EAI Endorsed Transactions
on Smart Cities

Research Article **EAI.EU**

Raspberry Pi-based Intelligent Cyber Defense Systems for SMEs and Smart-homes: An Exploratory Study

Sreenivas Sremath Tirumala[η] *, Narayan Nepal[α] and Sayan Kumar Ray[η]

[η]School of Digital Technologies, Manukua Institute of Technology, Auckland, New Zealand
[α] Yoobee Colleges, Christchurch, New Zealand.

Abstract

Ongoing ransomware attacks have forced business to think about security of their resources. Recently, small-to-medium enterprises (SMEs) and Smart-homes have become easy targets for attackers since they don't have cyber defense mechanism in place other than simple firewall systems which are quite vulnerable. Cyber defense systems are costly and often not within the budget of SMEs or families which inspired to think about low cost yet highly efficient cyber defense solutions. Regular individuals and families who use internet for day to day use often end-up becoming a possible resource for using them as Trojan or bitcoin nodes. This research explores the prospects of implementing a Raspberry Pi (Raspberry Pi)-based intelligent cyber-defense system (iCDS) for SME networks and Smart-homes to filter malicious contents from incoming traffic and detect malware using artificial intelligence.

Primarily, the work presented in this paper tries to evaluate the hardware capability of network interfaces (both internal, and attached) of Raspberry Pi for handle high volumes of incoming traffic. For this, we measure the network performance of the Raspberry Pi using the speed test software and try to explore the possibility of a light weight machine learning (ML) based malware detection. The results show that the built in Ethernet interface outperforms the built in WiFi and external attached USB to Ethernet Adapter in terms of latency, download and upload throughput. Also, a new DNA based ML approach was successfully able to produce over 19.5% better accuracy rates of over classifier trained with hash-sequence. The experiment results further emphasise on the importance of generating complex malware signatures with variety to face existing threats which has taken a new form due to increase in malware based attacks, particularly for ransomware. The complexity of the generated malware is based on generic yet strong encryption principles which produced good results which is quite encouraging at this stage.

Received on 15 March 2022; accepted on 25 July 2022; published on 03 August 2022

Keywords: Cyber defense, Raspberry-Pi, intelligent cyber-defense system

doi:10.4108/eetsc.v6i18.2345

1. Introduction

Cybersecurity is the point of interest in this internet reliant age of e-connectivity,e-appliances and smart homes. Privacy, Security and Trust are the three pillars of cybersecurity [1, 2]. The importance of privacy and trust are more related to data security and are protected through the implementation of security services, framework and standards [3]. The security relies on technology, software as well as principles of application. Though majority of security systems are based on standards the reliability of these software became questionable with the recent incidents of ransomware attacks. It can be noticed that the aspect of reliability is not just confined to standards or rules or even type of software. The reliability is based on

*Sreenivas Sremath Tirumala. Email: sreenivas.tirumala@manukua.ac.nz efficiency of the system to stop unknown variants or malware.

The internet revolution had notable impact on day-to-day activities of both individuals and businesses. Smart-devices became part and parcel of daily life. With the development of Smart-Homes the reach of internet had a new horizon. Also, recent cyber-revolution impacted the growth of small-to-medium enterprises (SMEs) due to reduction of cost and availability of resources. The cloudification (moving the software and other operations services to cloud) of SMEs partially impacted the dependence on local hardware and networking configurations.

This reliance on internet made SMEs and smart homes exposed to the rest of the world, particularly for hackers as soft targets for exploitation particularly through ransomware attacks. The operational implications of providing a secure environment for SMEs is costly due to demanding resource requirements like manpower and technology. With limited operational budget, majority of SMEs rely on internet service providers (ISPs) and local firewall or antivirus software for providing IT security. In countries like New Zealand, where majority of the business are SMEs, impose budget and resource constraints and are not be able to afford operational costs for providing cyber defense systems. According to a survey conducted by InternetNZ, about 48% of computers in SMEs are used by hackers for testing new malware and / or as bots to simulate Denial of Service (DoS) attacks. Also, considering the recent events where gaming devices are used for mining bitcoins, there is a high chance for Smart-Homes being easy targets by hackers. Hence, the internal networks of SMEs and Smart-Homes have to be secured enough to prevent such external attacks [4].

Simple rule-based firewalls (i.e., based on administrator defined policies) of SMEs have failed to prevent attacks from random malware. Rule-based intruder detection systems (IDS) have managed to counter the attacks to some extent but not fully capable to provide complete security to the organization's network. The rule-based systems simply monitor and filter incoming network traffic based on set of predefined rules (malware signatures) stored in the repository. From the literature and implementation documents [5] it can be concluded that highly efficient IDS is more powerful and assertive in identifying malicious packets entering a network. However, traditional IDS requires special equipment and manpower and thus are resource savvy and costly to install and maintain. Also, it requires regular upgrades to identify and respond to new threats. Thus, majority of the SMEs with limited budget find it difficult to implement and maintain an effective IDS. Implementing low cost IDS solution that can operate as Security as a Service (SECaaS) and can be offered as subscription-based service, is another option for SMEs to consider. However, SECaaS still relies on rule-based systems and incurs all drawbacks of cloud-based and other remote service offerings. Moreover, the fact that SECaaS is expensive, a major concern for SMEs and are not effective for networks with IoT based devices [6, 7]. With the rapid integration of IoT with traditional networks, SECaaS may become a burden as the subscriptions needs to be paid in spite of them being used few times, purging less resources or bandwidth.

Formerly, computer networks are protected by firewall from the external attacks which is not different for smart-homes and SMEs. However, the usage of algorithms to create malware with no standard structure or pattern challenged the capabilities of simple rule-based firewalls and IDS. Majority of the firewall systems as well as IDS are based on administrator defined policies, or in simple terms, rule based. At present, the traffic is monitored and 'filtered' based on a set of rules (malware signatures) present in the repository. The limitations of firewalls, IDS and SECaaS discussed above, indicate an immediate necessity of introducing a low-cost, low-resourced yet advanced network security solution for SMEs particularly for stopping, as much as possible, the malicious network traffic from entering the networks.

1.1. Malware Detection

Malware detection has been a key aspect of cybersecurity particularly with recent developments in cloudification i.e., moving application to cloud. Traditional malware detection is based on matching malicious imprints (hash) through a fuzzy logic based comparison. Signature based malware detection is popular and often considered as efficient for detecting malware in the incoming traffic for the signature that exists in the repository [8]. The signature based approach is capable of handling new unknown variants to some extent. However, it is time-consuming and often impractical to keep the repository updated based on new variant particularly with the rate of generation of new variants with complexity and variety. Recent advances in Machine Learning (ML) and encryption based methods have enabled attacker to adopt these approaches to generate new variants that challenged traditional repository based techniques including the signature based malware detection approaches [9].

Traditional malware detection, either rule based or signature based is time consuming and often not efficient in detecting all if not majority of malware imprints due to significant changes in the structure and patterns of the new variants [10]. As early as 2017, Institute for Critical Infrastructure Technology has hinted at the demise of signature based malware detection in the technical report [11]. Considering malware detection as a pattern recognition problem, several ML based malware detection approaches have emerged [12, 13]. Using bio-inspired approaches for malware detection is also area of high interest in recent times and has been considered an alternative for ML based pattern recognition problems. For instance, malware detection and analysis using DNA based approaches has been a topic of research interest since 2012 [14].

This inspired to undertake an exploratory study on designing an intelligent intruder detection systems (iids) that can be implemented on a low-cost device to provide a small budget solution to SMEs and smart-homes. There has been some background work on non-rule based (pattern recognition based) solution for detecting malware [15] This paper explores the prospects of implementing a low-cost intelligent cyber defense system (iCDS), in form of a filtering device, to protect the SMEs from malicious traffic. The proposal considers the plausibility of using Raspberry Pi device as a commercial IDS with the purpose of filtering malicious network traffic from entering SME networks. Primarily, through a systematic experimental evaluation this work tries to explore the capability of network interfaces of Raspberry Pi device to understand their competence in handling high volumes of incoming traffic similar to commercial IDS systems.

A comparative study of the performance of the inbuilt network interfaces, namely Ethernet (wired) and WiFi on the Raspberry Pi device, as well as an externally connected USB adapter interface (USB to Ethernet interface) are carried out in context to network parameters like latency, download throughput and upload throughput.

The key contribution of this paper is providing a framework for a systematic research on implementing low-cost IDS systems with advanced Machine Learning (ML) based approach. Although there has been research on using ML algorithms for detecting intruders, a combination of low-cost Raspberry Pi and ML based approaches for IDS have not been undertaken. Moreover, earlier works uses only one network interface whereas this research is proposing a systematic approach to evaluates a combinations of network interface devices.

This research also contributes towards using feature extraction and comparison for detecting malware or network anomalies. Typically, there will be a separate feature extractor and classifier. This research for proposes to use autoencoders, a special type of artificial neural networks which is can be used as feature extractor and classifier. This research would also encourage to use Raspberry Pi or similar devices for portable and low cost devices for IDS.

Smart cities often uses low powered internet based devices that can be controlled over internet. Hence, privacy and security are the two key aspects to be considered. Thus, all smart city solutions requires cyber security devices for detecting malware / intruders. Smart cities also requires IDS solutions that are low cost since the purpose is private and individual. The iCDS proposed in this paper is an ideal and well suitable for smart cities since iCDS is low cost, less computations savvy. Also, iCDS provides a smart and intelligent system that provides real-time updates for detecting malware.

The remainder of the paper is structured as follows. Section II provides a literature review of the different filtering approaches and DNA based malware detection approaches.This section also While, Section III explores the prospects of using Raspberry Pi device as an iCDS, Section IV discusses the evaluation results and Section V concludes the paper.

2. Related Work

There is a lack of systematic literature review on implementing low cost IDS solutions for Smart-Homes and SMEs. Furthermore, very few research projects have been done on the feasibility of implementing Raspberry Pi (or a similar device)-based low-cost IDS for Smart-Homes and similar small networks that exists in SMEs and smart homes. This research gap provides an immediate necessity of such a research study to start with. A standard case of identifying low cost IDS solution for smart homes and SME networks (containing different IoT devices), particularly using Raspberry Pi based implementation, is relevant to the current research. Also there is a significant rise in the usage of IoT based security devices for smart homes and SMEs [16]. Also with the recent advances in using smart devices, it is widely accepted that there are several security concerns that needs to be addressed [17].

2.1. IoT based IDS implementations

IoT-based IDS implementations proposed by the research fraternity are mostly for non-commercial purpose and are either policy-based or graph-based. Policy-based approaches [18, 19] depend on a fixed predefined policy based on a specific domain or problem-based scenario similar to traditional network traffic packet filtering approaches. The graph-based approaches [20] implement polices stored in a repository, which can be updated periodically (follows a dynamic rule). Such updates, however, lead to latency. A Raspberry Pi based firewall proposed by [21] to secure home networks, uses a remote cloud database with set of predefined rules. It uses on-board Ethernet interface for incoming network traffic and WiFi for outgoing traffic. The proposed approach is prone to delays and when applied for SME networks may incur significant latency. Another non-commercial implementation named as Pi- IDS is a Raspberry Pi 2.0-based standalone firewall implemented to filter websites in a school network. Although, an interesting concept, it has significant limitations in context to operation time and network traffic filtering capability.

Few research also proposed installing open source IDSs on Raspberry Pi so that it can replace a regular computer and can operate as a complete IDS of its own. For example, NetGaurd, proposed for traffic monitoring to track man-in-the-middle attacks, installs an open VPN and IDS software on Raspberry Pi to implement a complete IDS [22]. However, NetGaurd is nothing different to a traditional IDS and just provides privacy by hiding the IP of the monitoring source, as an extra feature. There are few other similar implementations like [23, 24]. The mere purpose of these implementation is to install and test IDS software on Raspberry Pi for various purposes. Two other research proposed by [25, 26], used classification techniques for detection malicious contents in incoming network traffic. However, not only these two proposals lacked the technical details of hardware and software limitations of Raspberry Pi when experimenting it as an IDS, but also, they considered limited traffic with known malicious variants during the experiments. So, previous research mostly focused on studying how Raspberry Pi-based IDS can be implemented and if it can replace the traditional rule-based IDS implemented on normal computers. These implementations, knowingly or unknowingly overlooked the different challenges, including hardware limitations, to make Raspberry Pi operate as a fully commercial and real-world implementation of IDS. Furthermore, such implementations are vertically divided into cloud based and non-cloud based and do not emphasize the need of a mixed model or fail- over model.

Filtering Approaches. Traditional firewalls and IDSs use packet inspection for filtering traffic based on malware impressions [27–29]. The workable solution proposed in [28] used a conceptual 'trust' based filtering that only allowed 'useful' packets to pass through. False positive results are often produced by the trust-based approach (similar to traditional fuzzy rule-based approach) and hence it was inconsistent in nature [28]. However, the proposed approach was successful in detecting malicious contents resulting from insider attacks in an organization. Since, iCDS mostly deals with identifying and

filtering malicious contents from external network traffic trying to penetrate inside an SME network, insider attacks at this stage of the research is not considered. The filtering approach presented in [27] consisted of a restriction and access policy working as a traditional gateway. However, no evidence of experimental evaluation of the approach is proposed. An interesting machine learning based filtering model using Support Vector Machines (SVM) and Naïve Bayes is presented in [29], which also provides a good practical implementation scenario. However, due to its resource heavy and computationally complex nature, this proposed approach is unsuitable for SMEs. All these discussed research work provide an overview of important methods proposed for malicious network traffic filtering based on purpose and relevance. However, these implementations are generic in nature, not cost effective, and demand high configuration hardware for implementation. The next subsection discusses the implementation of Raspberry Pi-based low-cost IDS systems.

Key Challenges to Consider in Raspberry Pi-based IDS. On a practical note, the following challenges need to be considered if implementing a Raspberry Pi- based iCDS for filtering malicious network traffic contents from entering SME networks.

- *Handling high volumes of traffic:* Raspberry Pi has one on-board Ethernet port, which limits and delays the flow of incoming (from the internet) and outgoing traffic (after filtering). How to handle such latency ? If external Ethernet adapter is used, what are its implications in terms of power, cost and heat?
- *Processing capabilities:* Raspberry-Pi, being an embedded system has a low end process and its processing capabilities may create some issue while handling the traffic and may effect a significant increase in processing and serialization delay too.
- *Heat and Power Source:* Is the hardware of Raspberry Pi capable enough to run continuously and uninterrupted for a week?
- *Storage and Real-time updates of Repository:* Efficient mechanism to store and update the repository (for rule based, signature based or any other approach).

The overall research consists of various plausibility studies for hardware, software and algorithms. The AI-based algorithmic evaluation is been initiated and published [4]. This systematic experimental evaluation presented in this paper is confined to understand the capability of input network interface(s) of Raspberry-Pi.

2.2. Malware Detection Approaches

Malware detection using signatures has been in the practise since the early days of anti-virus designing and development. A malware signature is an unique

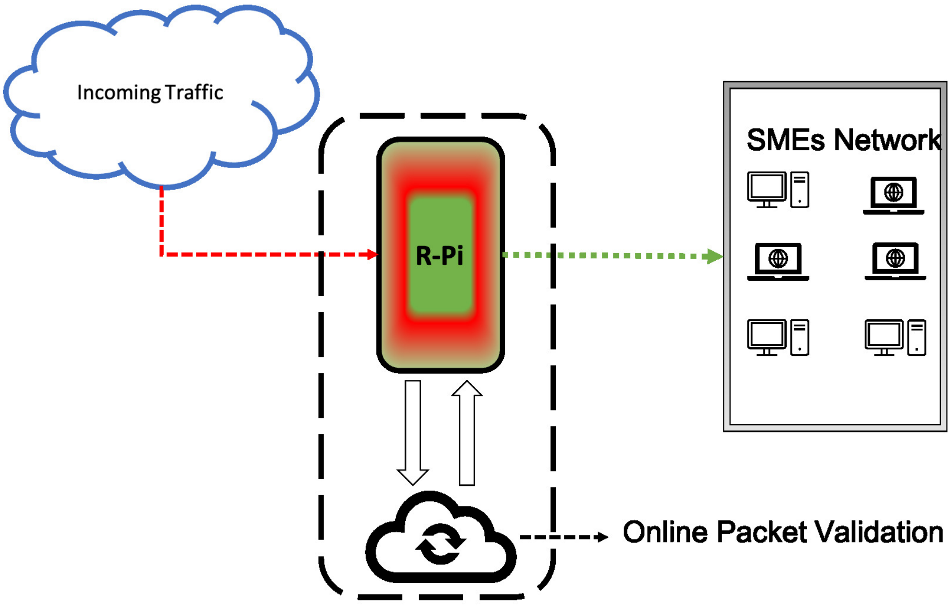

Figure 1. The block diagram of iCDS representing various components

identifier, typically a hash-sequence, defined using as set of hexadecimal character set. A malware signature can also be considered as an unique footprint of a particular malware with a set of characters. The detection of a malware is performed through searching for hash-values that are imprints of malware in the input data, files or streams. In other words, signature based malware detection requires a repository with existing signatures as a prerequisite. Malware detection approaches or methods can be categorised based on process or steps involved in detection, type of detection method used, techniques adopted for implementation or a combination of one of more of these approaches to form a hybrid method. Literature presents a vertical categorisation of malware detection into two main categories signature based and anomaly based. Fig. 2 presents the process of signature and anomaly based approaches which can be considered as the starting point for understanding the process of malware detection [30].

Signature based approaches relies on repository with signatures or rules or both where as anomaly detection process is based on creating a profile from the input data followed by identifying unknown anomalies. Malware detection through anomaly based detection technique is also known as behaviour or heuristic malware detection. The key issue with anomaly based detection is its credibility of creating efficient profile. Moreover, an anomaly cannot be identified based on only one type of data or patterns which is considered as a major drawback [31] of anomaly based detection. However, ML based, in particular deep learning based approaches trained with good data set are more successful due to their ability to recognising complex patterns from unknown data [32].

Considering the type of detection, malware detection process can be categorised as static, dynamic and hybrid [33]. Static malware detection is based on investigating hash-sequences (sometimes also referred as binary / byte codes) inside a file whereas dynamic approach tries to detect malware by executing the file in a controlled environment or docker to see the impact to differentiate safe and malicious content. The hybrid approach is a combination of static and dynamic approaches. Dynamic malware detection techniques are sometimes referred as behaviour detection techniques creating ambiguity in conventional naming. The implementation of dynamic behavioral approach attained considerable success with both ML [34] and bio-inspired approaches like gene-based malware analysis proposed in [35].

DNA based approaches for malware detection. Bio-inspired approaches are successful in many domains including cybersecurity. Bio-inspired approaches also termed as Nature Inspired approaches can be either based on natural process or human biological process. The Nature inspired computing is based on evolution process for species selection or based on performing a task like ant colony optimisation. Human biology and cognition based approaches includes artificial neural networks, DNA computing, immunity inspired approach etc. In DNA-based approaches, a unique identifier is extracted from malicious content which is called the DNA signature. Predominantly, majority of the DNA-based malware detection is preformed using this approach of extracting unique identifier

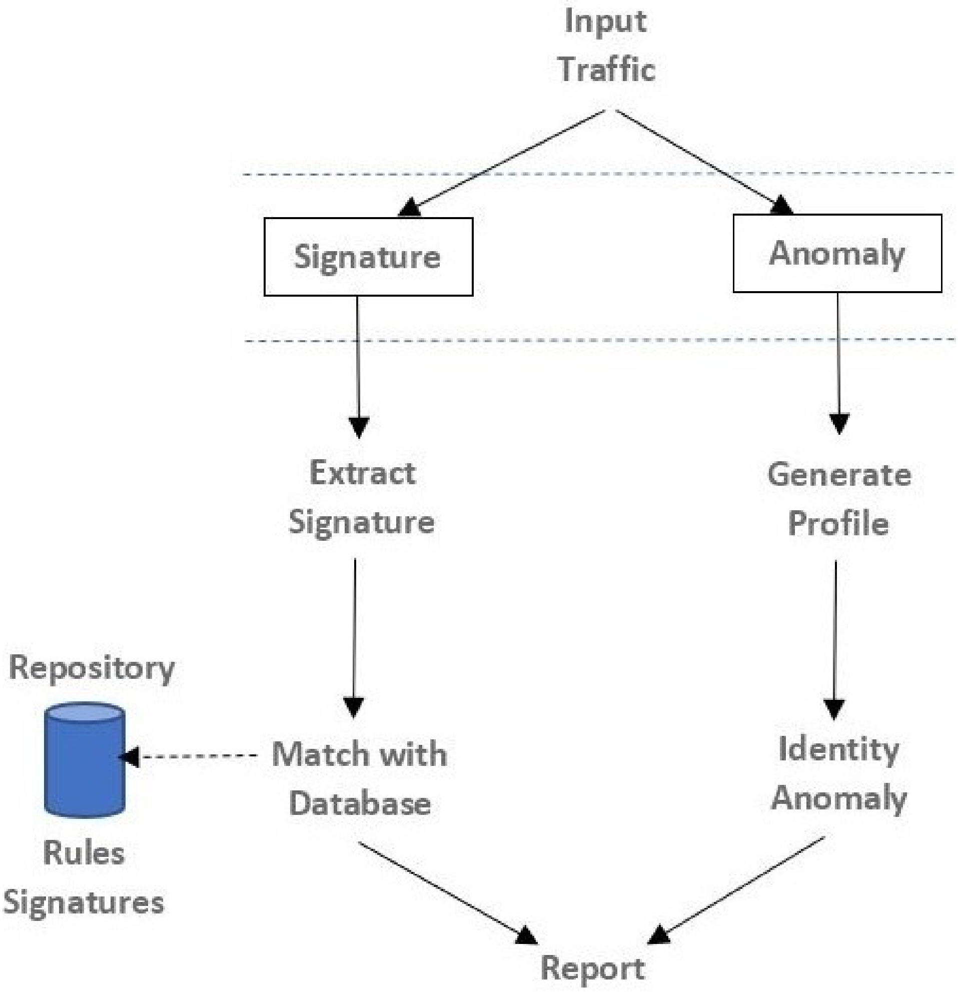

Figure 2. Malware Detection Process : The process of Signature based and Anomaly based malware detection approaches. It can be noted that signature based detection requires a rules / signatures

from a file with malicious content or data to create a repository based on the extracted DNA imprint (DNA signature) [36]. Extraction of DNA is performed using statistical or ML based feature extraction techniques. The DNA signatures thus extracted are used for detection process using ML methods through pattern recognition. Malware detection through data mining techniques was proposed in early 2000s [37]. A similar approach was proposed in 2014 for metamorphic malware detection by comparing files with and without a unique signature extracted with DNA-based approach [38]. Similarly, the DNA sequencing-based detection is recently implemented for mobile malware detection [39]. Such DNA-based techniques often used in extracting malware signatures and applying them for detecting malware by comparing the patterns. For instance, detection of android malware through generating DNA fingerprints in the package files is presented in [40].

There are several approaches in the literature which uses DNA (identifier) signature extraction applying a variety of feature extraction approaches used in pattern recognition problems including image analysis, water marking detection and other sequencing based detection. It is to be noted that the DNA signatures used in these approaches were able to extract DNA-based unique patterns to compare with a malware. But, the key issue here is whether there is enough variety and complexity in the repository or the signatures created using the repository. This poses a threat to the entire detection process since the malware detector may not be good enough to handle malware created by attackers which are often strongly encrypted and possess unique characteristics. There is also a possibility of creating malware from these signatures and adding them to the repository. However, the variety and veracity of the malware variants created through extracting DNA from the existing signatures cannot be guaranteed. The work presented in this paper is inspired from this requirement of guaranteeing variety and complexity and tries to propose a new approach for generating malware signatures. A 2013 work published in [41] cites a work (published in Korean) that emphasises on malware generation based on DNA signatures but not mentioning details about the approaches used to create variety and veracity. This can be considered as only work towards generating malware signatures from DNA-based signature repository and could not be compared to the proposed approach due to lack of technical and implementation details.

3. Proposed Testbed for the Experiments

Figure 3 shows the proposed system model for using Raspberry Pi 4 as the iCDS in order to filter malicious packets from entering SME networks. Ideally, Raspberry Pi 4.0 device with 4 GB of RAM and 1.5 GHz 64-bit quad-core Arm Cortex-A72 processor will be used. It has built in Ethernet and WiFi interfaces. The Gigabit Ethernet interface in Raspberry Pi 4.0 can reduce communication latency and provide faster network connectivity. The device also has USB 3.0 and 2.0 ports. USB 3.0 ports can enable transfer of data up to ten times faster than USB 2.0. Based on the discussion provided in the previous sub-section, the proposed model will likely opt for option 2, where the on-board Ethernet interface will be used for incoming network traffic from external networks trying to enter the SME network through the Raspberry Pi 4-based iCDS and an USB Ethernet interface (in form of an adaptor) will be used as the exit for the filtered outgoing traffic from the Raspberry Pi device to the gateway of the connected SME network (refer to Fig. 2). There is also an issue with choosing USB Ethernet for communication (option 2) as it may slow down the transfer of outgoing network traffic from the Raspberry Pi device to the gateway of the SME network, however, with the choice of proper USB Ethernet adaptor this shortcoming can be overcome. USBs are rated at speeds different to Ethernet, for instance, USB 3.0 is rated at 5 gigabits per second whereas USB 2.0 is rated at 54 megabits per second. For our proposed experimental tested in this research, a Raspberry Pi 4.0 device is used that has a Gigabit Ethernet interface. Also, to ensure that the network communication on the Raspberry Pi 4.0 board does not slow down, a USB 3.0 Gigabit Ethernet interface (adaptor) is used so that communication between the two Gigabit Ethernet interfaces (the on-board one and the USB one) can happen. All the incoming internet traffic meant for the SME network will first enter the Raspberry Pi based iCDS acting as a protective shield for the SME network.

This entire research work will be carried out in two phases. In the first phase, as mentioned before, the aim is to study the feasibility of using Raspberry device to develop the iCDS and to explore if hardware interfaces on the Raspberry-Pi device are capable of handling high volume of real traffic. These second phase activities is using Artificial Intelligence (AI) and DNA based approach for detecting malware with low hardware devices like Raspberry-Pi to support its usage as a commercial iCDS, which is what this paper will discuss. In the following phase, the incoming traffic on the Raspberry Pi device will be sent through a cloud-based validation system where the signatures of the packets will be thoroughly checked to identify malicious contents (e.g., malware). Such checking will be done at the signature-based detection online module (shown as cloud) of the proposed model where a lightweight AI-based pattern recognition and deep learning algorithm will inspect every packet to filter the malicious contents before letting the outgoing packets pass through the exit USB Ethernet interface to safely enter the SME network's gateway.

3.1. Raspberry Pi as an iCDS: From Perspective of Hardware Capability

This current research explores the prospects of implementing a Raspberry Pi (Raspberry Pi)-based low cost and intelligent cyber-defense system (iCDS) for SME networks, the architecture of which is presented in Fig. 1. In the iCDS, all incoming traffic to the network of the SME will go through the Raspberry Pi device that will scan the traffic for any malicious contents. The traffic will be monitored and filtered through a cloud-based filtering system and all malicious traffic will be quarantined for further actions by the SME. A deep learning-based signature verification system will be used for filtering the traffic in the next phase of this work. The primary focus of the work presented in this paper is to explore (a) the feasibility of using Raspberry Pi device to develop an iCDS, and (b) if the hardware components present in the latest Raspberry Pi devices are capable and compatible enough to support the use of Raspberry Pi-based iCDS for commercial SME networks. Use of Raspberry Pi as a low-cost device is becoming common in various IoT-based systems due to its simple operation, cost effective usage and support of open source software and operating systems. From the literature study presented in Section II, it can be concluded that, although, research has shown the effectiveness of using Raspberry Pi-based commercial IDSs, previous work done on this aspect (i.e., use of Raspberry Pi as a commercial IDS) have not evaluated the efficiency and capabilities of the hardware components, especially, the Ethernet and WiFi modules on the Raspberry Pi board when handling input and output traffic. Also, typically, a commercially

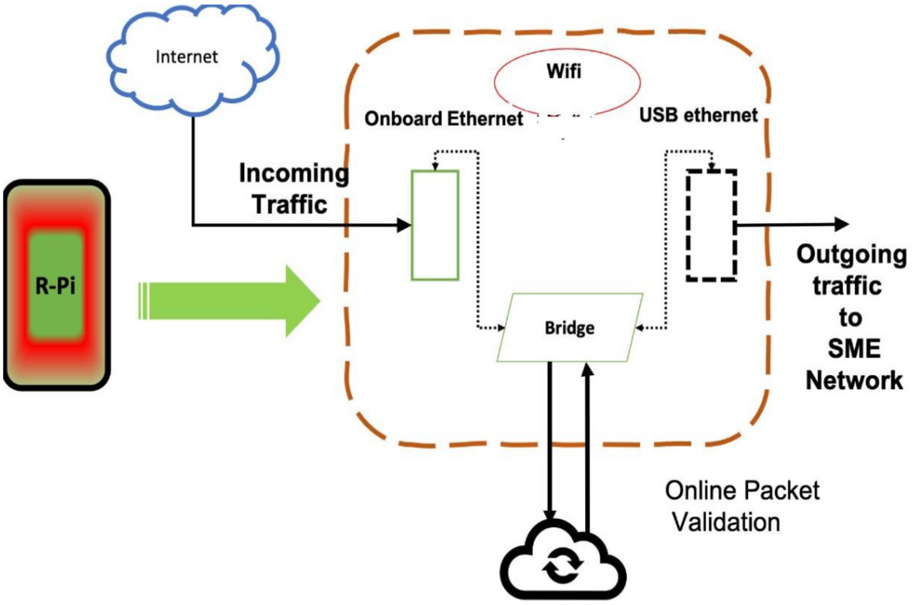

Figure 3. The hardware architecture for the proposed Raspberry Pi-based iCDS

available IDS/Firewall will need gigabyte Ethernet-based connections for its input and output interfaces depending on the network requirements but each iCDS, on the other hand, need to have at least two physical interfaces with high end throughput to segregate the internal and external network traffic from each other. Thus, to explore whether it is possible to develop a Raspberry Pi-based iCDS, monitoring the performance of the different hardware interfaces on the Pi device when handling high volume of real traffic, is necessary. The following sub-sections will discuss these in detail.

An important point that needs mentioning here is how the Raspberry Pi device can capture and track the network traffic flowing between its incoming and outgoing interfaces. This can be done in the following way. On starting, the Raspberry Pi device will load two scripts, the first of which is a shell script that will set up a software bridge connection between the incoming and outgoing interfaces. The bridge interface will have its own unique IP address assigned and will allow for network connectivity. The second Python script will tcpdump the network packets (flowing between the input and output interfaces) on the Raspberry Pi 4.0 device so that they can be captured and assessed.

3.2. Use of Raspberry Pi as an iCDS

Raspberry Pi is a low-cost computer that is commonly finding its usage in IoT and cyber-physical systems. Currently, Raspberry Pi 4 is the latest version and it has built in Ethernet interface and WiFi module. Owing to its tiny size, negligible power consumption and low cost, Raspberry Pi 4 can ideally be used as a commercial iCDS for filtering of malicious traffic entering the SME networks. However, traffic filtering using Raspberry Pi device will not be a straight forward process since Raspberry Pi can use only one network interface at any given time even if it may have multiple network interface connections (i.e., internet traffic only goes through the particular interface connection). For traffic filtering purpose an IDS needs at least two network interfaces, one for incoming traffic and the other for outgoing traffic. When connected to an external network, incoming and outgoing internet traffic to and from the network only flows through the particular interface of the Raspberry Pi that is directly connected to the external network, be it the Ethernet interface or the WiFi interface. Even if multiple USB adapters are connected to the different available ports in the Raspberry Pi device, internet traffic from the external network will only flow through one of these connections and that is an issue with the use of Raspberry Pi as an iCDS.

Using some channel bonding technology, however, it is possible to channelize the network traffic to flow through two separate network interface connections, one for incoming traffic entering the Raspberry Pi device from external network and the other for outgoing traffic from the Raspberry Pi device [42]. This will need two network interface connections (e.g., network adaptors or network interface cards) in the Raspberry Pi 4.0 board and such connections can be in any form, like, the on-board Ethernet interface, on-board WiFi interface, USB Ethernet, and USB WiFi. For traffic filtering purpose, the Raspberry Pi device connected to a SME network, will require incoming and outgoing network traffic flowing through any of the two separate network interfaces. Based on such flow of network traffic, the following combinations are possible:

- *Option 1:* Network traffic entering the Raspberry Pi device through the on-board Ethernet interface and flowing out through the on-board WiFi interface
- *Option 2:* Network traffic entering the Raspberry Pi device through the on-board Ethernet interface and flowing out through USB Ethernet interface
- *Option 3:* Network traffic entering the Raspberry Pi device through the on-board WiFi interface and flowing out through the USB WiFi interface
- *Option 4:* Network traffic entering the Raspberry Pi device through the on-board WiFi interface and flowing out through the USB Ethernet interface

There are, however, few issues with the selection of the different interfaces on the Raspberry Pi 4.0 board for incoming and outgoing network traffic unless proper channel bonding is used. One such issue, for example, when choosing option 1 (on-board Ethernet interface for incoming traffic and WiFi interface for outgoing traffic), the configuration will face an issue with the assigned IP addresses for the two interfaces. Generally, individual IP addresses will be assigned to the Ethernet interface and WiFi interface, respectively, for incoming packets entering the Raspberry Pi board to identify the particular entry interface's IP address and filtered outgoing packets (i.e., network traffic packets leaving the Raspberry Pi board to enter the SME network gateway) to identify the exit interface's IP address. Since, network traffic flows through only one connection (at a time) on the Raspberry Pi board, in absence of channel bonding technique, all traffic will just identify the Ethernet interface's IP address and flow through that, whereas, the other WiFi interface connection will remain unnoticed. This implies, that traffic will not enter the gateway of the SME network. Also, in case of option 3, when choosing two WiFi interfaces for incoming and outgoing traffic there can be an issue with the Raspberry Pi board not properly identifying the particular WiFi interface after every reboot operation (i.e., which interface is for incoming and which one is for outgoing traffic). There is a possibility that Raspberry Pi may not identify the WiFi interfaces correctly when rebooted and that may lead to incorrect communication of the network traffic. Thus, from these discussions it can be concluded that it is feasible to use Raspberry Pi device to develop an iCDS but proper channel bonding needs to be used for tracking the network traffic entering and exiting the different interfaces on board. In the following sections we study the performance of different interfaces on the Raspberry Pi device in handling high volume of real traffic entering the device.

4. Experiment Results and Discussion

4.1. Performance of the Raspberry Pi Interfaces

This section discusses the preliminary experimental results of the proposed Raspberry Pi architecture (refer to Figure 3). As explained in the previous section, in this first phase of the work, the aim is to study the performance of the different interfaces on the Raspberry Pi 4.0 device when handling high volume of real unfiltered network traffic entering the device (i.e., incoming traffic). Identifying malicious traffic entering the Raspberry Pi 4.0 device and filtering them before entering the SME network is not done in this work. The different interfaces on the Raspberry Pi 4.0 device are the Ethernet interface, WiFi interface and external USB interface and in the experiment conducted, these three interfaces are exposed to real unfiltered network traffic entering the Pi board separately through each of these interfaces and are measured over a time interval. For example, traffic entering the Pi device through the Ethernet interface is measured from time t till t+1. Similarly, traffic entering through the WiFi interface and the USB interface are separately measured from t to t+1 time interval. Based on the incoming traffic, performance of each interface on the Raspberry Pi device is measured in terms of latency, and download and upload throughput. All the graphs in the next subsection depict results based on the average of multiple measurements.

4.2. Measurement of Latency

Latency is a significant aspect in determining the efficiency of any network interface. In the experiments conducted, latency of each interface on the Pi board (i.e., Ethernet, WiFi, and USB interfaces) is measured individually based on the incoming unfiltered real network traffic entering each interface separately over a time interval of t to t+1. Figure 3 depicts the latency comparison of the three interfaces on the Raspberry Pi 4.0 device based on separate measurements of the incoming network traffic.

As can be seen in Figure 4, the latency of the built-in WiFi interface on the Raspberry Pi device is considerably high in comparison to the latency values of the built-in Ethernet and USB Adapter interfaces. Apart from the fact that Ethernet (wired) connections usually offers better network speed and significantly lower latency compared to WiFi (wireless) connections, the other reason can be that the built-in WiFi on the Pi device has a single antenna and not a MIMO, so lower speed and more latency anyway. On the other hand, the

Ethernet interface also offers lower latency than the USB adapter interface.

4.3. Measurement of Download Traffic Throughput

Similar to latency, download and upload throughput of network traffic are other important aspects of determining the efficiency of a communication interface. The download traffic for each interface on the Raspberry Pi 4.0 device is measured separately over the t to t+1 time interval and the comparison results for the three interfaces are shown in Figure 5.

From the presented figures, it is evident that the throughput of the built-in Ethernet interface on the Pi device is significantly higher than the throughput of the WiFi and USB Adapter interfaces. Again, this can be related to the fact that Ethernet connections generally offer better network speed and thus better (download) throughout in comparison to WiFi and the USB connections. Performance of the in-built WiFi and USB interfaces look somewhat similar.

4.4. Measurement of Upload Traffic Throughput

Figure 6 compares the throughput of the upload traffic for the three network interfaces on the Raspberry Pi 4.0 device. The upload throughout performance of the built-in Ethernet interface has somewhat outperformed the other two interfaces. The USB Adapter on the Pi 4.0 device, unlike the Ethernet, shares a common bus and hence its bandwidth is also distributed among other ports, which is why it experiences some internal delays and has a low throughput.

4.5. DNA based Malware Detection Approach using Deep Auto Encoders

The experiment design consists of a classifier that is used to match the pattern of signatures from the repositories, i.e., HashSigns and DNASigns. Initially, 60,000 files are created with random hexadecimal values followed by creating two sets of input data through injecting 13500 synthetic signatures into those files. The first set $input_1$ is created by injecting malware signatures into 36,000 files to obtain a ratio of 60:40 between malicious and non-malicious contents, the second $input_2$ is created by interchanging the ratio between malicious and non-malicious contents, i.e., 40:60 between malicious and non-malicious contents. Two sets of experiments are performed using $input_1$ and $input_2$.

The design of the experiments is presented in Fig. 7. Two deep autoencoder (DAEs) with 3 layers are used for the experiments namely DAE_{DNA} and DAE_{Hash} based on the training data set. DAE_{DNA} is trained using the repository of signatures, DNASigns, created using proposed DNA-based approach, whereas, the second autoencoder DAE_{Hash} is trained using the repository, HashSigns, created with hexadecimal signatures. The total input files of 60,000 are sent to DAE_{DNA} and DAE_{Hash}, one at a time (same input) for performing classification in order to identify malicious and non-malicious content as shown in Fig. 7. Outputs obtained from the DAEs are tabulated and presented as experiment results. Since there are two repositories $input_1$ and $input_2$, the experiments are performed in two independent cycles. The technical details of the experiment along with the results are discussed in the next section.

The details of the input data set and signature repository along with the technicalities of DAE are presented as follows. Experiments are performed using the similar approach adopted for pattern recognition used in [43] for determining watermarks.

4.6. Outcome of DAE Experiment

The DAE used for the experiment consists of three layers apart from the softmax layer used for training. Since, the DAE learns from the patterns in the input, the softmax layer is used for validation. The DAEs are trained using the signature repositories for 500 epochs for the first and last layers and 1000 epochs for the middle layer. The signatures are divided based on their lengths and DAE nodes are adjusted to fit the size of the signatures. Since the aim of the training was to make DAE learn the patterns in the signatures, the changes in the input size is insignificant. The learning rate and momentum are varied from 0.4 to 0.6 and from 0.2 to 0.4 respectively with an interval of 0.1 for both the parameters. The individual average time for training and validation are recorded as 23.5 minutes and 19.2 minutes respectively. The testing times are averaged between 19 and 32 minutes for different runs. The differences are often due to hardware and software limitations and also based on the input (files) selected (in terms of length and number of epochs). Since the emphasis of the current research is not optimising the DAE, the variance in execution times can be ignored.

The input size for validation and testing is determined by splitting the content of the file. In this research, the malware file consists of data and malware signature in the form of hexadecimal values. Therefore, the file is treated as a continues hexadecimal values and it is split based on the size of the file. The DAE cannot be used with varying number of input nodes, hence the file size is fixed at 250 lines with 80 characters per line. Since the input is supplied as continues stream, the size of the lines and characters are insignificant and has no impact on the functionality of DAE. All the experiments are performed on Mac Book Pro M1 having 8GB RAM, 256 SSD, and 8 core GPU. For generating the synthetic malware, Python 3.6 is used with a default random

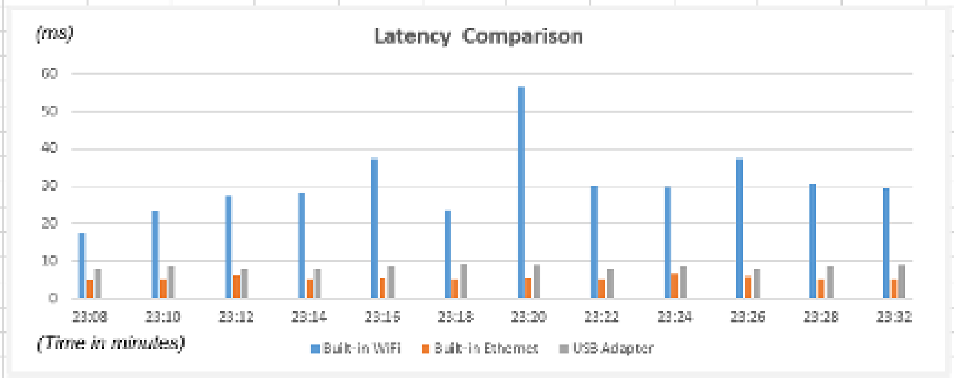

Figure 4. Latency comparison of Raspberry-pi interfaces when handling external traffic

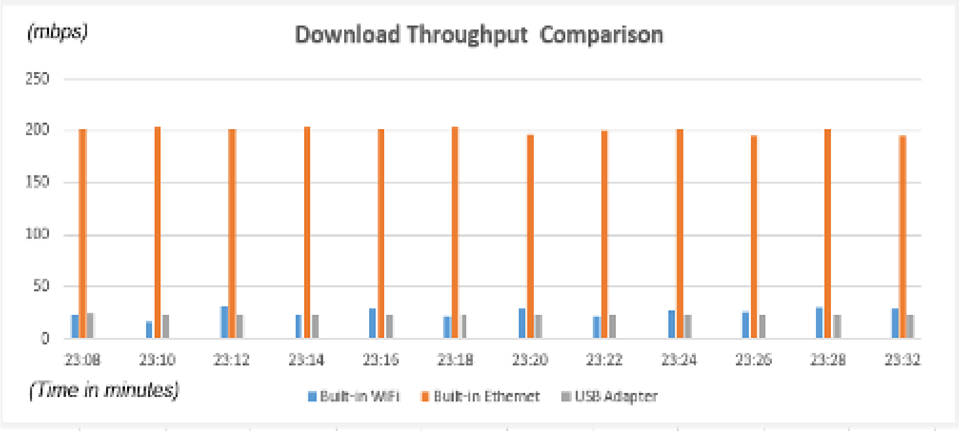

Figure 5. Traffic comparison for download throughput

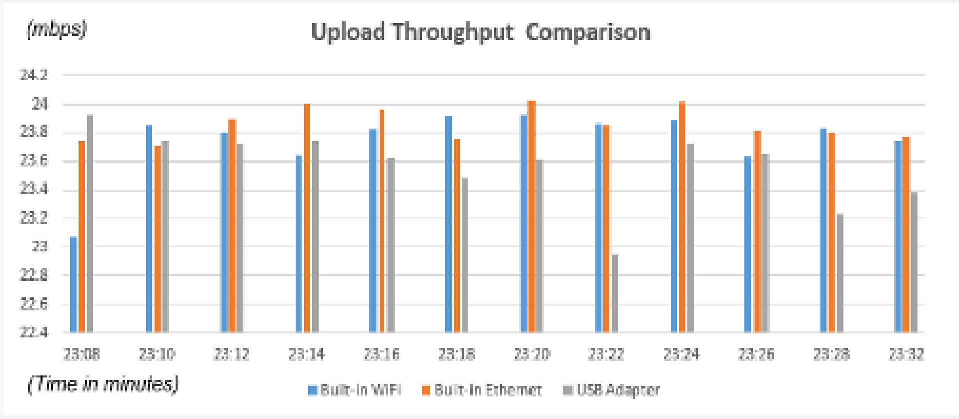

Figure 6. Traffic comparison for upload throughput

number generator that is modified to use only prime numbers. For DAE experiments, MATLAB 2020b (Mac version) is used.

Each experiment is performed 50 times. The validation and testing data is divided in 2:1 ratio due to DAE having no previous exposure to the files. A 3-fold cross validation is also performed on selected input files for further affirmation of results. The experiment results for the different input data sets along with the two different repositories are presented below in Table 1.

The results presented in Table 1 consists of training accuracies attained using DNASigns and HashSigns repositories. Along with the training results, validation and testing accuracies are also presented.

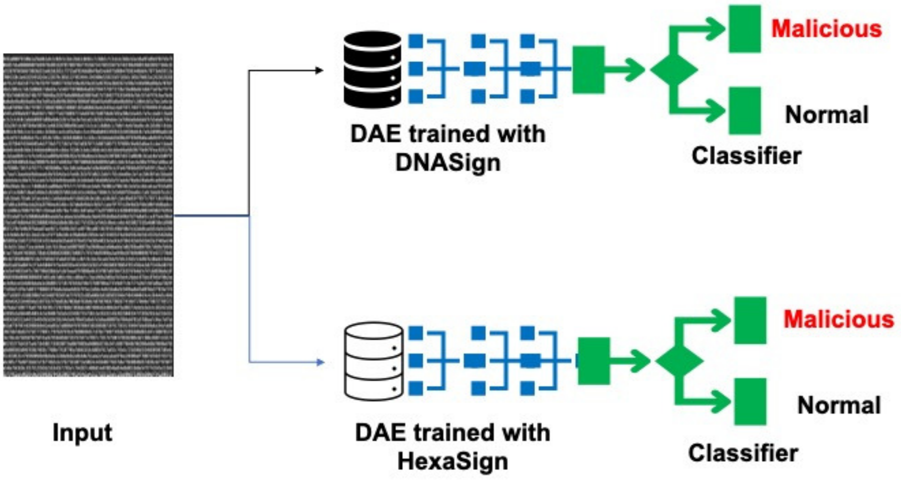

Figure 7. Experiment Design– The hexadecimal code of input (file) is sent as an input to two different deep autoencoder networks (DAE_{DNA} and DAE_{Hash}) trained using data from DNASigns and HashSigns repositories respectively. The DAE will be looking for pattern that might resemble malware signatures at least partly to identify and classify files with the potentially malicious content

Table 1. Experiment Results: Classification accuracies (training, testing and validation) with full set of DNA and HEXA signature repositories.

Type of Repository	malware ratio	Training %	Validation %	Testing %	RMS Error	T-Test Value
DNASigns	60:40	98.4	99	92.1	0.21	0.192
	40:60	99.2	99	94.8	0.38	0.263
HashSigns	60:40	99.6	99.3	72.6	0.64	0.211
	40:60	99.19	99.8	82.1	0.52	0.241

The first part of the results show that the proposed DNA-based approach is able to attain a classification accuracy of 92.1% for malware repository with 60:40 ratio, which is an improvement of 19.5% in comparison to the signature-based approach. The proposed DNA-based approach also performs better than the signature-based approach for the second set of experiments with 40:60 ratio of malicious and non-malicious data set with 94.8% of accuracy, which is an improvement of 12.7% over signature based approach (82.1%). It is noteworthy to observe that despite of better training accuracy for signature-based approach (with a minute deference of 0.2% for 60:40 and -0.1 for 40:60, respectively, in favour of signature-based approach) the proposed DNA-based approach attained better classification accuracy than signature-based approach. It is significant to observe that the difference of accuracies between the two experiments (i.e., with 60:40 and 40:60 ratios of malware and non-malware) the accuracy is reduced from 19.5% to 12.7%, a reduction of 6.8%. The reason for this reduction can possibly be attributed to the differences in the ratios between malicious and non-malicious inputs.

5. Known Limitations and Impact

The implementation of iCDS have some know shortcomings related to software and hardware capability. The experimental evaluation clearly indicates the capability of using Raspberry Pi as IDS. The assertion of the research on producing low cost iCDS is successful considering the hardware costs.

Raspberry Pi or similar devices have issues with Heat particularly when expected to run for long hours. Any IDS is expected to be online continuously which would incur a lot of heating for the devices. The research tried to look into this aspect and found that an external attachment is required to keep the heat levels low. However, due to time constraint, the research could not perform a continues test for weeks / days together.

In case of accidental damage, majority of the parts can be replaced as the modules of Raspberry Pi are easily available. Also, the crucial modules like wifi and networking modules can be replaced by external modules if required. Moreover, a parallel backup device could be attached to the network as a recovery module for fail-over.

In proposed approach, the DNA based malware detection requires a cloud based environment for feature extraction and comparison. The experiments in this publications are conducted in an ideal environment without time limits on upload the traffic in real-time. Though the upload and download throughput are tested for Raspberry Pi, the impact on upload for feature extraction may impact the performance.

5.1. Impact of the Research

Ongoing research on using low hardware and low cost ids devices has inspired to undertake this research. As mentioned previously, there was no formal research on using Raspberry Pi or similar type of devices for IDS. This results of the research provides a potential encouragement for other researchers to perform similar type of research and provide a low cost solution for cyber security. The research, for the first time provides various parameters and modules that needs to be tested for intelligent IDS which will inspire the research community to consider ML for low cost IDS devices.

On the other hand, there is a high chance of mimicking this experiments with some incompatible devices and produce an inefficient device. Also, not every ML approach is efficient and the results can be reproduced. Hence, the research might create potentially a negative impact on efficiency of the proposed research.

Since this research encourages low-cost and computation savvy devices, the research community working on IDS might get influenced to look into such devices for critical systems like healthcare. Using iCDS for critical systems cannot be evaluated at this stage due to limitations of this research.

6. Conclusion and Future Work

Primarily, the work presented in this paper has a two-fold focus: (a) to explore the feasibility of using Raspberry Pi device to develop a low-cost intelligent cyber-defense system or iCDS for commercial SME networks, and (b) to study if the hardware components present in the latest Raspberry Pi devices are capable and compatible enough to support the use of Raspberry Pi-based iCDS for SMEs. Based on the detailed discussions presented in the paper, it can be concluded that it is feasible to use Raspberry Pi device to develop a low-cost iCDS as an alternative to the traditional rule-based IDSs in use. Moreover, from the experimental results as discussed in Section IV, it is evident that the different interfaces on the Raspberry Pi 4.0 device, e.g., built-in Ethernet (wired) connection, WiFi and the external USB Adapter, studied in this research are capable of handling high volumes of traffic entering the Raspberry Pi device from outside networks. The evaluations also showed that in terms of network performance comparison carried out based on parameters, like, latency, downward traffic throughout and upward traffic throughput, the built-in Ethernet network interface has outperformed the other two interfaces and thus can be an ideal choice to use for handling external traffic.

On the other hand, from the experiment results of DNA based approach, it can be concluded that the DAE trained with the proposed DNA-based malware signatures was able to achieve better results compared to the DAE trained with traditional signature based data set. For data set with the malicious and non-malicious ratio of 60:40, the proposed DNA-based approach provides a classification accuracy of 92.1%, which is 19.5% better than the traditional signature-based approach in identifying malware variants despite of lower training accuracy. Similarly, for the 40:60 ratio of malicious and non-malicious data set, the proposed approach performs 12.7% better than the traditional approach. Also, DAE trained with the data set generated using proposed DNA-based approach was able to achieve better accuracy rates for partial signatures.

Acknowledgement

This research was funded by Manukau Institute of Technology

References

[1] Belanger, F., Hiller, J.S. and Smith, W.J. (2002) Trustworthiness in electronic commerce: the role of privacy, security, and site attributes. *The journal of strategic Information Systems* **11**(3-4): 245–270.

[2] Lu, Z., Qu, G. and Liu, Z. (2018) A survey on recent advances in vehicular network security, trust, and privacy. *IEEE Transactions on Intelligent Transportation Systems* **20**(2): 760–776.

[3] Tirumala, S.S., Sathu, H. and Naidu, V. (2015) Analysis and prevention of account hijacking based incidents in cloud environment. In *2015 international Conference on Information Technology (ICIT)* (IEEE): 124–129.

[4] Alnahari, W. and Quasim, M.T. (2021) Privacy concerns, iot devices and attacks in smart cities. In *2021 International Congress of Advanced Technology and Engineering (ICOTEN)* (IEEE): 1–5.

[5] Khraisat, A., Gondal, I., Vamplew, P. and Kamruzzaman, J. (2019) Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity* **2**(1): 1–22.

[6] Ali, B. and Awad, A.I. (2018) Cyber and physical security vulnerability assessment for iot-based smart homes. *sensors* **18**(3): 817.

[7] Granjal, J., Monteiro, E. and Silva, J.S. (2015) Security for the internet of things: a survey of existing protocols

and open research issues. *IEEE Communications Surveys & Tutorials* **17**(3): 1294–1312.

[8] Aslan, Ö.A. and Samet, R. (2020) A comprehensive review on malware detection approaches. *IEEE Access* **8**: 6249–6271.

[9] Jin, B., Choi, J., Kim, H. and Hong, J.B. (2021) Fumvar: a practical framework for generating f ully-working and u nseen m alware var iants. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*: 1656–1663.

[10] James, A.V. and Sabitha, S. (2021) Malware attacks: A survey on mitigation measures. In *Second International Conference on Networks and Advances in Computational Technologies* (Springer): 1–11.

[11] Scott, J. (2017) Signature based malware detection is dead. *Institute for Critical Infrastructure Technology* .

[12] Tirumala, S.S., Valluri, M.R. and Nanadigam, D. (2020) Evaluation of feature and signature based training approaches for malware classification using autoencoders. In *2020 International Conference on COMmunication Systems NETworkS (COMSNETS)*: 1–5. doi:10.1109/COMSNETS48256.2020.9027373.

[13] Usman, N., Usman, S., Khan, F., Jan, M.A., Sajid, A., Alazab, M. and Watters, P. (2021) Intelligent dynamic malware detection using machine learning in ip reputation for forensics data analytics. *Future Generation Computer Systems* **118**: 124–141.

[14] Choi, Y.H., Han, B.J., Bae, B.C., Oh, H.G. and Sohn, K.W. (2012) Toward extracting malware features for classification using static and dynamic analysis. In *2012 8th International Conference on Computing and Networking Technology (INC, ICCIS and ICMIC)* (IEEE): 126–129.

[15] Tirumala, S.S., Valluri, M.R. and Nanadigam, D. (2020) Evaluation of feature and signature based training approaches for malware classification using autoencoders. In *2020 International Conference on COMmunication Systems NETworkS (COMSNETS)*: 1–5. doi:10.1109/COMSNETS48256.2020.9027373.

[16] Minoli, D. (2017) Iot applications to smart campuses and a case study. *EAI Endorsed Transactions on Smart Cities* **2**(5): e4–e4.

[17] Banga, M., Patil, M. *et al.* (2020) Secured authentication systems for internet of things. *EAI Endorsed Transactions on Smart Cities* **20**(11).

[18] Kolias, C., Kambourakis, G., Stavrou, A. and Voas, J. (2017) Ddos in the iot: Mirai and other botnets. *Computer* **50**(7): 80–84.

[19] Lu, D., Huang, D., Walenstein, A. and Medhi, D. (2017) A secure microservice framework for iot. In *2017 IEEE Symposium on Service-Oriented System Engineering (SOSE)* (IEEE): 9–18.

[20] Pahl, M.O., Aubet, F.X. and Liebald, S. (2018) Graph-based iot microservice security. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium* (IEEE): 1–3.

[21] Gupta, N., Naik, V. and Sengupta, S. (2017) A firewall for internet of things. In *2017 9th International Conference on Communication Systems and Networks (COMSNETS)* (IEEE): 411–412.

[22] Taib, A.M., Zabri, M.T., Radzi, N.A.M. and Kadir, E.A. (2020) Netguard: Securing network environment using integrated openvpn, pi-hole, and ids on raspberry pi. In *Charting the Sustainable Future of ASEAN in Science and Technology* (Springer), 97–110.

[23] Jesús, R.L.J., Cristhian, P.V.O., René, R.G.M. and Heberto, F.M. (2019) How to improve the iot security implementing ids/ips tool using raspberry pi 3b. *Editorial Preface From the Desk of Managing Editor...* **10**(9).

[24] Tripathi, S. and Kumar, R. (2018) Raspberry pi as an intrusion detection system, a honeypot and a packet analyzer. In *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)* (IEEE): 80–85.

[25] Soe, Y.N., Feng, Y., Santosa, P.I., Hartanto, R. and Sakurai, K. (2019) Implementing lightweight iot-ids on raspberry pi using correlation-based feature selection and its performance evaluation. In *International Conference on Advanced Information Networking and Applications* (Springer): 458–469.

[26] Sumanth, R. and Bhanu, K. (2020) Raspberry pi based intrusion detection system using k-means clustering algorithm. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)* (IEEE): 221–229.

[27] Malikovich, K.M., Rajaboevich, G.S. and Karamatovich, Y.B. (2019) Method of constucting packet filtering rules. In *2019 International Conference on Information Science and Communications Technologies (ICISCT)* (IEEE): 1–4.

[28] Meng, W., Li, W. and Kwok, L.F. (2017) Towards effective trust-based packet filtering in collaborative network environments. *IEEE Transactions on Network and Service Management* **14**(1): 233–245.

[29] Serdechnyi, V., Barkovska, O., Rosinskiy, D., Axak, N. and Korablyov, M. (2019) Model of the internet traffic filtering system to ensure safe web surfing. In *International Scientific Conference "Intellectual Systems of Decision Making and Problem of Computational Intelligence"* (Springer): 133–147.

[30] Yu, B., Fang, Y., Yang, Q., Tang, Y. and Liu, L. (2018) A survey of malware behavior description and analysis. *Frontiers of Information Technology & Electronic Engineering* **19**(5): 583–603.

[31] Bulygin, M. and Namiot, D. (2021) Anomaly detection method for aggregated cellular operator data. In *2021 28th Conference of Open Innovations Association (FRUCT)* (IEEE): 42–48.

[32] Pang, G., Shen, C., Cao, L. and Hengel, A.V.D. (2021) Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)* **54**(2): 1–38.

[33] Sihwail, R., Omar, K. and Ariffin, K.A.Z. (2018) A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis. *International Journal on Advanced Science, Engineering and Information Technology* **8**(4-2): 1662.

[34] Ijaz, M., Durad, M.H. and Ismail, M. (2019) Static and dynamic malware analysis using machine learning. In *2019 16th International bhurban conference on applied sciences and technology (IBCAST)* (IEEE): 687–691.

[35] Ding, J., Chen, Z., Zhao, Y., Su, H., Guo, Y. and Sun, E. (2017) Mget: malware gene-based malware dynamic analyses. In *Proceedings of the 2017 International Conference on Cryptography, Security and Privacy*: 96–101.

[36] Naidu, V.J. (2018) *Identifying Polymorphic Malware Variants Using Biosequence Analysis Techniques*. Ph.D. thesis, Auckland University of Technology.

[37] Siddiqui, M., Wang, M.C. and Lee, J. (2008) A survey of data mining techniques for malware detection using file features. In *Proceedings of the 46th annual southeast regional conference on xx*: 509–510.

[38] Jang, E.G., Lee, S.J. and Lee, J.I. (2014) A study on similarity comparison for file dna-based metamorphic malware detection. *Journal of the Korea Society of Computer and Information* **19**(1): 85–94.

[39] Chen, L., Xia, C., Lei, S. and Wang, T. (2021) Detection, traceability, and propagation of mobile malware threats. *IEEE Access* **9**: 14576–14598.

[40] Karbab, E.B., Debbabi, M. and Mouheb, D. (2016) Fingerprinting android packaging: Generating dnas for malware detection. *Digital Investigation* **18**: S33–S45.

[41] Han, B.J., Choi, Y.H. and Bae, B.C. (2013) Generating malware dna to classify the similar malwares. *Journal of the Korea Institute of Information Security & Cryptology* **23**(4): 679–694.

[42] Tirumala, S.S., Nepal, N. and Ray, S.K. (2022) Raspberry pi-based intelligent cyber defense systems for smes: An exploratory study. In *International Summit Smart City 360°* (Springer): 3–14.

[43] Tirumala, S., Jamil, N. and Malik, M.A. (2018) A deep neural network approach for classification of watermarked and non-watermarked images. In *International Conference on Intelligent Technologies and Applications* (Springer): 779–784.