STUDY ON ARTIFICIAL INTELLIGENCE-BASED RANSOMWARE DETECTION FOR DIGITAL SUBSTATIONS

A Thesis

by

SYED RAQUEED BIN ALVEE

Submitted to the College of Graduate Studies
Texas A&M University-Kingsville
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2022

Major Subject: Electrical Engineering

STUDY ON ARTIFICIAL INTELLIGENCE-BASED RANSOMWARE DETECTION FOR DIGITAL SUBSTATIONS

A Thesis

by

SYED RAQUEED BIN ALVEE

Approved as to style and content by:

Dr. Taesic Kim

Taesic Kim, Ph.D.
Committee Chair

Sung-won Park, Ph.D.
Committee Member

Avdesh Mishra

Avdesh Mishra, Ph.D.
Committee Member

Scott C. Smith

Scott Smith, Ph.D.
DepartmentChair

Robert J. Diersing

Robert J. Diersing, Ph.D.
Interim Vice President for Research
and Graduate Studies

May 2022

ABSTRACT

Study on Artificial Intelligence-based Ransomware Detection for Digital Substations

May 2022

Syed Raqueed Bin Alvee

Bachelor of Science, American International University – Bangladesh

Chair of Advisory Committee: Dr. Taesic Kim

Ransomware is a modern form of malware that prevents victims from accessing their computer systems, important document files/folders, etc. The attacker would not release them until a ransom was paid through some secret channels. Ransomware has become a serious threat to the current computing world, requiring immediate attention to prevent it. Ransomware attacks can also have disruptive impacts on operation of smart grids including digital substations. This thesis research proposes a ransomware attack modeling method targeting disruptive operation of a digital substation and investigates an artificial intelligence (AI)-based ransomware detection approach. The proposed ransomware file detection model is designed by a convolutional neural network (CNN) using 2-D grayscale image files converted from binary files. The experimental results show that the proposed method achieves 96.22% ransomware detection accuracy, which is better than the other methods reported in the literature, including the RF method based on N-gram of opcodes, with an accuracy of 91.43%, which was best among its peers.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Taesic Kim and my committee members, Dr. Sung-won Park and Dr. Avdesh Mishra for their guidance and support throughout the course of this research. I am grateful to Dr. Taesic Kim for mentoring me both as a thesis student and graduate research assistant.

Thanks to all my friends and colleagues and the department faculty and staff for making my time at Texas A&M University-Kingsville a great experience. I would also like to thank all of my colleagues from the Cyber-Physical Power and Energy Systems (CPPES) Laboratory that helped me with my research. Finally, thanks to my mother and father for their encouragement and for their patience, love and encouragement in the pursuit of my MSEE degree.

Financial support from the following institutions/organizations is gratefully acknowledged:

- Korea Electrotechnology Research Institute (KERI)
- The National Science Foundation (NSF) under award No. EEC-1359414
- The Department of Energy (DOE) under award No. DE-EE0009026

NOMENCLATURE

CKC	Cyber Kill Chain
CNN	Convolutional Neural Network
GRU	Gated Recurrent Unit
ICS	Industrial Control System
ICT	Information and Communications Technology
IED	Intelligent Electronic Device
KNN	K-Nearest Neighbor
NB	Naïve Bayes
RF	Random Forest
SCADA	Supervisory control and data acquisition
SDN	Software Defined Network
SPP	Spatial Pyramid Pooling
SVM	Support Vector Machine

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER 1. INTRODUCTION

Ransomware is a malware which usually encrypts the important or credential data and theft of control of a system in demand of a ransom. Ransomware is mostly propagated through user-initiated actions, such as falling for phishing emails that contain malicious attachments or from users unknowingly visiting an infected website, in which case the malware is downloaded and installed without the user's knowledge. There are hundreds of ransomware variants, all of which work slightly differently. However, once the payload is executed on the target machine, one of the first actions taken is the encryption of the files on the hard drive. The ransomware attacker then delivers a ransom note demanding payment in exchange for the decryption key of the victim's files.

When run on the system, the ransomware can lock the computer screen or, in the case of cryptographic ransomware, encrypt the specified files. The first scenario displays a full screen or notification on the screen of the infected system, making the system unavailable to the victim. This notification also includes instructions on how the user will pay the ransom. In the second scenario, ransomware prevents access to potentially important or valuable files such as documents and spreadsheets.

Ransomware is the most prevalent cyber threat since 2005. According to the Cybersecurity Insiders 2020 Malware Report, 43% of cybersecurity experts surveyed will experience an attack, and 80% of respondents are at least likely to experience another attack within the next year. In addition, 82% of respondents were most concerned about ransomware of all types of malwares.

1.1 Background

In 2011, WinLock emerged as the first locker ransomware, a variant that completely locks victims out of their devices. The malware infected users' systems through malicious websites. Launched in 2012, Reveton was the first Ransomware as a Service (RaaS). This is a rental service provides ransomware on the dark web to cybercriminals who have limited technical skills. Starting with Reveton, the public has the potential to threaten victims with ransomware. Reveton displayed a fraudulent law enforcement message accusing the victim of committing a crime. The attacker threatened the victim on prison terms if the victim did not pay the ransom. A ransomware attack was discovered early in September 2013 [1]. The attack initiated by sending emails with an attachment that contains CyrptoLocker ransomware (i.e., an originally executable file (.exe), but disguised as a normal PDF file). The executed ransomware encrypted certain types of files on local hard drives and network drives using Rivest–Shamir–Adleman (RSA) public key cryptography, while the malware control servers stored the private key which is only provided if a payment is made. It is also observed that the ransomware file was able to be propagated using Gameover Zeus trojan and botnet as well [1]. Ransomware programs might be detected by legitimate security professionals, and they provide a master code to decrypt the system. However, finding a solution without paying the ransom is still challenging. In October 2015, FBI agents stated that by default, victims should pay the ransom only if the system is locked by ransomware. In April 2015, it was reported that many police stations were forced to pay ransom to computer criminals. The ransomware attacks became a global concern after more than 1,400,000 Kaspersky users were attacked across various sectors in 2016 [2]. In 2017, about 400,000 machines in 150 countries were infected by the WannaCry ransomware [3].

Therefore, many security researchers in information and communications technology (ICT) domains have paid special attention to ransomware detection in recent years.

Ransomware assaults on industry control systems (ICS) have increased by around 500 percent between 2018 and 2020 [4]. The Colonial Pipeline was hit by ransomware in 2021, which crippled the pipeline's digital machinery. For the decryption tool, the corporation had to pay 4.4 million dollars in Bitcoin [5]. It is anticipated that ransomware attackers would increasingly target vital power infrastructures like substations and wind/solar farms. As a result, it is critical to detect and prevent ransomware attacks as soon as possible.

1.2 History of Malware Attacks in Power Grids

BlackEnergy:

In 2014, ICS-CERT issued a series of notifications outlining a sophisticated malware campaign that used a variation of the BlackEnergy malware to compromise several ICSs. According to DHS research, this campaign has continued since at least 2011 [6]. GRIZZLEY STEPPE [7], a 2016 DHS and FBI Joint Analysis Report that identified Havex as a member of the RIS group, also linked BlackEnergy to them.

The effort targeted human machine interface (HMI) solutions from various ICS suppliers, including GE Cimplicity, Advantech/Broadwin WebAccess, and Siemens WinCC. Because the malware is modular, not all features are available to all victims. Modules that look for any network-connected file shares and removable devices that could aid the virus in lateral movement inside the affected environment have been found in typical BlackEnergy infections [6].

DHS confirmed in December 2014 that a BlackEnergy three malware strain was present in an attack on a Ukraine energy infrastructure that resulted in a power outage. To the DHS

secure portal website, ICS-CERT released a special TLP Amber version of a warning with extra information about the malware, plug-ins, and indicators. Asset owners and operators are highly encouraged to use the indications to look for symptoms of a breach within their control system environments, according to ICS-CERT.

BlackEnergy, like Havex, targeted key ICS including power grids. When enemies target critical infrastructure control systems, it is a cause for concern. We learn about nation-state threat actors and their tools to target critical power infrastructure by BlackEnergy.

Ukraine Power Grid:

A cyber-attack disrupted energy to roughly a quarter-million Ukrainians two days before Christmas in 2015. This was the first time of a successful cyber-attack on a power grid.

According to Reuters, a power provider in the western part of Ukraine experienced a power outage that affected a vast area, including the regional capital of Ivano-Frankivsk [7]. Attackers knocked down power at 30 substations, stranding 230,000 people for up to six hours. Supervisory control and data acquisition (SCADA) equipment were rendered unusable, and power restoration had to be done manually, which added to the time it took to restore power [8].

Investigators revealed that attackers aided the outage by exploiting macros in Microsoft Excel documents with the BlackEnergy malware. Spear-phishing emails were used to infect the company's network with malware [9]. The malware was analyzed by ICS-CERT and US-CERT in collaboration with the Ukrainian CERT and international partners. A BlackEnergy 3 variant was determined to be present in the Ukrainian power system [6]. The attack was blamed on Russian attackers by the Ukrainian intelligence community [10].

A US interagency team consisting of representatives from ICS-CERT and US-CERT, as well as DOE, the FBI, and the North American Electric Reliability Corporation, visited Ukraine

at the request of the Ukrainian government to gather information about the incident and identify potential mitigations [11].

This incident demonstrated to the world that a cyber-attack could destroy the power grid, and it served as a wake-up call to guarantee that the US power grid might be fortified against such attacks. In the instance of Ukraine, the attackers utilized crude, technically inept methods to accomplish their goal [12]. The cyber-attack on Ukraine's power infrastructure was a watershed moment in cyber history.

Second attack on Ukraine's Power Grid targeting Substations:

Kyiv fell dark again on December 17, 2016, almost a year after Ukraine's power grid was hit by a catastrophic cyber-attack. Monitoring stations were rendered blind as a result of cyber-attacks. Breakers at 30 substations tripped, knocking out power to nearly 225,000 people. To prolong the outage, attackers conducted a telephone denial-of-service (TDoS) attack against the utility's call center, similar to the strategy used in 2015. On their way out, the intruders rendered devices such as serial-to-Ethernet converters dysfunctional and unrecoverable, making it more difficult to restore power to clients [13]. Despite these losses in the initial attack, electricity was restored in most cases in three hours. However, workers had to travel to substations and manually close breakers that the attackers had remotely opened [14], [15] because the attackers had destroyed management systems. On the other hand, the second attack was far more sophisticated than the first [14]. The second attack is thought to have utilized sophisticated malware to directly manipulate SCADA systems, whereas the first used remote-control software to manually trip breakers.

On October 24, 2016, there were mass attacks with ransomware called Bad Rabbit [16]. It has been targeting organizations and consumers, mostly in Russia but there have also been reports of victims in Ukraine. Figure 1 shows how the ransom message was displayed on encrypted devices. This ransomware infected the devices through multiple hacked Russian media websites.

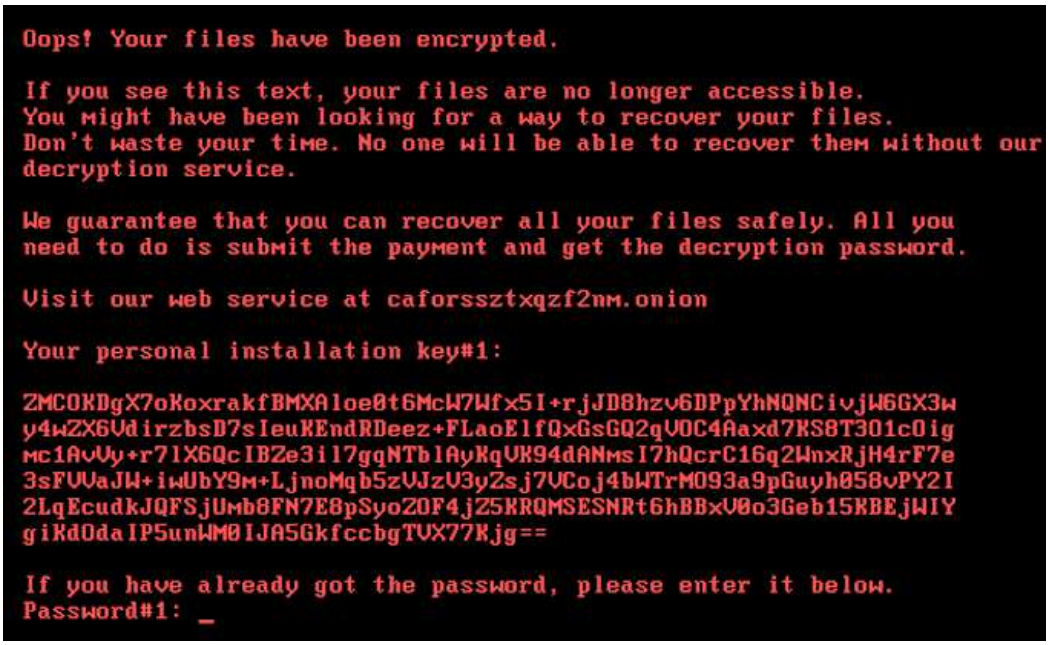


Figure 1. Attack in Ukraine by the Ransomware- BadRabbit. [16]

Triton/ Trisis/ HatMan:

At the end of 2017, FireEye released a report on TRITON, a new ICS attack framework to disrupt critical infrastructure operations. TRITON [13], according to FireEye, was targeting industrial safety systems in the Middle East. In late 2017, Symantec produced another report on the same malware, dubbed it Trisis [13]. Meanwhile, the Department of Homeland Security ICS-CERT published a Virus Analysis Report on the same malware in December 2017 but referred to it by a third moniker, HatMan [14].

The malware targets Schneider Electric's Triconex safety instrumented system by modifying in-memory firmware to add malicious functionality that allows an attacker to read/modify memory contents and execute custom code on demand by receiving specially crafted network packets from the attackers [14], as well as additional programming to disable, inhibit, or modify the ability of a process to fail safely. This malware can be physically dangerous since it targets safety systems of power grid equipment [13].

The most alarming element of TRITON is that it is the first malware known to attack industrial safety systems designed to protect human life. Other attackers may utilize this attack to cause actual hurt or damage to people.

1.3 Problem Statement

Digitalization of substations and the use of Intelligent Electronics Devices (IED)'s have increased the surface for different types of cyber-attacks. Since ransomware attacks are more widely seen in the industrial sector now-a-days, digital sub-stations are at a risk of facing ransomware attacks and face financial damages as well as loose sensitive data to the future attackers.

1.4 Objective

In this thesis, a ransomware detection method is presented based on a deep learning algorithm. The goal of this thesis project is to explore ransomware attacks in a digital substation and investigates an artificial intelligence (AI)-based ransomware detection method. A cyber kill chain (CKC)-based ransomware attack modeling method is designed, which targets disruptive operation of a digital substation. A convolutional neural network (CNN) model is designed and trained using 2-dimensional (2-D) grayscale image files from real ransomware binary files.

CHAPTER 2. LITERATURE REVIEW

In general, ransomware detection methods are classified into two categories: Static analysis and dynamic analysis methods. Static malware analysis examines ransomware without executing the actual binary files. Simple static analysis methods utilize static data such as file header information, file hash, and URL. There are open-source tools/servers providing static malware analysis such as VirusTotal [17]. Although conventional static malware analysis methods are simple to detect known malware and easy to implement [18], they are largely ineffective against sophisticated ransomware attacks [19]. Recently, static malware detection methods using artificial intelligence (AI) have been proposed to improve detection accuracy [20]. Since ransomware is evolving, new malware should be detected as well.

Dynamic analysis methods detect ransomware attacks using abnormal behavioral data caused by the compiled ransomware or ransomware events by adversaries in the target system. The authors in [21] collected packets and data from network traffic between an infected computer and a command and control (C2) server. Using the network data, a random forest (RF) machine learning (ML) method detected ransomware with over 86% of detection accuracy. In [21], an ML combining Navies Bayes (NB) and Support Vector Machine (SVM) is used to detect ransomware attacks by using network data from function virtualization (NFV) and software-defined network (SDN). The authors claim that this approach could achieve a 99.99% detection rate. Compared to static malware analysis methods, dynamic methods might provide a better capability for detecting sophisticated and unknown ransomware. However, a huge amount of network and event data is required.

Deep learning algorithms can learn data representations with various degrees of abstraction automatically [22]. This can aid in the production of very accurate forecasts. As a

result, we find them used successfully in a variety of domains, including image and audio recognition, natural language processing, and recommender systems [23], [24], [25]. Deep learning algorithms have been proposed to use in the field of malware detection as well. This section discusses some of the works that have employed these various deep learning techniques for malware detection.

CNNs can extract local features from image data. Therefore, they are widely used for image classification and face recognition. However, their use in malware detection is hampered by the lack of methods to represent malicious files as images. Tobiyama et al. [26] analyzed the sequence of API calls made by malware and then turned them into malware images for detection using a CNN approach. However, this method is ineffective, and it takes a long time to identify malware binaries. He et al. [27] suggested a malware detection approach based on picture recognition. They used the CNN model and the Spatial Pyramid Pooling (SPP) layer to transform malware files into RGB visual representations and identify them. Grayscale imaging is immune to redundant API injection attacks, even though the SPP implementation is not viable owing to large memory needs. From Dalvik bytecode, Xiao [28] utilizes CNNs to understand the features of Android malware. The authors claimed that their approach is extremely efficient with a 93% accuracy rate. However, the method has trouble coping with different-sized malware files and converting them to the same size picture for classification because these algorithms are only partially supervised.

In, Nataraj et al [29]. introduced the technique of converting binary executable files into grayscale images in bytes. Their findings revealed that malware images from the same family had comparable structural texture traits, whereas malware images from different families had quite diverse features. To classify malware samples, the k-nearest neighbor (KNN) technique

was used to compare the Gist texture properties of the malware grayscale image. The advent of malware visualization was a watershed moment in homology determination and following research has improved the process significantly. To finish the family categorization, malware files are converted to binary grayscale images, and texture features are extracted using image processing technologies. This approach is easy to implementation and provides low processing complexity, and excellent classification accuracy compared to other traditional static and dynamic methods. It has the same classification accuracy as dynamic analysis but takes 95% less time to perform.

The entropy diagram was first introduced in [30]. This method compares the entropy value of image rather than extracting texture information from the malware grayscale image. This method achieves equal classification ability and greatly improves efficiency when compared to the Nataraj method. Some researchers have utilized deep learning to categorize malware photos due to the rapid growth of the technology in recent years. Cui et al. [31] utilized a CNN to categorize grayscale malware images using a bat algorithm to balance the number of families. To categorize malware images, Venkatraman et al. suggested a hybrid architectural solution based on CNN and gated recurrent unit (GRU) [32]. Yuan et al. used bytes transfer probability matrices to convert malware binaries into Markov images [33]. The deep CNN is then used to classify Markov images. Ghouti and Imam [34] categorized malware files based on typical digital images using basic algebraic dot products and support vector machines. In [35], a novel deep learning-based architecture is proposed which can classify malware variants based on a hybrid model. The main contribution of the study is to propose a new hybrid architecture that integrates two wide-ranging pre-trained network models in an optimized manner. The method was tested on Malimg, Microsoft BIG 2015, and Malevis datasets. This architecture consists of

four main stages: 1) data acquisition, 2) designing deep neural network architecture, 3) training deep neural network architecture, and 4) evaluation of the trained deep neural network as shown in Figure 2 This approach includes several exhaustive pre-trained networks which rely upon the transfer learning method. These methods demonstrated the effectiveness of deep learning in the classification of malware images.

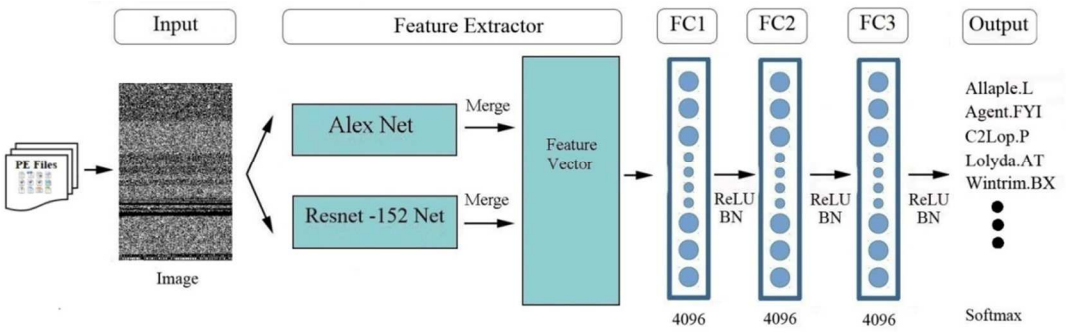


Figure 2. A deep learning-based malware classification method [35].

CHAPTER 3. PROPOSED CNN-BASED RANSOMWARE ATTACK DETECTION METHOD

3.1 Ransomware Attack Vector

This thesis assumes that an attacker's goal is to encrypt the substation control and diagnosis unit system by ransomware. Figure 3 introduces three potential ransomware attack vectors, targeting to disrupt a substation control and diagnosis unit operation in a local substation control room. Attack vector 1 is an external network attack path initiated from the platform information technology (PIT) involving vendor access servers, diagnosis centers, control centers, and other remote access points. Attack vector 2 is a local network attack route started from the internal substation in the operational technology (OT). Attack vector 3 is a physical intrusion. An intrusion detection system (IDS)-activated demilitarized zone (DMZ) is established between the PIT and OT networks. The IDS implements a deep packet inspection and ransomware detection programs against all incoming ransomware from the PIT network.

The CKC model is an attack modeling method that describes the chain of a cyber threat actor's actions in terms of attack tactics, techniques, and procedures. The latest substation related CKC version is MITRE's ATT&CK for ICS framework [36] which enumerates the actions of a cyber adversary might occur with an ICS environment. This thesis design attack models on a digital substation based on the MITRE's ATT&CK for ICS framework.

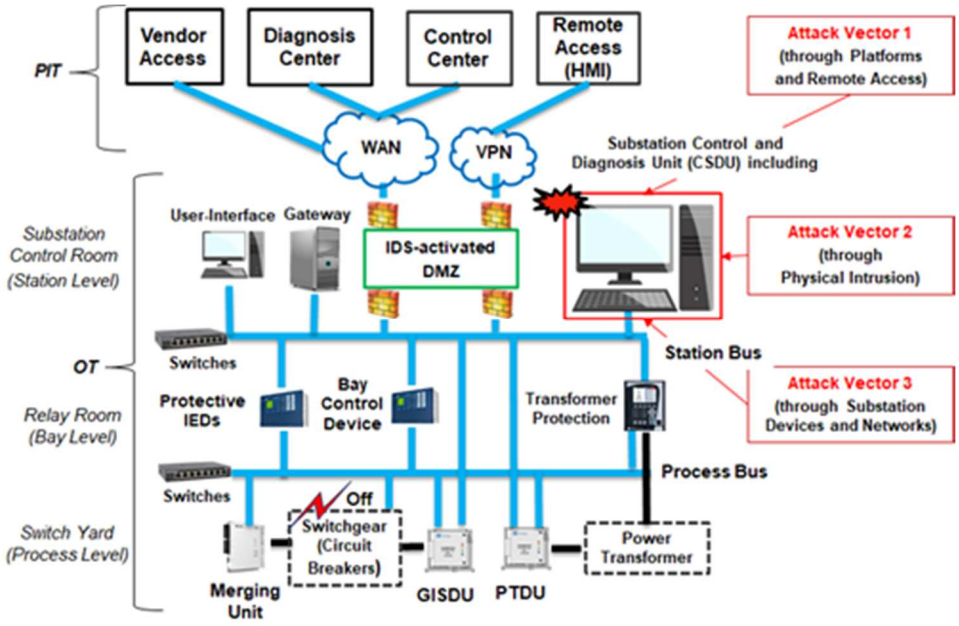


Figure 3. Ransomware attack vectors targeting a local server in a digital substation.

Figure 4 shows a CKC ransomware attack model for a digital substation having twelve ransomware attack phases. The attack scenario has been created by referring the Colonial Pipeline ransomware attack incident reports [37]. An advanced persistent threat (APT) actor is accessed a PIT system (e.g., a control/diagnosis center server) by social engineering (e.g., phishing) or exploiting remote access accounts leaked in the dark web (1. Initial Access). A backdoor malware is established then executed in the system (2. Execution). The adversary is continuously maintaining a foothold using a valid remote desktop protocol (3. Persistence). The malicious cyber actor is manipulated an access token to get ownership of a malicious running process (4. Privilege Escalation). With the previous technique, the attacker is masqueraded himself as a high privilege user to avoid a detection system (5. Evasion). Afterward, the substation information is gathered from the PIT (6. Discovery). The APT actor can access the on-site SDU system with the field network authority, including all connected local devices (7. Lateral Movement). The APT collects field device data to learn the operation of the target

substation. At this phase, malicious behaviors might be conducted over The Onion Router (TOR) and Cobalt Strike for anonymous communication (8. Collection). The ransomware file is loaded to the SDU system (9. Command and Control). By encrypting the response function, control process, and security alarm-related programs, the SDU controller is disabled to use (10. Inhibit Response Function and 11. Impair Process Control). Finally, the APT group demands the substation operator pay a ransom with an additional threat to disclose the collected substation data and system weakness information to the dark web (the double extortion ransomware attack). Finally, the substation can be shut down during the ransom negotiation period due to the encrypted software.

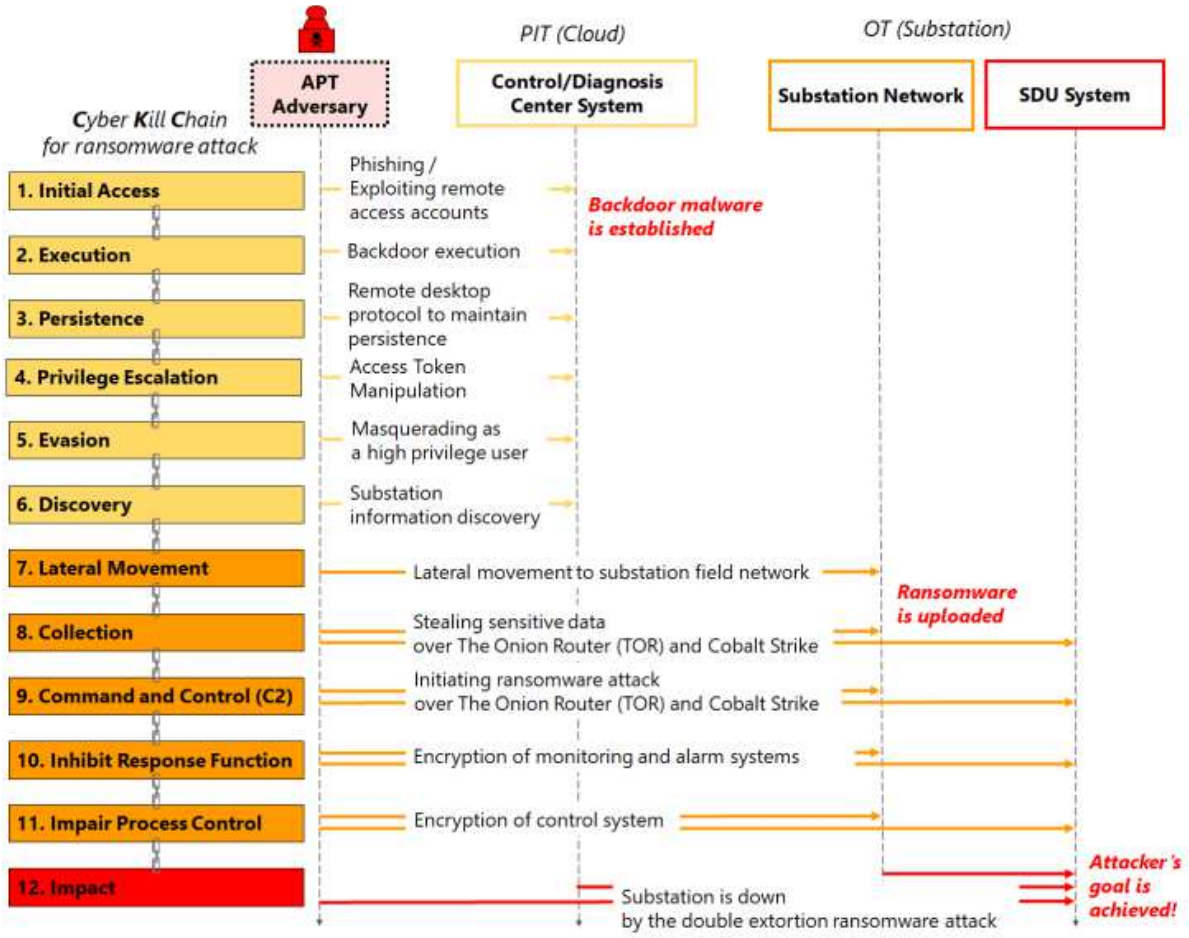


Figure 4. A cyber kill chain model for a substation ransomware attack.

3.2 Dataset

The data preprocessing of files to be seen as 2-D picture format is shown in Figure 5. By reading unsigned 8-bit integers, an executable ransomware or goodware file (i.e., binary file format) is first converted to a vector form. The size of the binary file is then used to generate a 2-D array [38].

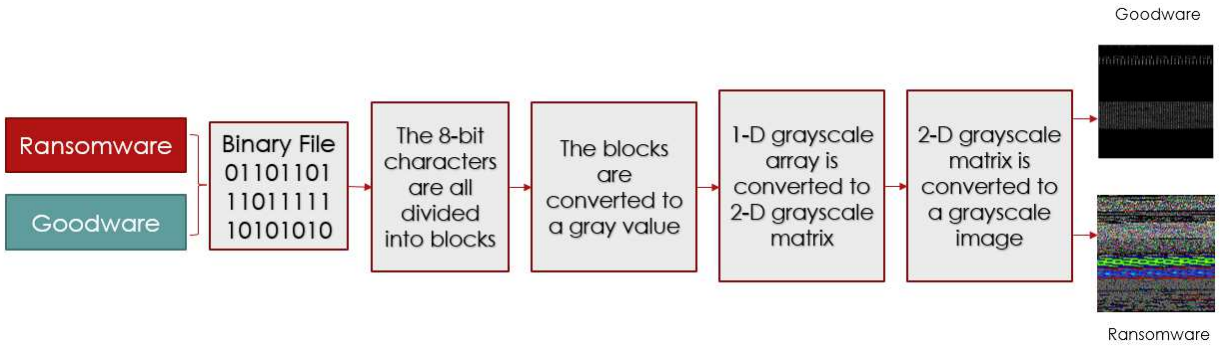


Figure 5. Data preprocessing.

Table 1 lists the appropriate image widths based on file size. Following the formation of the matrix, each value is assigned in grayscale hues ranging from 0 to 255. (255: white, 0: black). The 1-D grayscale array is then turned to a 2-D grayscale matrix, which is ultimately converted to a 2-D grayscale image.

Table 1: Image Width According to Various File Size

File Size Range	Image Width
<10kB	32
10 kB – 30 kB	64
30 kB – 60 kB	128
100 kB – 200 kB	256
200 kB – 500 kB	384
500 kB – 1000 kB	512
>1000 kB	1024

In computer vision and image classification tasks, unbalanced datasets are a typical issue. Underfitting and overfitting can occur due to a lack of pictures in each layer, which has a significant impact on CNN performance. As a result, the ransomware dataset includes a data augmentation strategy for improving the classifier's performance. Data augmentation is a technique for improving the quality of a dataset that is widely used to train neural networks. Fresh data are generated in the enhancement phase from classes with lower population in the datasets. To avoid unequal representation, this technique overcomes the limited impact on the data. Common preprocessing techniques such as rescaling and sample-wise standardization are also used to boost attack detection rates within a limited number of ransomware samples.

Researchers and practitioners can gain a better understanding of ransomware by viewing the binaries as images, as the patterns inside such images become more obvious. Deep learning is capable of detecting patterns inside such images. Ransomware and malware families can also be identified using the most essential patterns of features in malware images. Images for a certain family share similar pattern, allowing a deep learning network to discover relevant patterns by extracting features automatically. CNN models are particularly good at categorizing images because they can extract significant features inside an image by subsampling, pooling, and other computations. For the aim of classification, CNNs hunt for the most important elements inside an image from a certain malware family. A technique that turns a binary portable executable (PE) file into a sequence of 8-bit vectors or hexadecimal values can be used to convert malware binaries into images. In the range 00000000 (0) to 11111111 (255), an 8-bit vector can be expressed. Each 8-bit vector represents a number, and can be converted into pixel in an image, as shown in Figure 6. Each segment of the grayscale image represents a specific portion of binary file. The code segment contains the code which executes the file. The

full black parts indicate the zero padding segments. The values computed during the convolutions on the input channel are stored in this output channel as a matrix of pixels. If zero-padding = 1, the original image with pixel value = 0 will be one pixel thick. The data part consists of the uninitialized data and the initialized data.

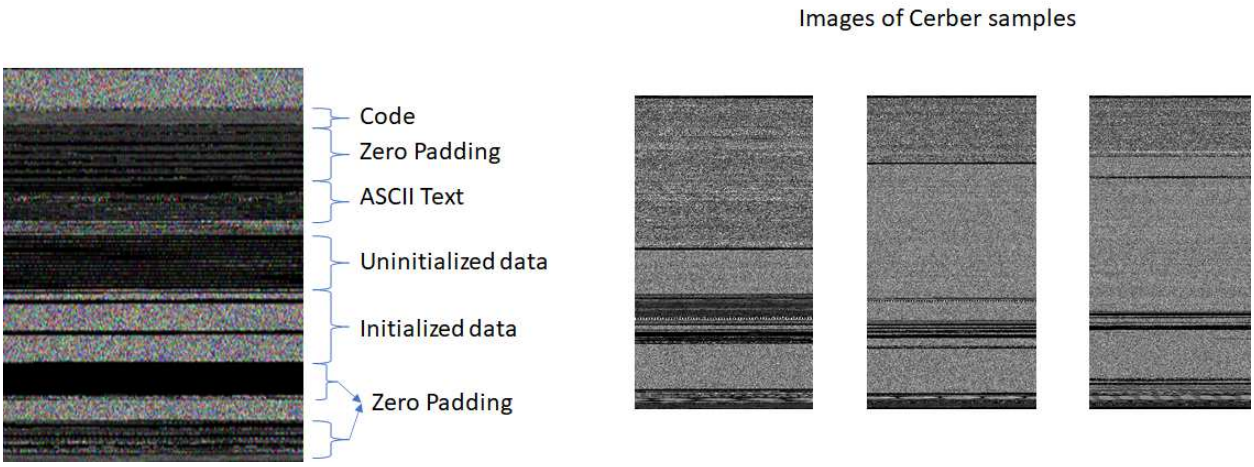


Figure 6. Ransomware Sample (Cerber).

3.3 Methodology

The proposed deep learning-based ransomware attack detection approach is described in this section. Figure 7 shows the proposed deep learning approach for detecting ransomware files created by a CNN model utilizing a gray-scale image as an input. Data pre-processing, feature extraction, and classification are the three sequential steps in designing the ransomware detection method. To prevent the dangerous payload of malware files from reaching the digital substation, the proposed AI model can be deployed in the IDS in Figure 3.

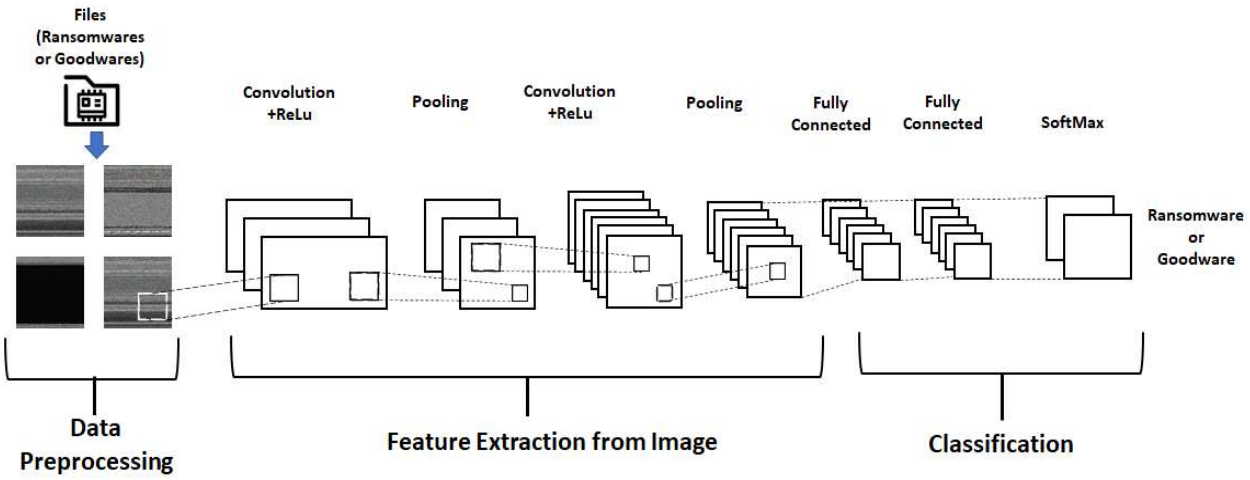


Figure 7. Proposed CNN-based ransomware detection method

The CNN model architecture has a multi-layered structure consisting of two convolutional layers (CLs), two Max pooling layers (MPLs), two fully connected layers, and Softmax. CLs and PLs are used to extract multiple features from the preprocessed inputs. ReLu is chosen as the activation function as it does not change the size of the image.

Two fully connected layers utilize the output from the convolution process and predicts the class of the image based on the features extracted in previous stages. Each neuron is processed by a point element between small regions and weights related to the amount of information. Softmax is used in the layer of CNN which normalizes the CNN output between 1 (i.e., ransomware) and 0 (i.e., goodware). User-configurable hyper-parameters including learning rate, number of hidden layers, number of hidden nodes, number of epochs, stack size, and type of activation function are optimally chosen by trial-and-error effort.

A convolution layer converts an input into a stack of feature mappings of that input in general. We may or may not reduce the size of the input depending on how we configure our padding. The number of filters defined for a layer determines the depth of the feature map stack.

The width and height of each filter will be defined, but the filter depth should be the same as the input. For Grayscale images our filter had a depth of 1 but for a colored image, we would need a filter with depth of 3, which contains filters for each of the RGB color channels. Instead of manually implementing the feature extractor, CNN leverages it in the training phase. The CNN feature extractor is made up of many types of neural networks that determine the weights during the training process. CNN is a neural network that extracts input visual features and classifies them using another neural network. Figure 8 shows the feature extraction network using the input image, and the neural network uses the extracted feature signals to classify the data. The result is subsequently produced by the neural network classification, which is based on the image features. Convolution layer piles and sets of pooling layers are used in the feature extraction neural network. The pooling layer combines the pixels of adjacent pixels into a single pixel. The image dimension is subsequently reduced by the pooling layer. The convolution and pooling layers are essentially in a two-dimensional plane.

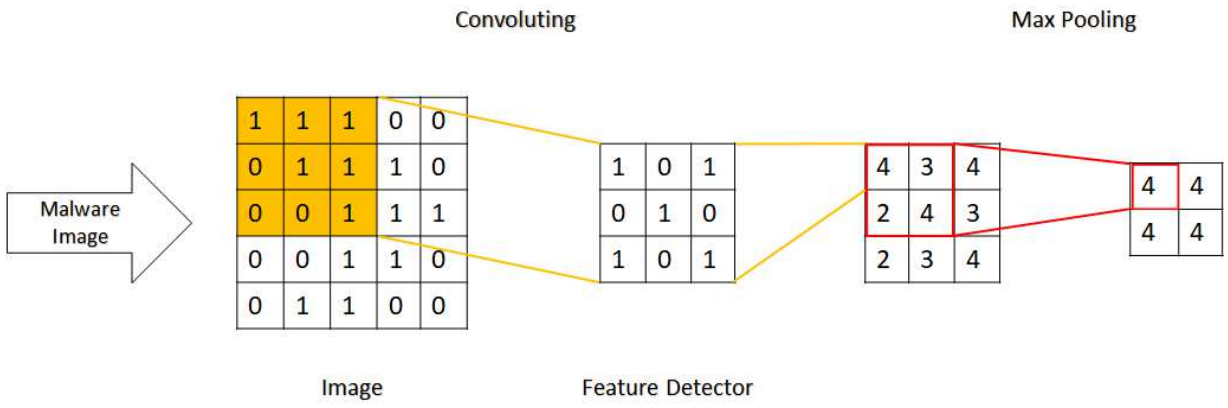


Figure 8. Feature Extraction in the CNN Model.

3.4 Hyper-Parameter Fine-Tuning

The process of modifying the classification part of the selected CNN for our use case is known as fine-tuning the model. Figure 5 depicts a typical CNN divided into input, feature extractor, and classifier components.

The model hyper-parameters are tuned by tweaking the classification part of the model (Figure 6) and added a GlobalAveragePooling2D pooling layer to reduce the output (dimensions) from the CNN. For the classifier part additional feature extraction was added by adding a fully connected or dense layer with 1,024 neurons and a ReLu activation function. This neuron size is the output size of our model's feature extraction pooling layer. Also, for class prediction, a fully connected or dense output layer of 8 neurons with a Softmax activation function is used.

Model complexity, learning capacity, and the rate of convergence for model parameters are all defined by hyper-parameters. Thus, finding the optimal value for hyper-parameters leads to improved efficiency and results. The learning hyper-parameter is one of the parameters that a user can configure. The learning rate, number of hidden layers, number of hidden nodes, number of epochs, batch size, and type of activation functions are among the hyper-parameters that a user can configure. The learning rate, number of hidden layers, number of dense nodes, and batch size are the hyper-parameters considered in this study.

- **Number of neurons**: The number of dense neurons in each dense layer was chosen as the first hyper-parameter. When using a neural network, selecting the number of hidden neurons is critical because it can lead to over-fitting or under-fitting. Over-fitting occurs when the user selects too many hidden neurons. Over-fitting occurs when a network trains itself so closely to the data that it loses its ability to predict on new data. Under-fitting occurs when the model does not have enough hidden neurons

to detect the signal used to distinguish each target variable from others, resulting in poor accuracy. In this case, the number of neurons in every layer is set to be the same. It also can be made different. The number of neurons should be adjusted to the solution complexity. The task with a more complex level to predict needs more neurons. The number of neurons range is set to be from 10 to 100.

- **Number of hidden layers**: The number of hidden layers is the next hyper-parameter to be considered. The number of hidden layers in convolutional neural networks can refer to either convolutional layers or dense layers. The number of hidden layers in this study refers to the number of dense layers. The purpose of either layer is simple: it transforms the data received by an activation function and sends the resulting value to the output layer or another hidden layer. There is currently no comprehensive study to provide a general guideline for how many convolutional layers should be used; however, problems requiring more than two hidden layers are uncommon. However, this does not preclude the user from adding more. By adding more layers, the network's ability to learn more complex patterns within a data set can be increased by successively learning from each other. In a network used for image classification, for example, the purpose of each convolutional layer may be different, and the network will automatically assign its function. The first hidden layer may be used to detect image edges; the second layer to detect geometric shapes; and the third layer to detect facial features
- **Learning Rate**: The learning rate is one of many hyper-parameters that can be tweaked in a neural network to improve model accuracy. The learning rate is a constant ranging from 0 to 1 that aids in the adjustment of network weights toward a

local or global minimum for an error objective function. The learning rate can be thought of as the speed at which the network learns, or how quickly the network's weights converge. When the learning rate is low, the weight adjustments are small, causing the network to take a long time to converge. If the learning rate is increased too quickly, the learning becomes unstable and may exceed the optimal solution. A higher learning rate allows the model to learn more quickly, but it may miss the minimum loss function and only reach its surroundings. A lower learning rate increases the likelihood of finding a minimum loss function. As a tradeoff, a lower learning rate necessitates longer epochs, or more time and memory capacity.

- **Batch-Size**: The final hyper-parameter selected is batch size. The batch size is the number of training samples used to make a single update to the model parameters until all training samples have been fed through the network. Ideally, each training sample would be used to update the model parameters to train a network; however, this is extremely inefficient and time consuming. The average of the gradients of each sample in the batch is calculated and then used to update the model parameters by using a batch of samples per iteration. Intuitively, batch size affects both the time it takes to train the network and its accuracy.

This concludes the section on hyper-parameters and why each hyper-parameter selected will be used to perform hyper-parameter optimization. To summarize, hyper-parameters are gears that tune model parameters, resulting in a more accurate learning model.

CHAPTER 4. VALIDATION

The dataset consists of 672 goodware samples and 845 ransomware samples. The ransomwares files consist of five different families: Cerber, TeslaCrypt, Locky and Darkside. The goodware PE files are collected from windows platform and the from Portable Apps platform [39]. The datasets are split into two for training and testing purposes. The model is trained with 90% of the real ransomware file samples and augmented samples. The proposed CNN-based ransomware detection model is designed and trained in the COLLAB a cloud computing platform provided by Google. The dimensions, activation shapes and sizes for the CNN architecture are shown in Table 2.

Table 2: The Dimensions, Activations Shapes and Sized of the CNN Architecture

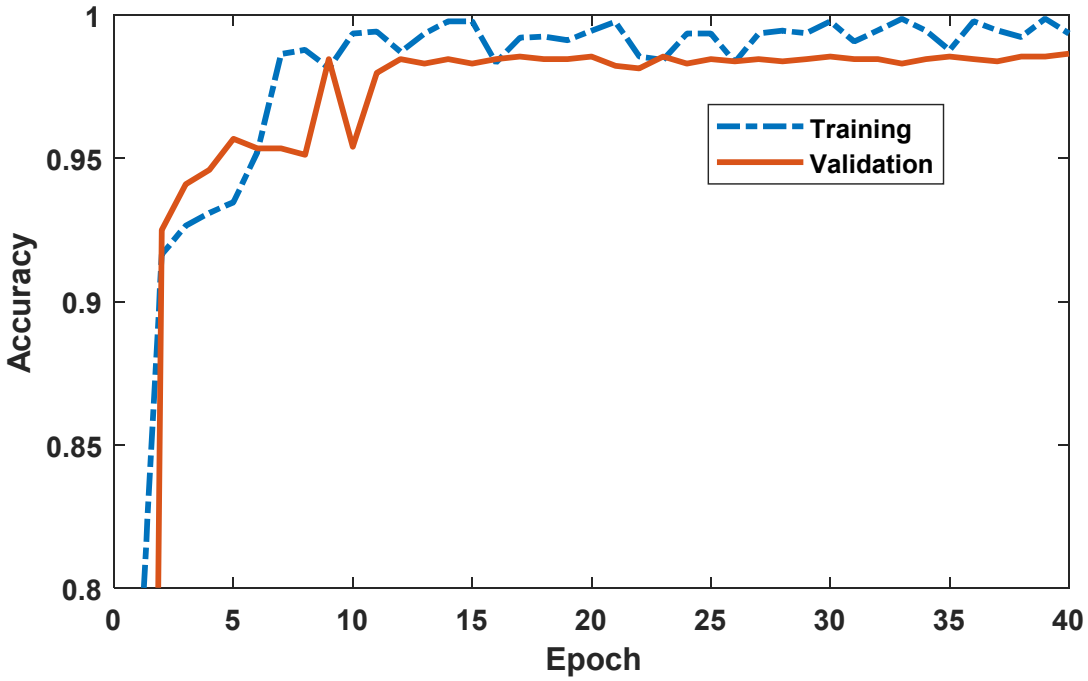
Layer	Number of Filters	Activation Shape	Activation Size
Input Image	-	(28,28,1)	784
Conv2d (f=3, s=1)	8	(28,28,8)	6,272
MaxPool (f=2, s=2)	-	(14,14,8)	1568
Conv2d (f=5, s=1)	16	(10,10,16)	1600
MaxPool (f=2, s=2)	-	(5,5,16)	400
Flatten	-	(400,1)	400
Flatten	-	(120,0)	120
Dense	-	(64,1)	64
SoftMax	-	(10,1)	10

The experiment is run on Windows computer running i7 9750H, RTX 2060, 16GB RAM. The program is written in Python 3.9.7 with Keras and PyTorch as backend. Table 3 shows the range and optimal values of these hyperparameters chosen by the CNN model.

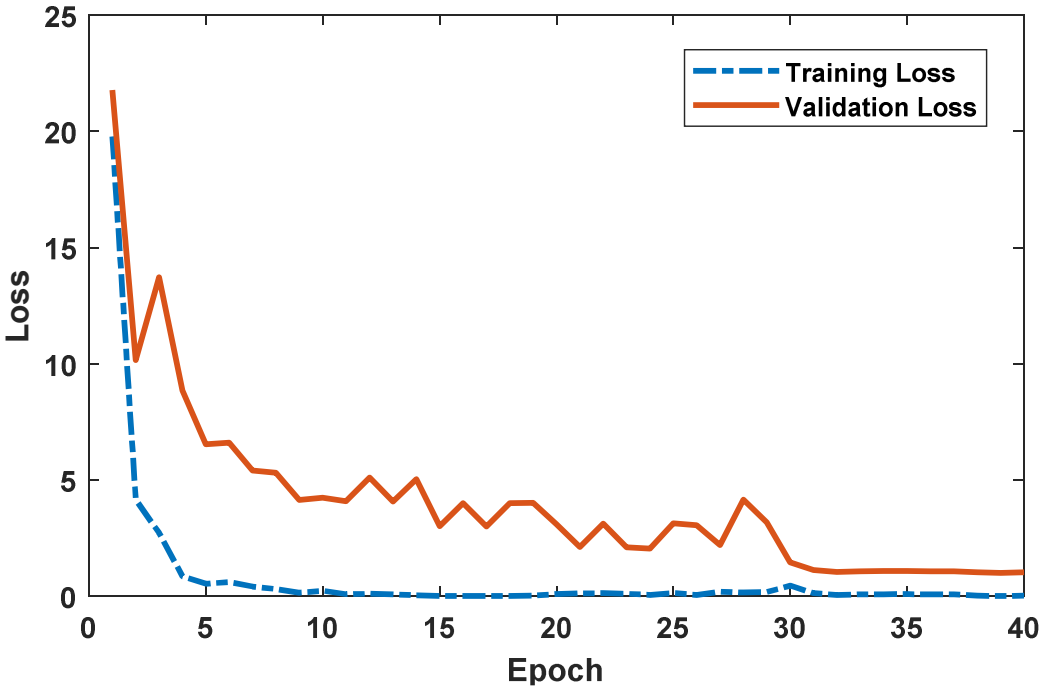
Table 3: Optimal Hyper-Parameters Selected for the Model

Hyperparameter	Range	Optimal Value
Learning Rate	[0.001, 0.01, 0.1, 0.2]	0.01
Batch Size	[5,10,15,20]	15
Epochs	[10,20,30,40,50]	40
Optimizing Algorithm	SGD, Adam, RMSprop	Adam

Figure 9(a) depicts the accuracy results of the training and experiments of the CNN-based detection model for 40 epochs. The accuracy values of training and validation converge to 99 % and 96.22%, respectively. The difference between these two accuracies is negligible. Figure 9(b) shows the loss curves of training and validation converging to 0.0056 and 0.0388, respectively. This shows that the proposed model is unbiased for the training images, but also it provides high ransomware detection accuracy. These results show the proposed CNN-based detection method is accurate and suitable for the ransomware file detection.



(a)



(b)

Figure 9. Training and Validation results of the CNN model: (a) accuracy and (b) loss.

The precision, recall and the F-1 score of the proposed model are shown in Table 4.

Table 4: Performance Metrics of the CNN Model

Metrics	Result
Precision	0.966
Recall	0.953
F-1 Score	0.976

We tested a pre-trained VGG16 model with the same dataset. But the training accuracy and validation accuracy were lower than that of the tuned CNN model. The training accuracy and the validation accuracy converged to 95% and 92% respectively as shown in Figure. 10

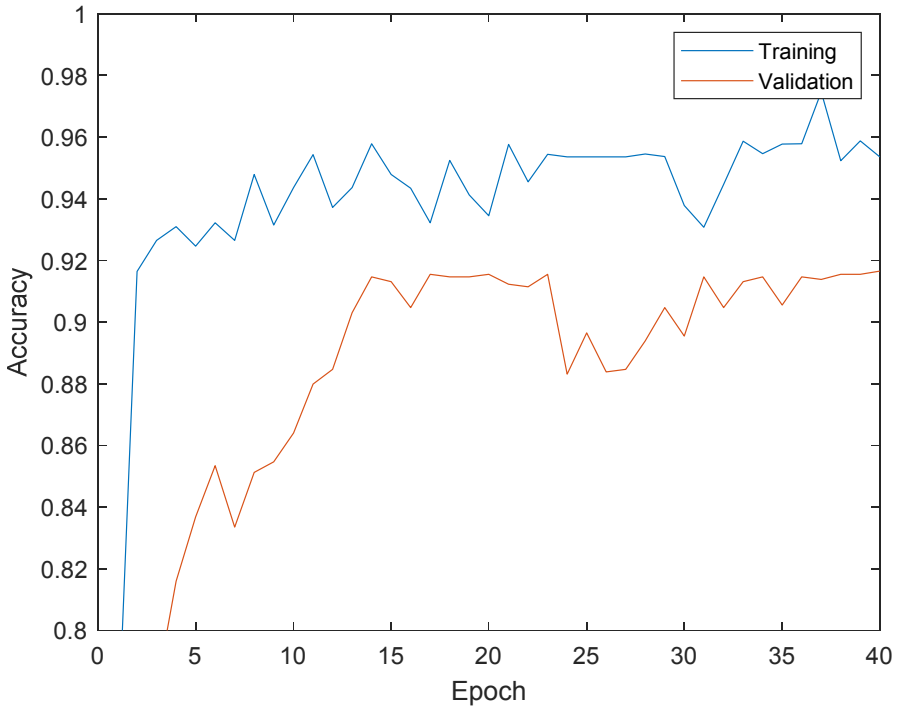


Figure 10. Training and validation results of the VGG16 model.

Table 5 shows the comparison of the proposed CNN method and a RF method based on N-gram of opcodes which shows the best accuracy among ML classifiers among decision tree (DT), K-nearest neighbors algorithm (KNN), naive Bayes (NB), and gradient boosted decision trees (GBDT) []. Moreover, the opcode-based feature extraction technique requires a disassembler to get the opcode from the file, while the CNN-based method does not require the disassemble process by extracting features directly from the raw data. The comparison shows that the proposed CNN model using images provides better accuracy compared to the ML methods using opcodes.

Table 5: The Comparison of Ransomware Detection Algorithms

Method	Feature Extraction	Datasets (ransomware/goodware)	Accuracy
Proposed	Images from raw files	672/845	96.22
Zhang et al. [39]	Opcodes from raw files using a disassembler	1787/100	91.43

CHAPTER 5. CONCLUSION

This thesis has reviewed ransomware attacks and detection methods. Moreover, this thesis explored the potential attack surface of ransomware attacks in a digital substation and provided a CKC-based ransomware attack model and proposed an AI-based ransomware file detection method. This study proposes a strategy for detecting harmful internal/external assaults without misprediction. Images from benign files and ransomware have been used to identify ransomware since the images have similarities with other variants. Secondly, by using a deep learning model based on CNN, malware has been detected with higher accuracy than peers. Since new cyber threats emerge every day, it is beneficial to use machine learning and deep learning techniques to help schemes learn and become resistant to various types of malware attacks. Furthermore, the AI-based algorithms may be automated to learn from new assaults and defend against them in the future. Future works also include: 1) evaluating the proposed algorithm in the IDS in a real-time hardware-in-the-loop (HIL) cybersecurity testbed for a smart substation, 2) improving the detection accuracy and reducing detection time, 3) studying and developing firmware malware attacks and defense methods.

REFERENCES

[1] Alert (TA13-309A), CryptoLocker ransomware infections [Online]. Available: https://us-cert.cisa.gov/ncas/alerts/TA13-309A [Accessed: 10th April 2022]

[2] Kaspersky Lab, Kaspersky Security Bulletin, 2016.

[3] Alert (TA17-132A), Indicators associated with WannaCry ransomware, [Online]. Available: https://us-cert.cisa.gov/ncas/alerts/TA17-132A [Accessed: 10th April 2022]

[4] S. Larson and C. Singleton, "Ransomware in ICS environments," Dragos, Inc., White Paper, Dec. 2020.

[5] Novinson, M, Colonial pipeline hacked via inactive account without MFA. Accessed July 7, 2021, [Online]. Available: https://www.crn.com/news/security/colonial-pipeline-hacked-via-inactive-account-without-mfa [Accessed: 10th April 2022]

[6] ICS-CERT, Alert (ICS-ALERT-14-281-01E): Ongoing sophisticated malware campaign compromising ICS, ICS-CERT, [Online] Available: ics-cert.uscert.gov/alerts/ICS-ALERT-14-281-01B [Accessed: 08th April 2022]

[7] US-CERT, [Online] Available: us-cert.gov/sites/default/files/publications/JAR 16- 20296A GRIZZLY STEPPE-2016-1229.pdf (Accessed: 03th April 2022)

[8] P. Polityuk, Ukraine to probe suspected Russian cyber-attack on grid, Reuters, [Online] Available: www.reuters.com/article/us-ukrainecrisis-malware/ukraine-to-probe-suspected-russiancyber-attack-on-grid-idUSKBN0UE0ZZ20151231[Accessed: 10th April 2022]

[9] K. Zetter, Everything we know about Ukraines power plant hack, Wired, [Online] Available: www.wired.com/2016/01/everything-we-know-aboutukraines-power-plant-hack/ [Accessed: 12th April 2022]

[10] K. Zetter, Inside the cunning, unprecedented hack of Ukraine's power grid, Wired, [Online] Available: www.wired.com/2016/03/inside-cunningunprecedented-hack-ukraines-power-grid/ [Accessed: 12th April 2022]

[11] N. Zinets, Ukraine hit by 6,500 hack attacks, sees Russian 'cyberwar', Reuters, [Online] Available: www.reuters.com/article/us-ukraine-crisiscyber/ukraine-hit-by-6500-hack-attacks-sees-russiancyberwar-idUSKBN14I1QC [Accessed: 12th April 2022]

[12] ICS-CERT, MAR-17-352-01, Hatmansafety system targeted malware, ICS-CERT, [Online] Available: ics-cert.uscert.gov/sites/default/files/documents/MAR-17-352- 01 [Accessed: 11th April 2022]

[13] Symantec, Triton: New malware threatens industrial safety systems, Symantec, [Online] Available: www.symantec.com/blogs/threat-intelligence/ triton-malware-ics [Accessed: 10th April 2022]

[14] B. Johnson, D. Caban, M. Krotofil, D. Scali, N. Brubaker, and C. Glyer, Attackers deploy new ICS attack framework 'TRITON' and cause operational disruption to critical infrastructure, FireEye, [Online] Available: www.fireeye.com/blog/threat-research/2017/12/attackersdeploy-new-ics-attack-framework-triton.html [Accessed: 10th April 2022]

[15] R. Lee, M. Assante, and T. Conway, Analysis of the cyber-attack on the Ukrainian power grid, NERC, [Online] Available: nerc.com/pa/CI/ESISAC/ Documents/E-ISAC SANS Ukraine DUC 18Mar2016.pdf [Accessed: 10th April 2022]

[16] [Online]. Available: https://threatpost.com/badrabbit-ransomware-attacks-hitting-russia-ukraine/128593/?fbclid=IwAR2fiGm8rZPNeorlhZ8mVn8o3NzqMi88O6_k_6aMKlNe-pkNA5EFc26bPc0 [Accessed: 10th April 2022]

[17] [Online]. Available: https://www.virustotal.com/gui/home/upload [Accessed: 4th April 2022]

[18] B. Ahn, G. Bere, S. Ahmad, J. Choi, and T. Kim, "Blockchain-enabled security module for transforming conventional inverters toward firmware security-enhanced smart inverters," in *Proc. 2021 IEEE Energy Conversion Congress and Exposition*, Vancouver, Canada, Oct. 10-14, 2021, pp. 1307-1312.

[19] A. Kharraz et al., "Cutting the gordian knot: A look under the hood of ransomware attacks," in *Proc. Int. Conf. Detection of Intrusions and Malware and Vulnerability Assessment*, 2015.

[20] B. M. Khammas, "Ransomware detection using random forest technique," *ICT Express*, vol. 6, no. 4, pp. 325–331, Dec. 2020.

[21] G. Cusack, O. Michel, and E. Keller, "Machine learning-based detection of ransomware using SDN," in *Proc. the 2018 ACM International Workshop on Security in Software Defined Network & Network Function Virtualization*, Tempa, AZ, USA, Mar. 21, 2018, pp. 1–6.

[22] F. Maimo, *et al.*, "Intelligent and dynamic ransomware spread detection and mitigation in integrated clinical environments," *Sensors*, vol. 19.5, no. 1114, 2019.

[23] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition,* Las Vegas, NV, USA, June 27-30, 2016, pp. 770–778.

[25] X. Li, W. Jiang, W. Chen, J. Wu, G. Wang, and K. Li, "Directional and explainable serendipity recommendation," in *Proc. WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*,

[26] S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi, "Malware detection with the deep neural network using process behavior," in *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, pp. 577–582, 2016.

[27] K. He and D. S. Kim, "Malware detection with malware images using deep learning techniques," in *Proc. 18th IEEE International Conference on Trust, Security And Privacy In Computing And Communications / 13th IEEE International Conference On Big Data Science And Engineering, TrustCom/BigDataSE 2019, Rotorua, New Zealand, August 5-8, 2019*, pp. 95–102.

[28] X. Xiao, "An image-inspired and CNN-based android malware detection approach," in *Proc. 34th IEEE/ACM International Conference on Automated Software Engineering (ASE 2019), San Diego, CA, USA, Nov. 11- 15, 2019*, pp. 1259–1261, 2019.

[29] Z. Cui, F. Xue, X. Cai, Y. Cao, G. Wang, and J. Chen, "Detection of malicious code variants based on deep learning," *IEEE Trans. Industrial Informatics*, vol. 14, no. 7, pp. 3187–3196. Jul. 2018.

[30] K. Han, *et al.,* "Malware analysis using visualized images and entropy graphs," *Int. J. Inf. Secur.* vol. 14, pp. 1–14, 2015.

[31] Z. Cui, F. Xue, X. Cai, Y. Cao, G. -g. Wang and J. Chen, "Detection of malicious code variants based on deep learning," in *IEEE Trans. Industrial Informatics*, vol. 14, no. 7, pp. 3187-3196, Jul. 2018.

[32] S. Venkatraman, M. Alazab, and R. Vinayakumar, "A hybrid deep learning image-based analysis for effective malware detection," *J. Information Security and Applications*, vol. 47, pp. 377-389, 2019.

[33] Y. Baoguo, W. Junfeng, L. Dong, G. Wen, P. Wu, and X. Bao, "Byte-level malware classification based on markov images and deep learning," *Computers & Security*, vol. 92, 2020, 101740, ISSN 0167-4048.

[34] L. Ghouti and M. Imam, "Malware classification using compact image features and multiclass support vector machines," *IET Inf. Secur.*, vol. 14, pp. 419-429, 2020.

[35] Aslan, Omer & Yılmaz, Abdullah. (2021). A New Malware Classification Framework Based on Deep Learning Algorithms. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3089586.

[36] MITRE's ATT&CK for ICS, [Online]. Available: https://collaborate.mitre.org/attackics/index.php/Main_Page [Accessed: 10th April 2022]

[37] [Online]. Available: https://us-cert.cisa.gov/ncas/alerts/aa20-352a [Accessed: 10th April 2022]

[38] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in *Proc. 8th Invetenation Symposium on Visualization for Cyber Security*, Pittsburg, PA, USA, Jul. 20, 2011, pp.1–7.

[39] [Online]. Available: https://portableapps.com/apps. [Accessed: 10th April 2022]

[40] H. Zhang, et al., "Classification of ransomware families with machine learning based on N-gram of opcodes," *Future Gener. Comput. Syst*. vol. 90 pp. 211–221, 2019.

VITA

SYED RAQUEED BIN ALVEE
syed_raqueed_bin.alvee@student.tamuk.edu
(361) 228-1452

EDUCATION

Texas A&M University, Kingsville Expected graduation: May 2022
Kingsville, Texas
MS
Major: Electrical Engineering

American International University - Bangladesh July 2020
Dhaka, Bangladesh
B.Sc.
Major: Electrical and Electronics Engineering

SKILLS

Java, Python, C++, MySQL, Data Structures & Algorithm, AWS, Visual Studio, Google Analytics, Git, Linux, MATLAB, AutoCAD, PLC Programming, PSpice, LTSpice, SolidWorks, Proteus, MS Office, Adobe Premier Pro, Adobe After Effects

COURSEWORK

Digital Signal Processing; Data Mining; Digital Image Processing; Applications of Neural Network; Speech Processing; Machine Learning; Principles of VLSI Circuit Design; Embedded System Design

WORK EXPERIENCE

Graduate Research Assistant

Cyber-Physical Power and Energy Systems Laboratory, TAMUK

- Designed a Ransomware detection method based on CNN using Gray-scale image for digital sub-stations.
- Implemented a Deep Transfer Learning (DTL)-based firmware malware detection system for smart inverters.
- Designed a Ransomware security threat model for PV systems and smart inverters.
- Worked on False battery detection and classification, Battery Management system, Battery Energy Storage System.
- **Technologies used** - Python, PyCharm, VMware, MATLAB