A Signature based ransomware detection using convolutional neural network

MSc Research Project

Cyber security

Rahul Selvakumar

Student ID: X20231296

School of Computing

National College of Ireland

Supervisor: Niall Heffernan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Rahul Selvakumar
StudentID:	X20231296
Programme	MSc in Cybersecurity **Year:** 2021-2022
Module:	Research Project
Lecturer:	Niall Heffernan
Submissin Due Date:	15/08/2022
Project Title:	A Signature based Ransomware detection using CNN

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Rahul Selvakumar

Date: 15/08/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Signature based Ransomware Detection Using Convolutional Neural Network

Rahul Selvakumar

20231296

Abstract

In recent years, ransomware has been one of the most common types of cybersecurity threats to well-known businesses and organizations. Ransomware is a type of cryptovirology malware which threatens to block and publish the data on a computer system by encrypting it. Hence, there is a need to develop an effective method for detecting ransomware. Most of the proposed methods were used in identifying the ransomware during the execution stage. It's hard to say how long a programme needs to be examined to display its actual behaviour. In this paper, the ransomware is detected using Sha1 and MD5 signatures using a convolutional neural network. A convolutional neural network is used since it has the ability to extract and categorise the features using images and classify them with high accuracy. The proposed method achieves 97.04% accuracy. The results show that the technique is helpful and feasible for ransomware detection.

Keywords: Ransomware, Deep learning, CNN, signature-based

1. *Introduction*:

Due to the growth of computing and communication technologies, there has been an enormous growth in different types of threats. Which leads to the necessity of increasing the security standards that have created a critical challenge to the antivirus software producing industry. In recent years, ransomware has become one of the popular types of threat that either encrypts, obfuscates or block access to a victim's assets and data until the ransom is paid. Even when the ransom is paid, the recovery of the data or file is often not guaranteed. Ransomware is classified into two types: crypto ransomware and locker ransomware, which blocks access to the data respectively. The ransomware has the potential to duplicate and propagate to other devices or across the entire network. These attacks are common across many industries, and it is quite possible that any firm will become a victim of ransomware.(liska and gallo, 2316)

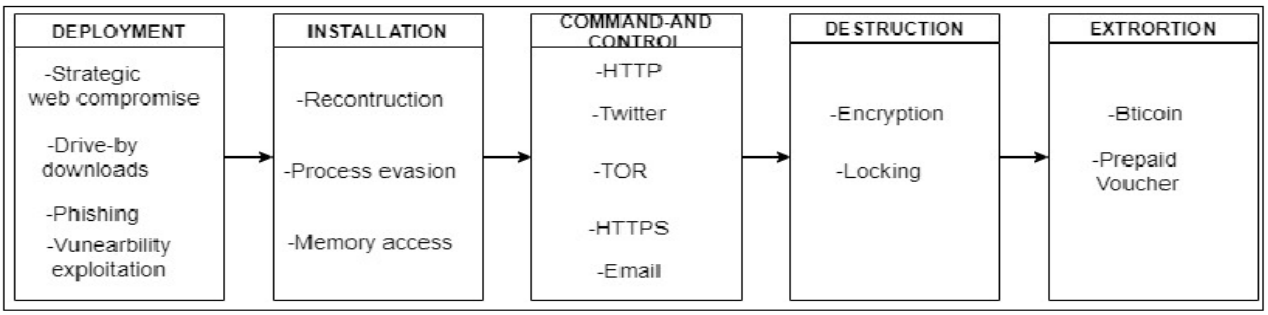


Fig 1: Working of ransomware

By targeting organisations, ransomware becomes a profitable venture for hackers. The vulnerable systems are the major cause of ransom attacks; attackers try to locate older versions of an application, operating systems, or by continuous traffic monitoring, browsing habits, or analysing email.

Deep learning techniques can be used in effective ransomware detection methods. In this research paper, CNN is used for detecting ransomware. CNN has the ability to process large amounts of unstructured data. Since CNN works with image processing, the result will be provided in the shortest amount of time and by CNN. Since this study is based on signature-based ransomware detection, the output must be obtained quickly with highly accurate recognition results, which can be provided by CNN. And CNN can be retrained in order to perform new tasks, which can be enabled by building pre-existing networks. While talking about other neural networks, the processes are classified based on the distinct nodes, which requires more time in order to achieve high accuracy. The scope of this study is based on signatures. The term "signature" refers to a unique "digital footprint" that is created by each piece of software and code that is used to represent itself in a system. (bhatia, 2018)(*Why convolutional neural network is better? - Intellipaat Community*, no date)

Based on the statistics given by Forbes, approximately 37% of the global organisations are falling victim to ransomware attacks in 2021, around which 304.7 million ransomware attacks happened in the first half of 2021, which is 151% higher than the year 2020. Around 80% of organisations fall victim to ransomware in the year 2021. The FBI and IC3 received 2084 ransomware cases in the first part of 2021. According to the data that was provided by virustotal, 95% of windows-based machines fall victim to ransomware. Some of the major attacks in an organisation and ransomware groups.(*73 Ransomware Statistics Vital for Security in 2022 - Panda Security Mediacenter*, no date)

Organisation	Attacking method	Attacking Group
Apple supplier quanta computers	VIA E-mail	Sodinokibi
Jet manufacturer bombardier	Social engineering	Clop
Crozer-Keystone Health System	Using employees VPN	Mailto
JBS	RDP Vulnerability	REvil group

2. Literature review:

2.1 Static analysis:

(Lee *et al.*, 2022) In this paper the test were conducted based on the EDR tool (End point detection). It was conducted to check if the EDR was able to identify the file change caused by the ransomware. GRR, osquery and OSSEC were the three tools that were selected for end point detection. The result of these detection were shown in the form of notification and in the form of logs especially for the files that were encrypted using all the three EDR tools. It is feasible to identify the moment at which the ransomware was run and examine the attack mechanism using this open source-based threat detection. Although this study was not evaluated under different environmental and settings, future work will assess the effectiveness of the EDR tool by comparing the accuracy and speed with which EDR identifies ransomware when deployed to big clients.

(Manavi and Hamzeh, 2021) proposed a LSTM network which is being used since it remembers information for a long period of time. The ransomware files were passed through the LSTM network with the help of the signature based technique the LSTM network were used to identify the ransomware with the help of PE headers. (Sheen and Yadav, 2018*)* proposed a method where the ransomware is detected by extracting the API calls in order to identify the ransomware and benign files. The

researchers used large number of imbalanced dataset. In order to avoid that smote was used. And then it was discovered that random forest applied to the balanced training set after smote produced the greatest results .The future work can be derived from dynamic analysis must also be evaluated.

(Ko *et al.*, no date) The researcher analysed two previous studies to detect the new variant of ransomware in real time. And they proposed a new method using Shannon entropy to detect the new variant of ransomware with less performance in an android environment. They focused on main functions by monitoring already analysed process. In a future study, the proposed methodology should be verified for detection accuracy utilising comprehensive analytical approaches. (M, no date) In this paper the researchers have detected and classified the ransomware using gradient tree boosting algorithm using static approach. Ransomware detection using gradient tree boosting algorithm to achieve better result and accuracy. The future work can be done by detecting and classifying the ransomware type.

(Kok et al., 2019) By using IRPs and APIs, they developed an early detection method for crypto malware. As a result, the threshold served as a border to divide characteristics related to crypto-ransomware lifecycles prior to encryption. In order to overcome data scarcity during this phase, they used the LSTM method to train both a data-centric and a process-centric model using the LSTM method. With the help of combining data-centric and process-centric techniques, they were able to gather more ransomware patterns, which eventually increased the strength of this strategy, which enabled crypto ransomware attacks to be identified before they were encrypted. Using this approach, a stronger defensive solution was developed by precisely characterizing the pre-encryption phase of ransomware attacks.

(Alqahtani, Gazzan and Sheldon, 2020) In this paper the process are splitted into three parts in order to detect the ransomware at the early stage. *Pre-Encryption Boundary Definition* At the first stage, the boundaries are accurately separated in a crypto-ransomware lifecycle. In order to address the attack at the initial stage before the encoding happens and to allow the model to collect enough information for a given period of time, which helps to avoid the premature cut-off from existing solution. *Pre-Encryption Feature Extraction*: Once the pre-encryption boundaries are defined the phase will be extracted using DTF-IDF. The result of this DTF-IDF technique helps to extract the feature which will be used to train the detection model. *A Hybrid Deep Learning-Based Crypto-Ransomware Early Detection Model.* With the help of the early phase the attack are classified into two types data-centric and process-centric which will fused to train the deep learning- based detection which are used to provide more insight about the attack. The LSTM were also used to build the detection model.

(Bahrani and Bidgly, 2019) In this paper the ransomware were detected using process mining. In order to detect the ransomware, the features was first extracted from the process model and with the help of these feature the classification was done. The end result of random forest and J48 had the best accuracy for the classifier in the proposed method. The drawback is that the static analysis of other neural networks were not stated.

2.2 Dynamic analysis:

(Almousa, Basavaraju and Anwar, 2021) In this paper the researcher used an API based obfuscation technique to detect the ransomware. The ransomware has the tendency of changing the behaviour so the researcher changed the flow of API to the detection of the ransomware by analysing each file for 10 minutes using the KNN classifier and hyper parameter tuning the results were collected which achieved high accuracy. This experiment was carried out by extracting the logs the logs thought the

sandbox environment. Even though this model detected high accuracy for unknown samples this research can be expanded by improving the file system activity and network activity.

(Baek *et al.*, 2021) The researcher proposed an SSD based ransomware detection and a data recover technique called as SSD-Insider++. Based on the behavioural analysis the SSD Insider++ was able to detect the ransomware at early stage. SSD Insider++ proven a strong indicator of ransomware activity it also provided perfect data recovery by update nature of NAND flash. The drawback in this model is that SSD-Insider++ provided a sluggish detection approach to solve the worst-case scenario in which it could not recognise ransomware assaults. The detection/recovery algorithms were implemented in an open-channel SSD to demonstrate the possibility of SSD-Insider++.

(Noorbehbahani and Saberi, 2020) In this paper, ransomware were detected using semi-supervised method and feature selections were examined. The result showed that the RF classifier and Chi-squared selection methods were very effective in detection of ransomware. The drawback of this method is that the applied feature selection methods are supervised. In order to overcome this Simplified Silhouette Filter (SSF) were used which resulted in poor detection. As a future work it is required to examine and propose a semi-supervised feature selection approach for ransomware detection.

(Zhang, Wang and Zhu, 2022) the main agenda was to detect the unknown encrypted ransomware attack. In order to perform that dual generative adversarial were used called as TGAN-IDS. TGAN-IDS, DCGAN were used to generate strong pseudo sample generation ability. TGAN was mainly used to discriminate strong pseudo sample. In order to improve the performance transfer learning mechanism was used. The discriminator of TGAN is produced as the final anomaly detector after adversarial training. The target function of TGAN is updated by including a reconstruction loss function to successfully suppress the problem of normal sample detection rate dropping during adversarial training of TGAN. As a result TGAN-IDS has the ability to improve the ability to detect the unknown ransomwares attacks. As future work, more malicious attacks will be attempted to identify, and the model structure will be modified further to increase detection performance of unknown attacks.

(Urooj, Aizaini Bin Maarof and Ali Saleh Al-Rimy, 2021) In this paper the Processes are divided into four prosp*er-Encryptions boundary definition:* At this stage the API were used to distinguish pre encryption with actual encryption processes. And used pseudo rocchio relevance feedback technique to define the boundaries.*Pre-Encryption Feature Extraction.* At this stage the ransomwares were labelled such as ransom or benign. *Pre-Encryption Feature selection*: Enhanced mutual information technique were used to select the feature. The feature used in this stage were used to train the classifiers of self- learning model *online classifier:* At this stage a stochastic gradient decent is used to detect a new ransomware types (Zero-day). This will be updating itself based on the new variant.

(Salehi *et al.*, 2018) In this paper two types ransomwares were analysed DGA based and HSR based ransomware. The analysis were classified into six different sections. At the initial stage the data will be monitored under the supervision of the expert. At the second stage the data were pre-processed with different classes of extraction. The extraction was done by analysing the DNS traffic as follows: *Random and Gibberish Characters in the Domains:* is a technique used to find the legitimate domain that are registered. *The Frequency of Different Domains:* Ransomware can be judged by how often it creates new domains, which can be found by looking at DNS traffic. *The Replication of the Same Domains in a Time Interval:* To analyse the domains are repeated. The fourth stage is used to detect the ransomware using gibberish detector. At the fifth stage the domains were blacklisted and whitelisted based on the above stages and the final stage was to analyse the network traffic which prevents other network being infected which is done using INETSIM.

(Sibi Chakkaravarthy *et al.*, 2020) In this paper the researcher proposed a novel IDH. This IDH utilizes a CEP technique which uses host, network feature and various events from the systems such as audit watch and firewall in order to detect the ransomware without minimal loss of data and to increases the accuracy and it can be deployed in the production environment easily. In order to detect the ransomware at the host, honey folder can be deployed. And to improve the protection of the network SDN infrastructure were used by applying simple control rule. As the feature work the IDH method can be used in transferring the learning ability to optimize the load and auto tuning feature can also be added for the proposed IDH.

(Agrawal *et al.*, 2019) The author used a novel method for detection of ransomware. The author performed a complete analysis of the ransomware in order to identify the structural properties which can be bypassed using machine learning system. The ransomware were identified within the long sequence of pattern by repeating corresponding encryption operations. Recurrent neural network was used to exploit the repeating patterns by bringing the attention on the input sequence learning module. And they used an ARI-LSTM, by observing the ransomware dataset. ARI-LSTM gave a better result compared to LSTM for identifying the ransomware dataset. The researcher presented strategy for including attention at sequence inputs using the ARI cell, which may be exploited by issues sensitive to relations within recent inputs. In this paper Static feature not included. (Silva and Hernández-alvarez, 2017) proposed a method where the data are stored in EcuCERT logs in order to study the behaviour of the threat which helps the user to identify the malicious activity. In this method previous year's data are collected and stored which was detected by the machine learning technique (Cognitive Security).

(Ayub, Continella and Siraj, 2020)) In this research paper the ransomware detection were conducted in a specific area which is among I/O request packet (IRP) logs is one of the best tool. In this paper 18 different ransomware families were analysed using data driven approach for detecting the ransomware and for analysing the behaviour. ANN Artificial neural network was used to detect the ransomware with three different experimental settings which gave high accuracy. The drawback of this paper is that this approach cannot be used for real time detection since the detection is based on ransomware infection logs. (Poudyal and Dasgupta, 2020) In this paper the researchers proposed an AI based ransomware detection framework and designed a ransomware analysis tool (AIRaD) using reverse engineering technique which were used to extract the multi-level feature by dynamic analysis. The ransomware behavioural chain is a novel technique that generates distinct ransomware detection signatures. Similarly, multi-level association rule mining showed distinct sequences. As a future work the ransomware family can be differentiated based on specifications and behaviour, allowing the ransomware detection tool to become fully operational

(Almousa, Basavaraju and Anwar, 2021) The malware was discovered in this article utilising an API-based method, along with k-NN and hyper parameter tweaking. The researcher altered the control flow of an API to escape detection by anti-malware software. The researcher retrieved the logs using the sandbox environment and passed them to the k-NN, which produced the high frequency. Despite the high frequency, the researcher stated that additional expansion can be done by increasing the analysis of file system activity and network activity logs, and that a multi-layer security solution may be implemented to guard against data ransomware assaults.

2.3 Hybrid analysis:

An efficient way of ransomware detection was used in identifying the ransomware by (Naik *et al.*, 2019).the researchers combined the hashing and clustering method to detect the ransomware. This method was applied by collecting ransomware samples by fuzzing hashing methods SSDEP,

SDHASH, and clustering methods such as K-Mean, PAM, AGNES, DIANA, CLARA. The result of the clustering was verified using three evaluation indexes Dunn index, Silhouette index and connectivity index in order to determine the accuracy and to maintain eh result at the same range. This method can used as both for dynamic and static analysis. As a result the clustering findings revealed that two clustering approaches, AGNES and DIANA, outperformed other methods. Furthermore, both the SSDEEP and SDHASH techniques worked well and discovered similarities in the majority of the cases.

(Medhat *et al.,* 2020) In this paper a hybrid based framework was used in detecting the ransomware samples based on memory dump and by YARA rule. The framework's computational costs can be minimised by employing whitelisting to prohibit contact with certain processes related to encryption, compression, and register-key management programmes. Furthermore, fine-tuning Windows security features for early prevention, such as AppLocker and application restriction rules, may increase overall system security dramatically. The drawback of this paper is that more diverse data set can be used to improve the coverage and detection. Furthermore, the memory dump technique can be adjusted such that it is unaffected by the protective measures of specific packers.

(Aljubory and Khammas, 2021) In this paper a new approach was suggested based on the static analysis in order to limitation of the dynamic analysis. The main concept of this paper is to extract the samples from byte-level. To improve the detection rate CF-NCF with n-gram was used. And for classification process SVM, RF and NB were effectively used to detect the ransomware. For future works Based on static analysis, a novel technique will be created to distinguish between ransomware and conventional malware.

(Shaukat and Ribeiro, 2018) Based on the analysis of the dataset. The researchers presented a layer of defence mechanism with novel compact by implementing a Controlled Sandbox Environment execution, as utilised in the experimental investigation of Ransomware variations. Early detection, zero day intrusion, preserving user data helped the ransom wall to obtain high accuracy with zero false positive gradient. As a future work ransom wall can be created on large-scale real setup.

(Salehi *et al.*, 2018) In this paper a new approach (DGA) was used to detect the ransomware. The main concept of this paper is to analyse the DNS traffic. The researchers proposed a method to stop the ransomware before it can develop the public key exchange procedure and disconnect from the command and control server. The ransomware before it can develop the public key exchange procedure and disconnect from the command and control server. This first approach focuses on DGRs in order to detect the ransomware at early stage to prevent the data from being encrypted. The results of the testing demonstrate this method can detect the most of DGRs. Second, it may be used to identify various types of malware, such as botnets and malware-based DGAs. Third, our suggested strategy may be used with other non-network traffic-based detection approaches to identify a wide spectrum of ransomwares.

2.4 Research Niche:

Author	ML/DL Model	Feature types	Shortcoming	Motivational Positives
(Manavi and Hamzeh, 2021)	LSTM network	Static analysis	Not able to differentiate ransomware and other types of	Used LSTM network to process the sequence and classify the ransomware.

			malware based on static analysis	
(Lee *et al.*, 2022)	EDR tools were used	Static analysis	This study cannot be evaluated under different environment	EDR tools such as GRR, osquery and OSSEC were used for end point detection.
(Ko *et al.*, no date)	Shannon entropy	Static analysis	This method cannot determine the accuracy using comprehensive analytical approaches	Shannon entropy were used to detect the new variant of ransomwares.
(Kok et al., 2019)	Used IRPs and APIs for early detection	Static analysis	Dynamic is limited	LSTM were used to train both data centric and process centric models which can be used as strong defensive solution.
(Alqahtani, Gazzan and Sheldon, 2020)	DTF-IDF	Static analysis	This can be carried out using genetic algorithm to improve prediction	LSTM networks were used to build the detection models.
(Bahrani and Bidgly, 2019)	RF and J48	Static analysis	Static analysis of other neural networks were not stated.	The ransomware were detected using process mining.
(M, no date)	Gradient tree bosting algorithm	Static analysis	Ransomware types cannot be found	Gradient tree bosting algorithm were used with static approach for detection.
(Almousa, Basavaraju and Anwar, 2021)	KNN	Dynamic analysis	This can be expanded by improving file system activity and network activity	API based obfuscation technique were used to detect the ransomware
(Baek *et al.*, 2021)	SSD-Insider++	Dynamic analysis	Smaller Dataset	The researcher proposed an SSD based ransomware detection and a data recover technique called as SSD-Insider++.
(Noorbehbahani and Saberi, 2020)	RF, Chi-squared selection method	Dynamic analysis	It is better to propose a semi supervised feature for ransomware detection	RF, Chi-squared selection method were used for detection and SSF were used which resulted in poor detection.
(Zhang, Wang and Zhu, 2022)	TGAN-IDS. TGAN-IDS, DCGAN were used to generate strong pseudo	Dynamic analysis	The model can be modified to increase the detection	TGAN was used to discriminate strong pseudo sample and TGAN-IDS was used to improve the ability and

	sample generation ability.		performance of unknown attacks.	detect the unknown ransomwares attacks
(Urooj, Aizaini Bin Maarof and Ali Saleh Al-Rimy, 2021)	API and pseudo rocchio relevance feedback technique was used	Dynamic analysis	Malware mentioned in the dataset were limited.	API were used to distinguish pre encryption and pseudo rocchio relevance feedback were sued to define the boundaries
(Salehi *et al.*, 2018)	DGA and HSR	Dynamic analysis	Only two types of ransomware were detected	The ransomware were classified based on DNS traffic.
(Sibi Chakkaravarthy *et al.*, 2020)	IDH and CEP technique	Dynamic analysis	The suggested IDH may be utilised to transmit the learning ability to optimise the load, and an auto tuning option can also be implemented.	IDH and CEP were used to detect the ransomware and SDN infrastructure were used for applying simple control rule.
(Agrawal *et al.*, 2019)	ARI-LSTM	Dynamic analysis	Static feature not included.	Novel method ARI-LSTM were used to detect the ransomware. Which gave a better result than LSTM.
(Silva and Hernández-alvarez, 2017)	Cognitive security	Dynamic analysis	Smaller Dataset	EcuCERT were used to study the behaviour.
(Ayub, Continella and Siraj, 2020))	ANN	Dynamic analysis	This approach cannot be used for real time detection. Since it is based on ransomware infection logs	IRP were used to analyse the logs which were then given as input to ANN.
(Poudyal and Dasgupta, 2020)	AIRaD tools was used for reverse engineering technique	Dynamic analysis	This method cannot different the ransomware types.	AIRaD were used to extract multi-level association.
(Naik *et al.*, 2019)	Hashing and Clustering	Hybrid analysis	Static feature not included.	Dashing methods SSDEP, SDHASH, and clustering methods such as K-Mean, PAM, AGNES, DIANA, CLARA were used in which SSDEEP and SDHASH techniques worked well and discovered similarities in the majority of the cases.
(Medhat *et al.*, 2020)	Memory dump and YARA rules	Hybrid analysis	The drawback of this paper is that	The ransomware samples were detected based on

			more diverse data set can be used to improve the coverage and detection	memory dump and YARA rule.
(Aljubory and Khammas, 2021)	NB ,SVM and RF	Hybrid analysis	This method cannot different the ransomware.	In this paper the extracted samples from byte-level. To improve the detection rate CF-NCF with n-gram was used.
(Shaukat and Ribeiro, 2018)	Sandbox environment	Hybrid analysis	The ransomware are created in small scale.	A layer of defence mechanism were created used sandbox environment
(Salehi *et al.*, 2018)	DGA	Hybrid analysis	Multi-layer security can be added to protect the curtail data ransom attack	DGA was used to analysis the DNA traffic. The researchers proposed a method to stop the ransomware before it can develop the public key exchange procedure.

3 Research Methodology

In this paper, CNN is used to detect the ransomware. This research shows that CNN is capable of handling a massive volume of unstructured data. Based on this research, the ransomware should be detected in the shortest period of time with high accuracy. And CNN has the ability to identify the key characteristics without any human supervision. CNN outperforms other networks because it can provide parameter sharing and dimensionality reduction, which other networks cannot. Whereas the processes of other neural networks are often divided into several nodes, which has a tendency to be a time-consuming procedure in order to bring about precision. In CNN, the different layers are convoluted into different filters in order to yield an accurate result for abstract and invariant features.

CNN uses images as its primary source of data, which it then processes and sorts into a different categories. The CNN will take in an image and an array of pixels as its input, and the quality of the image will determine the size of the array. The picture will be evaluated based on its H- height, B- breadth, and D- depth. The layers of CNN are broken down into the following

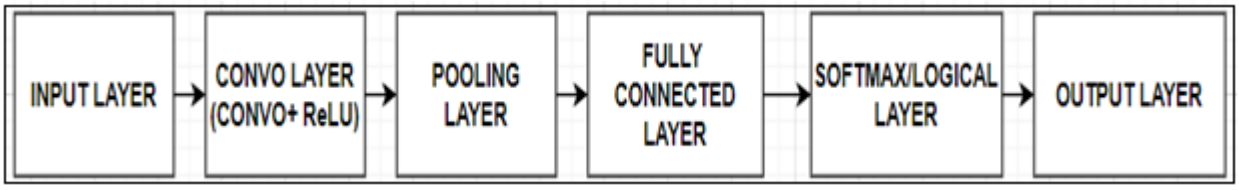


Fig 2: Working of CNN

3.1 Dataset description:

While analysing the malware in the RISS dataset, ransomware took up the majority of the space. Several ransomware strains employ complex packaging strategies, making structural analysis difficult.

The RISS dataset includes an application programming interface (API), a registration key, directory operations, a dropped file, and an embedded string. The dataset includes data that has been encrypted (PDF). Using a Pre-Encryption Detection Algorithm to Prevent Crypto-Ransomware. The dataset contains both crypto-ransomware and locky-ransomware that has been encrypted. This dataset includes 942 goodware, together with 582 ransomware that belong to 11 distinct families. (*Ransomware Dataset – RISS*, no date)

3.2 Clustering of data:

CNN uses images as its input form of data. Whereas in the "RISS dataset", the inputs are in the form of numerical values or string values. These sets of values will be converted into a form of 2D matrix. The CNN makes use of a 2D convolutional layer, since the results of the 2D convolutional layer will be summed into a single output result at the end of the layer it is the point at which the two-dimensional matrix will change into two distinct dimensional matrix feature. In order to form a grey scale image the dataset should be in the form numerical value. The image will be in the form of BGR. This BGR image will then be converted into an RGB image and then, from the RGB image, a grayscale image will be formed. To increase efficiency of CNN, these images are kept in form of grayscale. In order to turn all the character features in the dataset into numerical features, labels for the features in the dataset are encoded by dividing each character feature into its own column with 0 or 1 accordingly. Once the label are encoded the dataset is ready for training. In order to train the CNN, data must be filtered correctly within the network to avoid empty spaces. Which may come from different sources within the same range. If it is not filtered correctly, the learning process will slow down and it will affect the accuracy. For example let's say that the data ranges from 0 to 1, 00,000, which is an enormous difference. Now the whole range will be divided by larger values, so that the result will be decimal number. Numerical features will be applied for simple linear process.

$$A' = (A - A_{min}) / (A_{max} - A_{min})$$

$\mathbf{A_{max}}$ -features highest value, $\mathbf{A_{min}}$ -features lowest value

Once the normalisation is done, then the dataset is ready for training.

3.3 Matrix conversion:

Although the image has been pre-processed for training, it is usually stored as an array. To process the dataset, it must be changed into a matrix, which is then modified into grayscale images. The features that can be converted into matrices are converted. The conversion of some features into matrices is not possible due to their small values. In order to overcome this, repeating random features are used to expand and then convert them into a matrix. As a result of this process, the network becomes more stable, which helps in obtaining structural information features by extracting the correlations between them. The feature in the dataset are converted into images without any data loss which will be maintained throughout the process for perfect accuracy. The pixel value of each image will be divided by 255 since it is the highest possible value of the pixel. The output of this process will be given as the input to train the processes.

3.4 Model training:

The similar process is used to the other part of the dataset. In this phase the filters are added every model consist of an input layer followed by three convolutional layer and three pooling layer. A drop put layer is used to connect the connected layer and flattening layer this is done for achieving regularisation. Retified Liner Unit is added to each layer because they are simple, does not suffer from

gradient vanishing and they are fast to computer as well as they are responsible for converting the nodes summed and weighted input into the node's activation in the network. The 2d matrix will be given as the input to the convolutional layer. In the convolutional layer the data are filtered using three filters 32 bit, 64 bit and 64 bit filter with the size of 3x3. Each and every layer consist of maxpooling and with the dropout of 0.5 and 0.1. The flatter layer is used to convert the 2-D array from pooled feature into single continuous linear vector.(Hinz, Barros and Wermter, 2016)

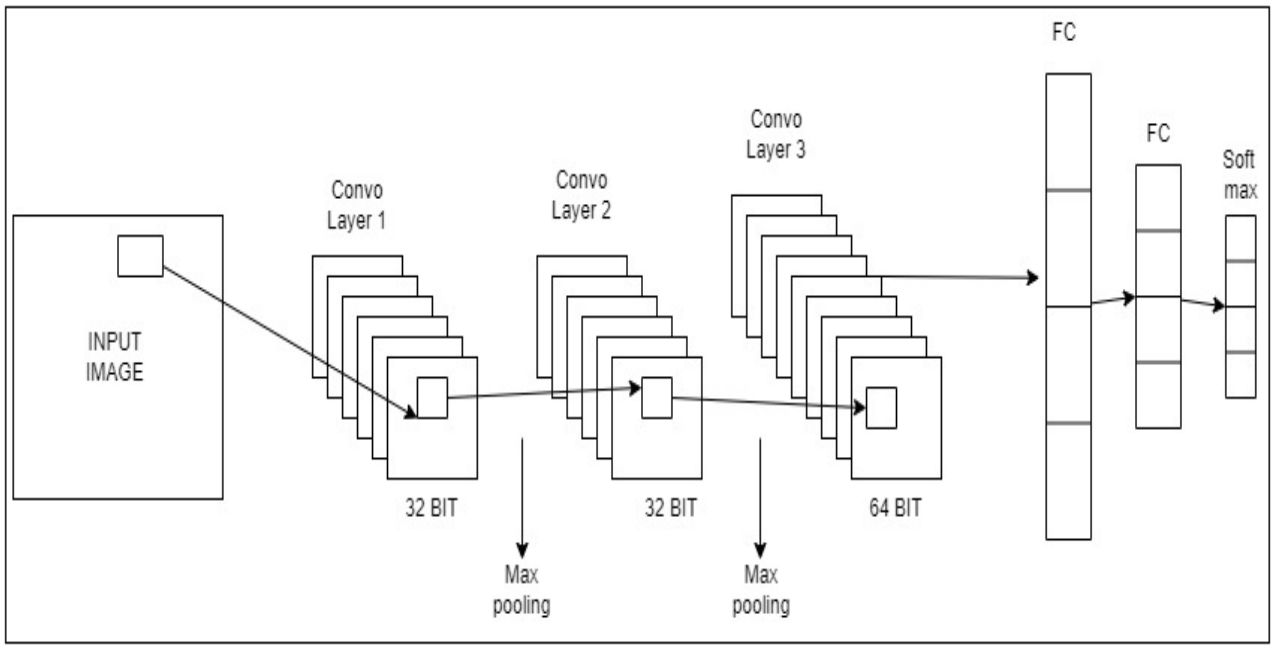


Fig 3: Working of filters

3.5 Model Testing:

In this phase, 50% of the dataset in fed into training and rest 50% is given to testing. Similar procedure is followed in testing phase and the result of the training part is loaded into it. After running the dataset through the training model, the accuracy of the single CNN will be retrieved.

In this paper, CNN, we will be able to identify the ransomware using the hashes. The CNN will be trained using these hashes and then it will be trained to detect the good ware and the ransomware with high accuracy and with faster detection rate.

4 Design specification:

The design flow are classified into four types in order to detect the ransomware and they are

Data Pre-processing: In this phase the IDS.CSV file obtained from RISS dataset. Python Framework and Juypter Notebook are used to process the dataset's features. The dataset is analysed, based on the number of rows and columns the dummy columns are added in order to form a perfect square matrix in order to improve the efficacy. Once the square matrix is formed 50% of the dataset are given to training and other 50% is given to testing. The output this stage will be in the form of convolutional matrix.

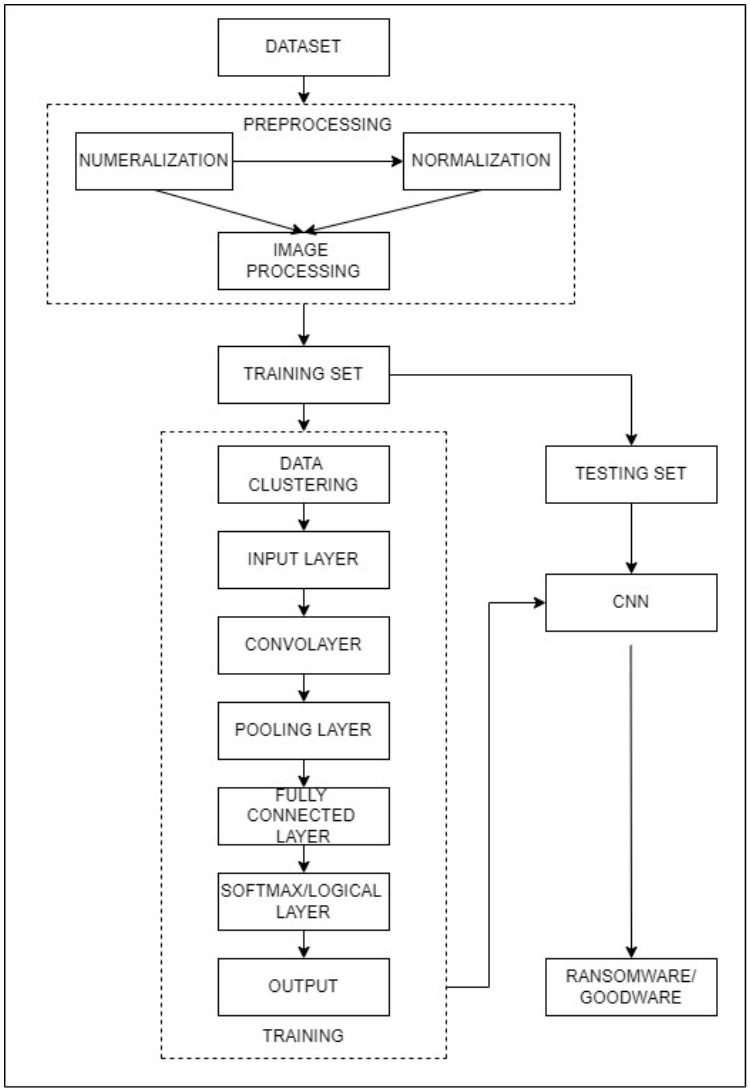


Fig 4: Working of CNN

Training: In this phase, the convolutional matrix will be converted into a greyscale image. Based on the classification, the images will be separated as anomalous and benign and stored in their respective folders. After this process, the images will be passed through the filter, such as the convolution layer pooling layer. The CNN is now ready for training. Based on the number of data points in the dataset, the value of epochs is calculated, and the batch size is varied based on the value of epoch. Based on the accuracy, epoch, loss, and model loss, the graph will be generated.

Testing: In this process, the same steps will be followed in terms of training. But once the image is formed in the testing, a "model" file will be created by CNN. This model file will be formed by CNN by understating the dataset in both the training and testing phases. With the help of this model, CNN will be able to identify the good ware and the ransomware.

Visualization: For ease of comprehension, the gathered results will be transformed into visual representations such as graphs and confusion matrices.

5 Implementation:

A Deep Neural Network is a term used to describe Artificial Neural Networks (ANN) with certain layers. It is considered the most effective tool in recent years, and it has the capability to handle huge amounts of data and has acquired widespread acceptance in the literature.

CNN have achieved ground-breaking results in a broad variety of pattern recognition domains, including image processing, as a consequence of improvements. CNN have the greatest benefit of reducing the number of parameters in ANN. This achievement has prompted researchers to employ larger models to solve difficult challenges.

5.1 Dataset preparation:

The RISS dataset is fed in to CNN which will converted into grey scale images. The null values are removed from the data set. The row “ransomware” is removed from the dataset and it is stored under the variable “X”. And the separated row (ransomware) will be stored under the variable “Y”. The values in the dataset will then be changed to binary values by creating new rows in order to convert the data into a matrix. And then the normalization is carried out in order to reduce the values.

The functions "train_test_split" and "Sklearn. Model_selection" are used to split the data into two halves. The column features 3025 rows of data, which is converted into a 55x55 matrix. The matrix will be based on the training and testing data.

5.2 CNN Training:

The images generated in training phase are grey scale images the resize will be 50x50 and these images are loaded to “X_train” and the labels are stored under “Y_train” only after the values are divided by 255. Since it is the highest possible value of the pixel.

CNN is started by importing “from tensorflow.keras.models import Sequential” now the image are passed through the filter which ranges from 32, 64 and 64 respectively and the size of the kernels 3x3.all these three layers consist of “Relu” and “maxpooling” with size 2x2. The data is passed through these layers and the matrix is retrivied in the final layer. And the final output is flatten using the ‘flatten()’ function.

A dropout layer and two thick layers are used to create the hidden layer. The CNN's final result is stored as anomaly and benign using the "softmax" function. This model is executed with the 10 epoch and batch size as 150 and processed with the help of “adam” optimizer in order to achieve maximum accuracy with minimal loss using “binary_crossentropy” function.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 46, 46, 32)        832

 max_pooling2d_4 (MaxPooling  (None, 23, 23, 32)       0
 2D)

 dropout_3 (Dropout)         (None, 23, 23, 32)        0

 conv2d_5 (Conv2D)           (None, 21, 21, 32)        9248

 activation_3 (Activation)   (None, 21, 21, 32)        0

 max_pooling2d_5 (MaxPooling  (None, 10, 10, 32)       0
 2D)

 batch_normalization_3 (Batc  (None, 10, 10, 32)       128
 hNormalization)

 conv2d_6 (Conv2D)           (None, 8, 8, 64)          18496

 activation_4 (Activation)   (None, 8, 8, 64)          0

 max_pooling2d_6 (MaxPooling  (None, 4, 4, 64)         0
 2D)

 batch_normalization_4 (Batc  (None, 4, 4, 64)         256
 hNormalization)

 dropout_4 (Dropout)         (None, 4, 4, 64)          0

 conv2d_7 (Conv2D)           (None, 2, 2, 64)          36928
```

```
activation_5 (Activation)    (None, 2, 2, 64)          0

max_pooling2d_7 (MaxPooling  (None, 1, 1, 64)          0
2D)

batch_normalization_5 (Batc  (None, 1, 1, 64)          256
hNormalization)

flatten_1 (Flatten)          (None, 64)                0

dense_3 (Dense)              (None, 280)               18200

dropout_5 (Dropout)          (None, 280)               0

dense_4 (Dense)              (None, 128)               35968

dense_5 (Dense)              (None, 2)                 258

=================================================================
Total params: 120,570
Trainable params: 120,250
Non-trainable params: 320
```

```
model = Sequential()
model.add(Conv2D(32,(5,5), input_shape=(50,50,1), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.5))

model.add(Conv2D(filters=32, kernel_size=(3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(BatchNormalization())

model.add(Conv2D(filters=64, kernel_size=(3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

model.add(Conv2D(filters=64, kernel_size=(3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(BatchNormalization())

model.add(Flatten())
model.add(Dense(units=280, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(128, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
```

Fig 5: Code snippet of filters

In the above fig 5 the CNN gets terminated once the flatten function is executed and the output is called by “model.h5”. And the above fig shows code snippet of the filters added to the snippet. The output is passed through all these three layers with a dropout of 0.5. “Softmax” contains the activation function of the output layer.

5.3 CNN Testing:

The images that need to be tested are loaded separately from the actual data to the given path and converted to images. The steps that followed are repeated in the testing. The “load model” function is used to call the trained function. And the inputs are passed to the “model_predict” function. The output of trained model was obtained with the accuracy of 97.04% .And the confusion matrix was obtained with accuracy, F1-score, recall. Precession.

5.4 Processing time:

It takes 10000 seconds to train the multi-CNN fusion model, which includes loading the image dataset and running 10 epochs with 150 batches. It takes 2 seconds to test a model.

6. Evaluation:

"Sklearn.metric" library with other few metric was used to calculate the evaluation.

Accuracy, Precision, and Recall are the three metrics used to verify the CNN model.

Accuracy in CNN:

In CNN, accuracy specifies how a model behaves in all classes. When all classes are comparable to one another. It is used to calculate the proportion of correct predictions to number of predictions.

$$\text{Accuracy} = \frac{\text{TRUE}_{\text{POSITIVE}} + \text{TRUE}_{\text{Negative}}}{\text{TRUE}_{\text{Positive}} + \text{TRUE}_{\text{Negative}} + \text{FALSE}_{\text{Positive}} + \text{FALSE}_{\text{Negative}}}$$

Precision calculation in CNN:

Using the precession, the accuracy of classified positive samples can be calculated. Precision is determined by the number of positive samples divided by the total number of positive samples (whether corrected or incorrectly classified).

$$\text{Precision} = \frac{\text{TRUE}_{\text{Positive}}}{\text{TRUE}_{\text{Positive}} + \text{FALSE}_{\text{Positive}}}$$

When CNN makes the classification as incorrect positive or only few positive there will be an increase in denominator to make precision small. The precision will be high only when more correct positive classification or less incorrect positive classification. Precision will pretend miscalculations of a negative sample as a positive sample and to make sure that all positive samples are positive.

Recall calculation in CNN:

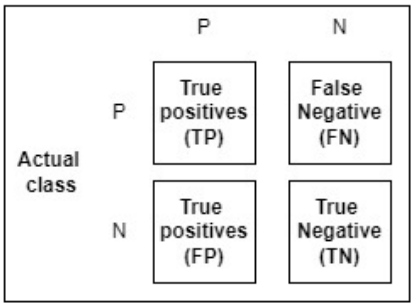
It is determined as the ratio of the number of properly identified positive samples to all positive samples.

$$\text{Recall} = \frac{\text{True}_{\text{positive}}}{\text{True}_{\text{positive}} + \text{False}_{\text{Negative}}}$$

F-Measure:

It represents the harmonic mean of recall and precision. (dshahid380, 2019)

$$\text{F-measure} = 2\frac{(presision*Recall)}{(presision+recall)}$$



Predicated Class	Actual Positive	Class Negative
Normal	0	517
Anomaly	1	850

The output of the CNN was 97.04% for the "RISS" dataset with a precision value of 0.40 for benign, whereas for anomaly it showed 0.00. And the value of recall was 1.00 for benign and the f1 score was 0.56 for benign with the support value of 119 for benign and support value for anomaly was 186

CNN	Precision	Recall	F1 score	support
0	0.36	1	0.56	119
1	0	0	0	186

```
from sklearn.metrics import accuracy_score

print(accuracy_score(y_test,y_pred))

0.9704918032786886
```

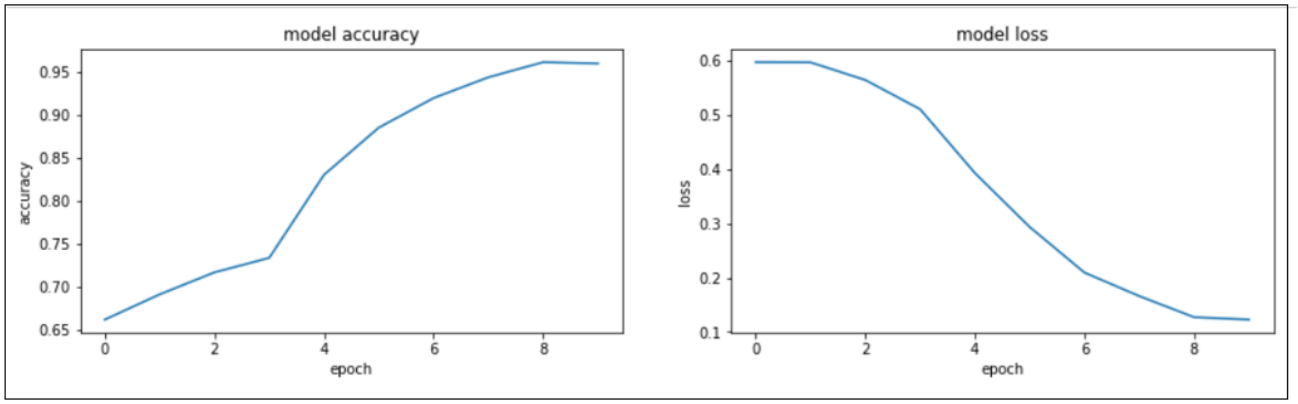


Fig 6: Accuracy graph

6.1 Discussion:

The ultimate goal of this paper is to detect ransomware. A "RISS dataset" was used, which was fed into CNN for detection. The MD5 and SHA1 signatures were used to detect the ransomware with faster results and high accuracy. Organizations have made significant efforts to strengthen their security over the years, but attackers have learnt how to spot them. Despite the fact that organisations spend millions of dollars on security, cybercriminals are continuously hunting for vulnerabilities to exploit in order to attack an organisation with ransomware. In order to overcome this, deep learning and machine learning techniques can be used in detection as proposed in this paper. Convolutional neural networks can be used in detection of ransomware, which can be used at early stage detection.

7. Conclusion:

In this paper, ransomware was detected using a signature-based method using convolutional neural networks. The detection was accurate to 97.04%. Many organisations are targeted by ransomware attacks, so the dataset used in this research consists of ransomware signatures that were commonly used by attackers to compromise the systems in an organisation. This method can be implemented in a real-time system for early detection in order to improve security.

Limitations: In this paper two commonly most hashes were used in detection the number of hashes can be increased.

Future works: This technique can be further improved by detecting the ransomware early stage by creating a sandbox environment using CNN using signature based detection.

Reference:

dshahid380 (2019) *Convolutional Neural Network. Learn Convolutional Neural Network from… | by dshahid380 | Towards Data Science*. Available at: https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529

73 Ransomware Statistics Vital for Security in 2022 - Panda Security Mediacenter (no date). Available at: https://www.pandasecurity.com/en/mediacenter/security/ransomware-statistics/ (Accessed: 14 August 2022).

Agrawal, R. *et al.* (2019) 'University of California , Santa Cruz , Santa Cruz , CA 95064 USA Microsoft Corp ., One Microsoft Way , Redmond , WA 98052 USA', pp. 3222–3226.

Aljubory, N. and Khammas, B. M. (2021) 'Hybrid Evolutionary Approach in Feature Vector for Ransomware Detection', *International Conference on Intelligent Technology, System and Service for Internet of Everything, ITSS-IoE 2021*. doi: 10.1109/ITSS-IoE53029.2021.9615344.

Almousa, M., Basavaraju, S. and Anwar, M. (2021) 'API-Based Ransomware Detection Using Machine Learning-Based Threat Detection Models', *2021 18th International Conference on Privacy, Security and Trust, PST 2021*. doi: 10.1109/PST52912.2021.9647816.

Alqahtani, A., Gazzan, M. and Sheldon, F. T. (2020) 'A proposed Crypto-Ransomware Early Detection(CRED) Model using an Integrated Deep Learning and Vector Space Model Approach', *2020 10th Annual Computing and Communication Workshop and Conference, CCWC 2020*, pp. 275–279. doi: 10.1109/CCWC47524.2020.9031182.

Ayub, M. A., Continella, A. and Siraj, A. (2020) 'An I/O Request Packet (IRP) Driven Effective Ransomware Detection Scheme using Artificial Neural Network', *Proceedings - 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science, IRI 2020*, pp. 319–324. doi: 10.1109/IRI49571.2020.00053.

Baek, S. *et al.* (2021) 'SSD-Assisted Ransomware Detection and Data Recovery Techniques', *IEEE Transactions on Computers*, 70(10), pp. 1762–1776. doi: 10.1109/TC.2020.3011214.

Bahrani, A. and Bidgly, A. J. (2019) 'Ransomware detection using process mining and classification algorithms', *Proceedings of 16th International ISC Conference on Information Security and Cryptology, ISCISC 2019*, (Iscisc), pp. 73–77. doi: 10.1109/ISCISC48546.2019.8985149.

bhatia, richa (2018) *Why Convolutional Neural Networks Are The Go-To Models In DL*. Available at: https://analyticsindiamag.com/why-convolutional-neural-networks-are-the-go-to-models-in-deep-learning/ (Accessed: 14 August 2022).

dshahid380 (2019) *Convolutional Neural Network. Learn Convolutional Neural Network from… | by dshahid380 | Towards Data Science*. Available at: https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529 (Accessed: 10 April 2022).

Hinz, T., Barros, P. and Wermter, S. (2016) 'The effects of regularization on learning facial expressions with convolutional neural networks', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9887 LNCS(September), pp. 80–87. doi: 10.1007/978-3-319-44781-0_10.

Ko, J. *et al.* (no date) 'Analyzed Android Applications', *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–5.

Kok, S. H. *et al.* (2019) 'Prevention of crypto-ransomware using a pre-encryption detection algorithm', *Computers*, 8(4), pp. 1–15. doi: 10.3390/computers8040079.

Lee, S. J. *et al.* (2022) 'Study on Systematic Ransomware Detection Techniques', *International Conference on Advanced Communication Technology, ICACT*, 2022-Febru, pp. 297–301. doi: 10.23919/ICACT53585.2022.9728909.

liska, allan and gallo, timothy (2316) *1. Introduction to Ransomware - Ransomware [Book]*. Available at: https://www.oreilly.com/library/view/ransomware/9781491967874/ch01.html (Accessed: 14 August 2022).

M, M. J. M. (no date) 'Gradient Tree Boosting', *Http://Scikitlearn.Org/Stable/Modules/Ensemble.Html#Gradi Ent-Boosting*.

Manavi, F. and Hamzeh, A. (2021) 'Static Detection of Ransomware Using LSTM Network and PE Header', *26th International Computer Conference, Computer Society of Iran, CSICC 2021*. doi: 10.1109/CSICC52343.2021.9420580.

Medhat, M. *et al.* (2020) 'YARAMON: A Memory-based Detection Framework for Ransomware Families', *2020 15th International Conference for Internet Technology and Secured Transactions, ICITST 2020*, pp. 10–15. doi: 10.23919/ICITST51030.2020.9351319.

Naik, N. *et al.* (2019) 'Lockout-Tagout Ransomware: A Detection Method for Ransomware using Fuzzy Hashing and Clustering', *2019 IEEE Symposium Series on Computational Intelligence, SSCI 2019*, pp. 641–648. doi: 10.1109/SSCI44817.2019.9003148.

Noorbehbahani, F. and Saberi, M. (2020) 'Ransomware Detection with Semi-Supervised Learning', *2020 10h International Conference on Computer and Knowledge Engineering, ICCKE 2020*, pp. 24–29. doi: 10.1109/ICCKE50421.2020.9303689.

Poudyal, S. and Dasgupta, D. (2020) 'AI-Powered Ransomware Detection Framework', *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020*, pp. 1154–1161. doi: 10.1109/SSCI47803.2020.9308387.

Ransomware Dataset – RISS (no date). Available at: https://rissgroup.org/ransomware-dataset/ (Accessed: 14 August 2022).

Salehi, S. *et al.* (2018) 'A Novel Approach for Detecting DGA-based Ransomwares', *2018 15th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology, ISCISC 2018*. doi: 10.1109/ISCISC.2018.8546941.

Shaukat, S. K. and Ribeiro, V. J. (2018) 'RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning', *2018 10th International Conference on Communication Systems and Networks, COMSNETS 2018*, 2018-Janua, pp. 356–363. doi: 10.1109/COMSNETS.2018.8328219.

Sheen, S. and Yadav, A. (2018) 'Ransomware detection by mining API call usage', *2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2018*, pp. 983–987. doi: 10.1109/ICACCI.2018.8554938.

Sibi Chakkaravarthy, S. *et al.* (2020) 'Design of intrusion detection honeypot using social leopard algorithm to detect IoT ransomware attacks', *IEEE Access*, 8, pp. 169944–169956. doi: 10.1109/ACCESS.2020.3023764.

Silva, J. A. H. and Hernández-alvarez, M. (2017) 'Security', pp. 8–11.

Urooj, U., Aizaini Bin Maarof, M. and Ali Saleh Al-Rimy, B. (2021) 'A proposed Adaptive Pre-Encryption Crypto-Ransomware Early Detection Model', *2021 3rd International Cyber Resilience Conference, CRC 2021*. doi: 10.1109/CRC50527.2021.9392548.

Why convolutional neural network is better? - Intellipaat Community (no date). Available at: https://intellipaat.com/community/46830/why-convolutional-neural-network-is-better (Accessed: 14 August 2022).

Zhang, X., Wang, J. and Zhu, S. (2022) 'Dual Generative Adversarial Networks Based Unknown Encryption Ransomware Attack Detection', *IEEE Access*, 10, pp. 900–913. doi: 10.1109/ACCESS.2021.3128024.