.1186/s42400-023-00157-w

Lu, Y., & Roychowdhury, V. (2007). Parallel randomized sampling for support vector machine (SVM) and support vector regression (SVR). Knowledge and Information Systems, 14(2), 233–247. https://doi.org/10.1007/s10115-007-0082-6

Lu, Z., Sun, J., & Butts, K. (2014). Multiscale asymmetric orthogonal wavelet kernel for linear programming support vector learning and Nonlinear Dynamic Systems identification. IEEE Transactions on Cybernetics, 44(5), 712–724. https://doi.org/10.1109/tcyb.2013.2279834

Mirzamomen, Z., & Kangavari, M. R. (2016). A framework to induce more stable decision trees for pattern classification. Pattern Analysis and Applications, 20(4), 991–1004. https://doi.org/10.1007/s10044-016-0542-2

Moussas, V., & Andreatos, A. (2021). Malware detection based on code visualization and two-level classification. Information, 12(3), 118. https://doi.org/10.3390/info12030118

Mulay, S. A., Devale, P. R., & Garje, G. V. (2010). Intrusion detection system using support vector machine and decision tree. International Journal of Computer Applications, 3(3), 40–43. https://doi.org/10.5120/758-993

New Zealand : Cylance unveils Cylance Smart Antivirus; AI-powered antivirus for consumers. (2018, Jul 25). MENA Report https://reddog.rmu.edu/login?url=https://www.proquest.com/wire-feeds/new-zealand-cylance-unveils-smart-antivirus-ai/docview/2075904307/se-2

Notley, S., & Magdon-Ismail, M. (2018, June 12). Examining the use of neural networks for feature extraction: A comparative analysis using deep learning, support vector machines, and K-nearest neighbor classifiers. Semantic Scholar. https://www.semanticscholar.org/reader/93bcdd1d404787429ca7f79064421465492fcb2f

Novikov, A., Podoprikhin, D., Osokin, A., & Vetrov, D. (2015). Tensorizing Neural Networks. https://proceedings.neurips.cc/paper/2015/file/6855456e2fe46a9d49d3d3af4f57443d-Paper.pdf

Palša, J., Ádám, N., Hurtuk, J., Chovancová, E., Madoš, B., Chovanec, M., & Kocan, S. (2022, July 1). MLMD-A malware-detecting antivirus tool based on the XGBoost machine learning algorithm. MDPI. https://www.mdpi.com/2076-3417/12/13/6672

Pundir, N., Tehranipoor, M., & Rahman, F. (2020). RanStop: A Hardware-assisted Runtime Crypto-Ransomware Detection Technique. Cornell University Library, arXiv.org. https://www.proquest.com/docview/2464274904?parentSessionId=NxXr5ubs1QkrH3OTCNt%2FOYG1E0jkiEY2E83Y7co9Qb8%3D&accountid=28365

Rhode, M., Burnap, P., & Jones, K. (2018). Early-stage malware prediction using recurrent neural networks. Computers & Security, 77, 578–594. https://doi.org/10.1016/j.cose.2018.05.010

Rhode, M., Burnap, P., & Wedgbury, A. (2021). Real-time malware process detection and automated process killing. Security and Communication Networks, 2021, 1–23. https://doi.org/10.1155/2021/8933681

Rhode, M., Burnap, P., & Wedgbury, A. (2021, December 6). Artificial Intelligence and Machine Learning Algorithms for Cyber Attack Countermeasures. Security and Communication Networks. https://www.hindawi.com/journals/scn/2021/8933681/

Saad, H. (2018). The application of data mining in the production processes. Industrial Engineering, 2(1), 26. https://doi.org/10.11648/j.ie.20180201.14

Sandnes, F. E., & Eika, E. (2017). Statistics-IDE: Supporting the design of empirical experiments for non-experts during early stages of research projects. Intelligent Human Systems Integration, 502–507. https://doi.org/10.1007/978-3-319-73888-8_78

Smith, D., Khorsandroo, S., & Roy, K. (2022). Machine learning algorithms and frameworks in ransomware detection. IEEE Access, 10, 117597–117610. https://doi.org/10.1109/access.2022.3218779

S. S, S. K. A, V. Vignesh, and S. P, “Assessment of Zero-Day Vulnerability using Machine Learning Approach”, EAI Endorsed Trans IoT, vol. 10, Jan. 2024.

Wang, Z., Liu, C., & Cui, X. (2021). Evilmodel: Hiding malware inside of neural network models. 2021 IEEE Symposium on Computers and Communications (ISCC). https://doi.org/10.1109/iscc53001.2021.9631425

Xue, C. (2018). A discourse parser language model based on improved neural network in machine translation. 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS). https://doi.org/10.1109/icitbs.2018.00158

Zhang, M., Peng, H., & Yan, X. (2020). Improved algorithm of Decision Tree based on Neural Network. Journal of Physics: Conference Series, 1693(1), 012081. https://doi.org/10.1088/1742-6596/1693/1/01208

Xuwei Zhang, Yanfang Wang, Analysis of computer network information security and protection strategy. Advances in Computer, Signals and Systems (2024) Vol. 8: 41-50. DOI: http://dx.doi.org/10.23977/acss.2024.080306.