Vol.15 No.3(2024),193-204
DOI: https://doi.org/10.61841/turcomat.v15i3.14790

# CNN2D Algorithm for Detection of Ransomware Attacks Using Processor and Disk Usage Data

Ms. Putta Srivani [1], J. Uma Sri [2], K. Reshmitha [3], P. Soukya [4]

[1] Assistant Professor, Malla Reddy Engineering College for Women (Autonomous Institution) Hyderabad.

[2] [3] [4] Student, Malla Reddy Engineering College for Women (Autonomous Institution) Hyderabad.

## ABSTRACT:

Commonly, ransomware encrypts data, turns off antivirus protection, and destroys the target computer and everything on it. The techniques used today to detect this kind of WannaCry include monitoring the files, system requests, and processes on the system that is being targeted and analysing the data collected. Monitoring numerous processes has a substantial overhead; more current ransomware may interfere with the monitoring and alter the collected data. A dependable and practical technique for locating ransomware operating within a virtual machine, also called a VM, is provided in this study. We construct a framework for detection by applying an automated machine learning (ML) evaluation to the whole virtual machine (VM) using data collected from the physical host computer pertaining to specific processors and disc I/O events. This approach eliminates the need to continuously watch every action on the system that is being targeted and lessens the likelihood that ransomware would contaminate data. It also endures shifts in the amount of labour that users must do. It provides fast and very likely detection of known ransomware (used to train this machine learning model) and also of unknown ransomware (not utilised for teaching the model). Out of the seven artificial neural network classifiers that we looked at, the randomly generated forest (RF) classification gave the best results. Across six different customer loads plus 22 instances of ransomware, the RF model detected malware with a 0.98 confidence in 400 milliseconds.

## INTRODUCTION

By locking or encrypts data on the target computer, malware often referred to as ransomware renders the machine and its contents unusable. Ransomware attacks are used by cybercriminals to

Vol.15 No.3(2024),193-204
DOI: https://doi.org/10.61841/turcomat.v15i3.14790

extract victims' money. Nation-state actors might potentially use ransomware assaults as a means of causing harm to its adversaries' key infrastructure. These attacks usually include the evacuation of the consumers' data in an effort to coerce them into paying a ransom or trading the data on the dark web. In 2022, around 70% of businesses experienced ransomware attacks from rogue actors. Up from every nine seconds in 2021, ransomware is predicted to attack a business, individual, or gadget every two seconds by 2031. Vicente Alarcon-Aquino is the assistant editor who is in charge of coordinating the manuscript's assessment and approving publishing till 2031. Damage was estimated to have cost $20 billion in 2021 and is likely to exceed $265 billion. Recently, lots of investigators have examined the detection of ransomware attacks. Signature-based detection uses the hash values generated by antiviral programmes for popular ransomware to search the target computer for files that match overall hash values. Nonetheless, detection through signatures is not immune to polymorphic or metamorphic forms of known ransomware. Thus, behavioural or runtime ransomware detection improves signature-based detection methods. A dynamic research called behavioural analysis looks at the behavioural patterns of the virus, or the actions the ransomware does after it has gained access to the target computer. Although ransomware may take many various forms, it has to follow specific steps in order to quickly encrypt every document as possible. Some of the most recent ransomware software, such as LockBit2.0, Darkside, the Backmatter, only encrypt the first few bytes of files to be able to quickly render other files inaccessible. Because ransomware must quickly encrypt user data, its runtime behaviour will likely differ from that of a benign application. According to the theory, a system under ransomware assault must exhibit some kind of permanent aberrant conduct. The ransomware needs to access records from the diskette in order to protect the data, for example. Increased activity results from this, which may be identified by appropriately trained machine learning algorithms. Utilising runtime detection on the target system requires ongoing monitoring of various components, subsystems, and processes in addition to collecting and analysing event-related data to identify anomalous behaviour. Ransomware may try to hide its runtime activity by creating new processes and activities. Still, the fact remains that a martyr attack-affected system will exhibit higher activity with the correct analysis. If monitoring is done on the target system, runtime detection is resource-intensive and intrusive since several processes need to be monitored and it could be difficult to

Vol.15 No.3(2024),193-204
DOI: https://doi.org/10.61841/turcomat.v15i3.14790

identify the ransomware-related activities. Furthermore, this kind of monitoring is often disabled by ransomware designed to terminate processes before encrypting data. Special purpose registers called physical performance counters as well as or HPCs for short, track system and engine events for individual processes or the whole system. The hundreds of calculations and system events that the contemporary processors may count include the number of completed instructions, queue misses, plus off-chip memory accesses. Common applications for the data collected with HPCs include performance analysis as well as system software improvement. Still, a lot of recent research has focused on its use in virus detection. Alam et al. extracted HPC data from each process running on the system. However, because doing so might materially compromise the system's performance, it is not feasible to keep an eye on every action. Pundir et al. used machines to collect the data. The testing's scope is limited to a single the company virtual computer (VM) workload, however, and modifying the workload (by adding or subtracting applications) may have a significant impact on the accuracy of the identification.

## RELATED WORK

### Malware detection and analysis based on behaviour

Malware such as worms, Trojan horses, and spyware pose a major danger to the Internet. Even while ransomware and its variants may vary significantly from content signatures that are generated, we discovered that they have a few higher-level behavioural aspects in common that are more reliable at revealing the underlying intent of malware. This paper looks at the methodology of ransomware behaviour extraction in addition to outlining the formal Computer behaviour Featuring (MBF) extraction technique and offering a malware algorithm for identification based on hazardous behaviour characteristics. The results of our tests show that our MBF-based spyware identification system, if developed and put into use, can identify recently discovered unknown malware.

Vol.15 No.3(2024),193-204
DOI: https://doi.org/10.61841/turcomat.v15i3.14790

**Ran Stop is a runtime technique that uses hardware assistance to identify crypto-ransomware.**

Crypto-ransomware is one of the many malware programmes currently now in use, and it is especially hazardous since it has the ability to monetarily extort victims by encrypting their documents without their knowledge, holding them captive, and threatening legal action. Millions of dollars are lost annually as a result of this on a global scale. The number of variants of ransomware is growing because it may evade many antivirus products and software-only detection techniques for malware that rely on static behavioural patterns. In this work, we propose to detect crypto-ransomware infestations in commodity microprocessor early on by using hardware-assisted methods such as Rain Stop. Using information from hardware performance registers found in the performance control unit of modern CPUs, Rain Stop monitors micro-architectural occurrence sets and detects known and unknown crypto-ransomware variations. In order to assess micro-architectural incidents within the hardware of a device environment during the propagation of both benign and multiple ransomware variants, this study focuses on retraining a recurrent neural network-based artificial neural network architecture using an LSTM model. Through the use of worldwide average pooling as LSTM modelling, we create time series using the data from linked HPCs to develop intrinsic statistical properties that improve Rain Stop's detection rate and reduce noise. Rain Stop may be able to accurately and early identify ransomware within 2 milliseconds of the programme starting to run by examining HPC data collected at 20 timestamps separated by 100us. Given its early detection, a ransomware cannot now do much damage, if any. In addition, Ran Stop's ransomware detection accuracy is 97% on average over fifty random trials when tested against safe programmes that exhibit behavioural (sub-routine-centric) similarities to crypto-ransomware.

**Regard is a real-time detection technique for cryptographic ransomware.**

The widespread virus known as ransomware, which preys on people and company victims in an effort to achieve financial benefit, has recently experienced a resurgence. Because the existing

Vol.15 No.3(2024),193-204
DOI: https://doi.org/10.61841/turcomat.v15i3.14790

ransomware detection tools cannot provide an early warning in instantaneous fashion a large number of files are irrevocably encrypted, and post-encryption techniques (such key extraction and record restoration) have many limitations. Moreover, the detection systems that are now in use have a significant false positive rate since they cannot distinguish between ransomware encryption and the actual purpose of file alterations. Put another way, they are unable to discern between a user-initiated file activity (such innocuous encryption or compression) and a major file modification induced by ransomware. To address these challenges, we introduce in this paper Context, a ransomware detection algorithm that can detect crypto-ransomware on a user's computer in real-time by (1) employing decoy strategies, (2) closely monitoring the file system as well as operating interprets for suspicious activity, and (3) rejecting the flagging of harmless file changes by observing users' encryption patterns. We evaluate our strategy with samples drawn randomly from the 15 most prevalent families of cyberattacks to far. Our research shows that RWGuard is an excellent real-time ransomware detection tool with an inefficiency of just \(\sim \)1.9%, with zero fake negatives and minimal negative result (\(\sim \)0.1%) rates.

**Concerning whether it is possible to detect malware online using performance counters**

Malware spreads within every domain at the same pace as computers do. Even on the most recent mobile platforms, systems are plagued with malware of all kinds, including viruses, rootkits, spyware, and adware. Anti-virus (AV) software does not totally eliminate ransomware threats; in fact, there are more and more ways to get past AV protection. In actuality, contemporary attackers infect computers by taking advantage of security holes in antivirus software. In this study, we examine the feasibility of building a malware detection system in hardware utilising the performance counters that are already in use. We find that task counter data could potentially utilised to identify malware, and our detection techniques are immune to even the tiniest modifications in malicious software. As such, even after examining a small number of those versions, we are able to detect many variations within a malware species on the Intel Ubuntu and iPhone ARM platforms. Additionally, our proposed hardware modifications may allow the malware detector to function securely outside of the system software, opening the door to less

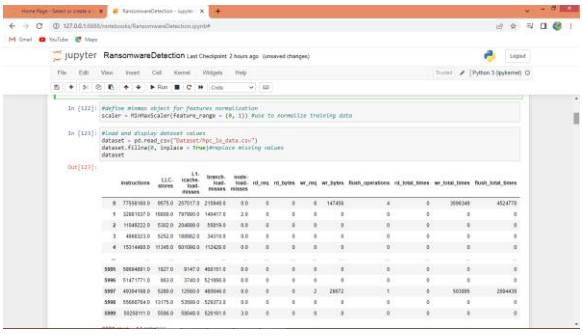
Vol.15 No.3(2024),193-204
DOI: https://doi.org/10.61841/turcomat.v15i3.14790

complex and error-prone AV systems compared with those that rely on software AV. Combining hardware antivirus techniques' security and resilience might lead to an improvement in state-of-the-art internet malware detection.

## METHODOLOGY

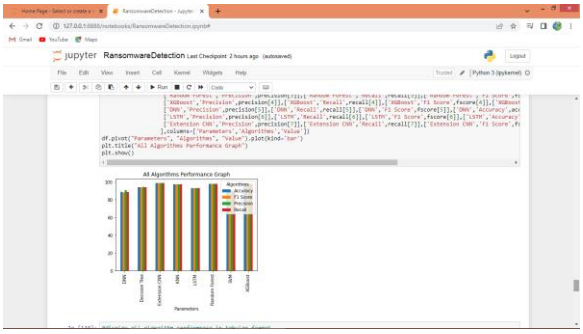
We have created the following modules in order to carry out this project. loading each and every package and class
2. Determining a class to standardise attributes
3. pre-processing the dataset by rearranging and normalising the values before showing the results.
4. Using 80% training data to train the SVM algorithm, 20% test data is used for prediction, and accuracy as well as additional metrics are computed. training the KNN algorithm 5. training the LSTM algorithm 6. training the CNN2D algorithm 7. training the Random Forest method 8. training the XGBOOST algorithm 9. training the DNN algorithm 10. training the LSTM algorithm 11. training the CNN2D algorithm 12. presenting the results of all the algorithms in a tabular format
13. based on test data prediction

## RESULT AND DISCUSSION

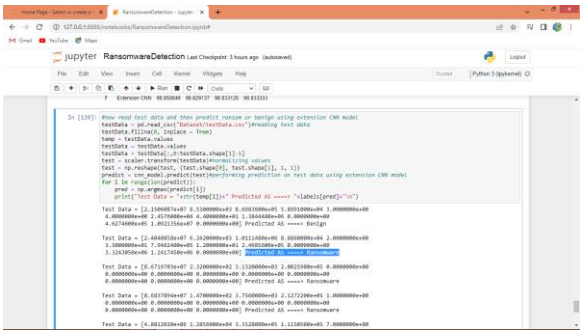


Vol.15 No.3(2024),193-204
DOI: https://doi.org/10.61841/turcomat.v15i3.14790

The class is defined on the screen above to normalise the characteristics, and the dataset values are then loaded and shown.



The algorithm names are shown on the x-axis in the above graph, while effectiveness and other metrics are shown on the y-axis in various colour bars for all algorithms Extension. CNN achieved good accuracy.



The test data values are shown in the output before the arrow symbol =[FSI], and the predicted values, "Ransomware or Benign," are displayed after the arrow symbol. The above screen shows the test data being read and the CNN algorithm object extension being used to do prediction on the test data.

## CONCLUSION

This article outlines a technique for quickly and accurately identifying ransomware that is operating on a VM, or virtual machine, by collecting the host computer's CPU as well as disc I/O traffic events, then analysing the data using machine learning techniques. Using the proc tool and

Vol.15 No.3(2024),193-204
DOI: https://doi.org/10.61841/turcomat.v15i3.14790

the equipment's performance counters (HPCs), recursive feature deletion with crossover validation is applied to determine five events from a pool that contains more than forty events in order to obtain processor-event data; disc I/O-event data is obtained for eight events using viral Domb stats. Five ML and two DL classifiers were considered. For every classifier, three models were developed: one that only utilised disc I/O data, one that solely used HPC data, along with a third that blended the two forms of data into a single model. The merged model performed the best across all seven classifiers. The random woodland, or RF, classifier performs better than the other three classifiers in terms of detection accuracy and needs less training time. Overall, the results of the RF-integrated model—which hadn't been utilised during training—in recognising ransomware, both known and unknown, are promising. We see two advantages when comparing our approach to earlier ones that monitor ransomware activities directly on the targeted machine. First, there is no overhead for the target computer since the monitoring is done from the host system. Secondly, ransomware running on the target machine cannot interfere with or distort the data collection process. By assessing the ML/DL-based models and detection on the desired virtual machine (VM), as well as under various user workloads, this study significantly advances the field. Most previous studies evaluate the models they employ for a specific workload situation. We observe that even when a machine learning model is trained on collected data in the absence of any user activity or background workload, it performs poorly in terms of detection performance when the user performs tasks such as visiting the internet, playing audio or video clips, or by employing productivity software like Adobe tools, Microsoft Word, or Excel. Our detection technique may be used to create virtual machines which can be detected at the host or kvm level. One advantage of this approach is that, regardless of the operating system used for the computer in question, the data collection process remains consistent. An further advantage is that the virus functioning inside the virtual computer remains unaware of the monitoring, meaning it cannot obstruct the data collection process. Using our detection algorithms, cloud hosting companies may provide their clients additional protection against ransomware attacks. The increasing trend of computers moving to the cloud and virtual machines (VMs) makes this additional ransomware protection pertinent and necessary. For this study, we did not particularly examine ransomware's capacity to steal data. We want to collect internet activity from the chosen virtual machine (VM) that's running

Vol.15 No.3(2024),193-204
DOI: https://doi.org/10.61841/turcomat.v15i3.14790

on the host server and investigate potential exfiltration activities using Hcp and I/O data analysis. Any and every data collected during ransomware activities is categorised as dangerous. However, some ransomware examines the target machine and network covertly for a considerable amount of time (tens of thousands of seconds or a greater extent) before to commencing its destructive activities. This kind of espionage generates noisy data when it is used; the virus remains idle for a considerable portion of the experiment's life; the information gathered defaults to matching the system load but is categorised as hazardous. Given that our objective is to detect malware who are actively altering files, that sort of information may lead to a less accurate detection. Assume that the data collected during encryption operations may be classified as harmful, and the data collected during the reconnaissance phase as scouting. We believe that at that point, the reliability of detection will increase and function as a more trustworthy indicator of virus in its most destructive form. Our goal for our next project is to investigate more accurate data tagging. In our next study, we want to apply additional degrees of workloads to increase the detection models' robustness to shifting user behaviour. We assessed the models we presented in this study using information obtained from further testing cycles. In the future, we want to use the algorithms for real-time identify ransomware while it's still active. While our notion is only applicable to virtual machines currently, we want to adapt it in future work to freestanding desktops as well. The models' applicability for a system configuration with different memory or CPU core counts, or one substantially less memory, has not been evaluated. In the not so distant future, we would want to investigate this

## REFERENCES

[1] SR Department. (2022). Ransomware victimization rate 2022. Accessed: Apr. 6, 2022. [Online]. Available: https://www.statista. com/statistics/204457/businesses-ransomware-attack-rate/

[2] D. Braue. (2022). Ransomware Damage Costs. Accessed: Sep. 16, 2022. [Online]. Available:

Vol.15 No.3(2024),193-204
DOI: https://doi.org/10.61841/turcomat.v15i3.14790

[3] Logix Consulting. (2020). What is Signature Based Malware Detection. Accessed: Apr. 3, 2023. [Online]. Available: https://www.logixconsulting. com/2020/12/15/what-is-signature-based-malware-detection/

[4] W. Liu, P. Ren, K. Liu, and H.-X. Duan, ‘‘Behaviour-based malware analysis and detection,’’ in Proc. 1st Int. Workshop Complex. Data Mining, Sep. 2011, pp. 39–42.

[5] (2021). Polymorphic Malware. Accessed: Apr. 3, 2023. [Online]. Available:

[6] M. Loman. (2021). Lock file Ransomware’s Box of Tricks: Intermittent Encryption and Evasion. Accessed: Nov. 16, 2021. [Online]. Available:

[7] N. Pundir, M. Tehrani poor, and F. Rahman, ‘‘Ran Stop: A hardware assisted runtime crypto-ransomware detection technique,’’ 2020, arXiv:2011.12248.

[8] S. Mehnaz, A. Budgerigar, and E. Bertino, ‘‘Regard: A real-time detection system against cryptographic ransomware,’’ in Proc. Int. Symp. Res. Attacks, Intrusions, Defences. Cham, Switzerland: Springer, 2018, pp. 114–136.

[9] J. Demme, M. Maycock, J. Schmitz, A. Tang, A. Waksman, S. Seth Madhavan, and S. Stolfo, ‘‘On the feasibility of online malware detection with performance counters,’’ ACM SIGARCH Compute. Archit. News, vol. 41, no. 3, pp. 559–570, Jun. 2013.

[10] A. Tang, S. Seth Madhavan, and S. J. Stolfo, ‘‘Unsupervised anomaly-based malware detection using hardware features,’’ in Proc. Int. Workshop Recent Adv. Intrusion Detection. Cham, Switzerland: Springer, 2014, pp. 109–129.

[11] S. Das, J. Werner, M. Antonakakis, M. Polychronakis, and F. Monrose, ‘‘SoK: The challenges, pitfalls, and perils of using hardware performance counters for security,’’ in Proc. IEEE Symp. Secure. Privacy (SP), May 2019, pp. 20–38.

Vol.15 No.3(2024),193-204
DOI: https://doi.org/10.61841/turcomat.v15i3.14790

[12] S. P. Kadiyala, P. Jadhav, S.-K. Lam, and T. Srikanthan, ‘‘Hardware performance counter-based fine-grained malware detection,’’ ACM Trans. Embedded Compute. Syst., vol. 19, no. 5, pp. 1–17, Sep. 2020.

[13] B. Zhou, A. Gupta, R. Jahanshahi, M. Egale, and A. Joshi, ‘‘Hardware performance counters can detect malware: Myth or fact?’’ in Proc. Asia Conf. Compute. Common. Secure., May 2018, pp. 457–468.

[14] S. Aurangzeb, R. N. B. Rais, M. Aleem, M. A. Islam, and M. A. Iqbal, ‘‘On the classification of Microsoft-windows ransomware using hardware profile,’’ Peer Compute. Sci., vol. 7, p. e361, Feb. 2021.

[15] M. Alam, S. Bhattacharya, S. Dutta, S. Sinha, D. Mukhopadhyay, and A. Chattopadhyay, ‘‘RATAFIA: Ransomware analysis using time and frequency informed autoencoders,’’ in Proc. IEEE Int. Symp. Hard. Oriented Secure. Trust (HOST), May 2019, pp. 218–227.

[16] K. Thumbpad, R. Boppana, and P. Lama, ‘‘HPC 41 events 5 rounds,’’ Harvard Dataverse, 2022, doi: 10.7910/DVN/MA5UPP.

[17] K. Thumbpad, R. Boppana, and P. Lama, ‘‘IO 41 events 5 rounds,’’ Harvard Dataverse, 2022, Doi: 10.7910/DVN/GHJFUT.

[18] K. Thumbpad, R. Boppana, and P. Lama, ‘‘HPC 5 events 7 rounds,’’ Harvard Dataverse, 2022, Doi: 10.7910/DVN/YAYW0J.

[19] K. Thumbpad, R. Boppana, and P. Lama, ‘‘Io 5 events 7 rounds,’’ Harvard Dataverse, 2022, Doi: 10.7910/DVN/R9FYPL.

[20] K. Thumbpad, R. Boppana, and P. Lama, ‘‘Scripts to reproduce results,’’ Harvard Dataverse, 2023, Doi: 10.7910/DVN/HSX6CS.

[21] M. Rhode, P. Burnap, and A. Wedgbury, ‘‘Real-time malware process detection and automated process killing,’’ Secure. Common. Newt., vol. 2021, pp. 1–23, Dec. 2021.

Vol.15 No.3(2024),193-204
DOI: https://doi.org/10.61841/turcomat.v15i3.14790

[22] A. Kharrazi and E. Karda, ‘‘Redemption: Real-time protection against ransomware at end-hosts,’’ in Proc. Int. Symp. Res. Attacks, Intrusions, Defences. Cham, Switzerland: Springer, 2017, pp. 98–119.

[23] F. Mabolo, J.-M. Robert, and A. Salishan, ‘‘An efficient approach to detect torrent locker ransomware in computer systems,’’ in Proc. Int. Conf. Cryptal. Newt. Secure. Springer, 2016, pp. 532–541.

[24] K. Lee, S. Lee, and K. Yim, ‘‘Machine learning based file entropy analysis for ransomware detection in backup systems,’’ IEEE Access, vol. 7, pp. 110205–110215, 2019.

[25] C. J. Chew and V. Kumar, ‘‘Behaviour based ransomware detection,’’ in Proc. Int. Conf. Compute. Their Appl., in Epic Series in Computing, vol. 58. 2019, pp. 127–136