IOT BOTNET DETECTION

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF A DEGREE OF

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

KARAMVEER

(2K20/CSE/10)

Under the supervision of

DR. RAJESH KUMAR YADAV

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

May, 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College Of engineering)

Bawana Road, Delhi-110042

DECLARATION

I, the undersigned, therefore certify that the project report **IOT BOTNET DETECTION** is based on my own work completed during our studies under the direction of **DR. RAJESH KUMAR YADAV**, Assistant Professor, DTU.

I affirm that the assertions I've made and the conclusions I've reached are the results of my research. I also confirm that –

I. I wrote the report under my supervisor's supervision.

II. The work has not been submitted to any other university, in India or abroad, for a degree, diploma, or certificate.

III. We followed the university's report-writing regulations.

IV. When we utilised data, theoretical analysis, or text from other sources, we cited them and provided their contact information in the references.

Karamveer

Place: Delhi

Karamveer

2K20/CSE/10

M.TECH CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College Of engineering)

Bawana Road, Delhi-110042

CERTIFICATE

This is to authenticate that the research project entitled, **IOT BOTNET DETECTION**, presented to the Department of Computer Science and Engineering, Delhi Technological University (DTU), in partial fulfillment for the award of the degree of **Master of Technology** in Computer Science and Engineering, is a document of legitimate work completed by **Mr. Karamveer**, Roll no. **2K20/CSE/10**, under my supervision and direction.

All of the assistance he got from various sources has been recognized.

This report has not been submitted anywhere else for the granting of any other degree.

Place: Delhi

Dr. Rajesh Kumar Yadav

Assistant Professor

Delhi Technological University

Date: 26th May, 2022

ABSTRACT

The Internet consists of multiple interconnected systems/networks, one of which being the “Internet of Things”. In spite of their flexibility, numerous IoT devices/gadgets are technically weak in terms of security, which makes them an ideal target for a variety of security breaches, including botnet assaults. IoT applications in the smart city are currently being targeted by advanced persistent threats (APT). Botnets are a piece of malware that permits hackers to take control of several systems and carry out destructive operations. IoT-based botnet assaults have become increasingly common as a result of the development of IoT gadgets, which are more readily hacked than desktop PCs. To combat this new danger, advanced approaches for identifying attacks initiated from infected IoT devices and distinguishing between day and milliseconds duration assaults must be developed.

This study aimed to find, assess, and present a comprehensive overview of experimental works on IoT botnet detection research. The identification methods used to identify IoT botnets, their stages, and the botnet stealth strategies were all investigated in this study. The writers examined the nominated study as well as the major approaches used in it. The authors analyzed the botnet stages when detection is done and categorized the detection methods depending on the strategies utilized. The authors examined current research gaps and proposed future research topics as a consequence of this investigation and proposed a network-based anomalous detector that leverages deep learning to identify aberrant network traffic flowing from exploited IoT nodes by extracting network behavioral snapshots. On the UNSW dataset with a slew of neural network architectures and hidden layers, the suggested model combining CNN and LSTM has been trained and assessed. To test our strategy, I employed a dataset of various commercial IoT nodes infiltrated with

Mirai and BASHLITE, two popular IoT botnets. The results of our tests showed that our suggested strategy could correctly and quickly detect assaults as they were launched from hacked IoT nodes that were members of a botnet.

ACKNOWLEDGMENT

I'd like to convey my sincere gratitude to my supervisor, **DR. Rajesh Kumar Yadav**, Assistant Professor (DTU), for his unwavering support and advice. He was a constant source of inspiration and was always eager to help in any way he could during the study effort. Without continual support, direction, and help, this effort would not have been feasible. Their patience, expertise, and resourcefulness are qualities I will always aspire to achieve.

Karamveer

Place: Delhi

Karamveer

2K20/CSE/10

MOTIVATION

Because the internet is utilized in almost every area in today's world, internet security is just as crucial as it is for other things. Because I work in the area of network security and server load testing, I came up with the concept of building a model that can distinguish between legitimate traffic and bot/crawler traffic and protect web servers from their negative consequences, such as DDOS assaults. DDOS assaults are on the rise these days, and they are costing businesses more money. Competitors may also use DDOS attacks to shut down a competitor's service, affecting their income. Further considering this issue, I performed some research on CDN providers and attempted some reverse engineering on a prominent CDN provider's security. I discovered that they use some unique browser challenges that are automated and identify actual browsers without any human participation. They don't even ask for a captcha or anything else related. This reverse engineering inspired me to try to duplicate it. Also, because the challenges are timed, you cannot respond to them before or after the deadline. However, if time is not an issue, you can manually solve the task and respond afterwards. After implementing this approach, I discovered that it is not fail-safe, therefore I looked at machine learning-based ways for reliably identifying botnet traffic by training an anomaly detector. The threat of an IoT botnet necessitates effective defence and response approaches and strategies. By looking over the current studies on IoT botnet detection, we discovered that there is a dearth of in-depth research for IoT botnet identification approaches. The majority of researchers concentrated on detecting botnets using network traffic after they had launched assaults on their targets. As a result, the research is relatively immature yet has a lot of potential. The creation of a botnet occurs in numerous stages and detection techniques vary depending on which phases are targeted. Because each phase may exhibit distinct actions a thorough

examination of each phase's detection strategies is necessary. However there has never been a complete and detailed assessment of IoT botnet detection that takes into consideration botnet stages.

CONTENTS

DECLARATION i

CERTIFICATE ii

ABSTRACT iii

ACKNOWLEDGMENT v

MOTIVATION vi

CONTENTS viii

LIST OF FIGURES xi

LIST OF TABLES xii

LIST OF ABBREVIATIONS xiii

Chapter 1: Introduction 1

1.1 Overview: 1

1.2 Problem Formulation 3

1.3 What are Botnets? 3

1.4 How Botnet Works? 4

1.5 Stages of Botnet 4

1.5.1 Propagation 5

1.5.2 Infection 6

1.5.3 Command and Control 6

1.5.4 Attacks 6

1.6 Stealth Tactics of Botnet 7

1.6.1 Modifying Registry Values: 7

1.6.2 Task Scheduler: 7

1.6.3 Process Hollowing: 8

1.6.4 DLL Injection: 9

1.6.5 File-less Malware Approach: 9

1.7 Objective of Project 10

Chapter 2: Related Work **12**

2.1 Literature Review 12

2.2 Publicly Available Datasets 15

2.2.1 MedBIoT 15

2.2.2 IoT-23 15

2.2.3 UNSW-NB15: 15

2.2.4 N-BaIoT: 15

2.3 Detection Methods 16

2.3.1 Supervised Learning Based: 16

2.3.2 Un-supervised Learning-Based: 17

2.3.3 Deep Learning-Based: 17

2.3.4 Blockchain-Based: 19

2.3.5 SDN-Based (Software-Defined-Networking): 19

2.4 Analysis 19

Chapter 3: Technology Used **23**

3.1 Deep Learning 23

3.2 Convolutional Neural Network 23

3.2.1: Convolutional Layer: 23

3.2.2: Polling Layer: 25

3.2.3: Fully-Connected-Layer: 25

3.3 LSTM (Long-Short-Term-Memory) 26

Chapter 4: Methodology **29**

4.1 Dataset Used 29

4.2 Proposed system 31

4.2.1 Data Collection 31

4.2.2 Feature Extraction 31

4.2.3 Training and Testing 32

Chapter 5: Experimental Results **33**

5.1 Evaluation Metrics 33

5.2 Results 33

5.3 Outputs 34
5.3.1 Performance Evaluation 34
5.3.2 Training Outcome 35
5.3.3 Confusion Matrix 35
5.3.4 Accuracy vs Epoch 37
5.3.5 Model Loss 37
Chapter 6: Conclusion **38**
References **39**
List of Publications **46**

LIST OF FIGURES

Fig 1. Life cycle of Botnet 4
Fig 2. Botnet Structure 5
Fig 3. Retadup's deobfuscated code for stealth 8
Fig 4. Retadup's scheduled tasks 9
Fig 5. DLL Injection 10
Fig 6. Detection Methods 18
Fig 7. Various DL Methods 24
Fig 8. CNN Layers 25
Fig 9. Polling Layer 26
Fig 10. Fully Connected Layer 27
Fig 11. RNN Architecture 27
Fig 12. LSTM Architecture 28
Fig 13. Selected Features 29
Fig 14. Dataset Snapshot 30
Fig 15. Distribution of Data 30
Fig 16. CNN-LSTM Method 31
Fig 17. Equations 33
Fig 18. Performance Comparison 35
Fig 19. Confusion Matrix 35
Fig 20. Training Outcome 36
Fig 21. Accuracy-Epoch Graph 37
Fig 22. Model Loss 37

LIST OF TABLES

Table I: Previous Studies on IoT Botnet …….. 13

Table II: Comparison of various Detection techniques ……. 20

LIST OF ABBREVIATIONS

ML	Machine Learning
IoT	Internet of Things
SVM	Support vector machine
NB	Naïve Bayes
ANN	Artificial Neural Networks
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory
FC	Fully Connected Layer
DDOS	Distributed denial of service
CDN	Content Delivery Network
IDS	Intrusion Detection System
FPR	False Positive Rate
TPR	True Positive Rate

Chapter 1

Introduction

1.1 Overview:

SMART cities aim to increase the productivity, livability, and environmental sustainability of the city. They do this by implementing innovative technologies. In order to increase city operations' efficiency and provide residents with IoT services, smart cities' IoT apps mix ICT and numerous physical gadgets. Using the IoT, objects such as automobiles and home appliances may speak with one another and share data in real time by being part of a network of sensors, actuators, and connectivity. It is predicted that by 2023 there will be 30.1 billion IoT devices on the market; this estimate includes smartphones, tablets, and personal computers. By 2025, the market value of IoT devices is estimated to reach the mark of trillions [1]. People all across the world are buying, installing, and using an increasing number of smart devices, raising new worries about personal and societal security. Even in "smart cities," additional IoT devices are needed to increase energy efficiency, minimize traffic congestion, and improve the quality of the water. Smart city IoT networks remain vulnerable to more sophisticated attacks like botnets, despite the growing number of IoT devices.

This study focuses on botnets especially. The term "botnet assaults" refers to attacks on a victim's resources, The botnets have expanded to large networks of numerous machines in order to paralyze a target [2]. This means that the botmaster can simultaneously direct the bots in his or her network to carry out a coordinated crime. More and more botnets are popping up, and they have the potential to cause havoc both online and off. As a result, the botmaster is able to perform large-scale tasks

that were previously impractical with malware. The remote botmaster has the ability to send attack commands to infected machines on the fly in order to achieve their objectives. For example, the Mirai attacks used botnets to get access to the default username and password combinations. DDoS attacks employing tens of thousands of hacked IoT devices were used to shut down major websites by the well-known Internet of Things botnet. Botnet attacks against smart city infrastructure require an all-encompassing defense-in-depth strategy. Cities can employ a wide range of strategies, including firewalls, intruder detection systems, access control and authenticating systems. An attack that takes use of Internet-of-Things (IoT) vulnerabilities may be prevented from propagating throughout the Internet if these technologies were used. If a hacked workstation is analyzing the network for security flaws, an IDS can detect these attacks. A compromised device or combination of compromised devices, on the other hand, is more likely to lead to detection. City network managers can detect any anomalous traffic patterns using monitoring and diagnostic techniques.

It is common practice for many of the commercially available cyber security systems to employ several detection and prevention techniques such as heuristic and statistical approaches in addition to threshold-based methods. Detection methods are excellent at spotting attacks and botnets that have been previously identified. Because of this, they are utterly unable to recognize new attack or botnet versions. To combat these new breeds of botnets, we'll require domain experts. This has resulted in a large increase in the cost of maintaining the present systems as well as the potential that new types of botnets could undermine them, resulting in major financial losses for countless businesses. [3], [4], [5], [6] are examples of self-learning systems. Machine learning models must be pre-trained on a frequent basis to stay up with the ever-changing nature of botnets. Engineers use feature

engineering to design a botnet detection model with the best features. Automated feature extraction using deep learning models does not necessitate human intervention.

1.2 Problem Formulation

The Internet consists of multiple interconnected systems/networks, one of which being the Internet of Things. In spite of their flexibility, numerous IoT devices/gadgets are technically weak in terms of security, which makes them a ideal target for a variety of security breaches, including botnet assaults. IoT applications in the smart city are currently being targeted by advanced persistent threats (APT). Botnets are a piece of malware that permits hackers to take control of several systems and carry out destructive operations. IoT-based botnet assaults have become increasingly common as a result of the development of IoT gadgets, which are more readily hacked than desktop PCs. To combat this new danger, advanced approaches for identifying attacks initiated from infected IoT devices and distinguishing between day and milliseconds duration assaults must be developed.

1.3 What are Botnets?

Botnets are a form of network-based assault that aims to turn several computers into "zombie" systems at the same moment. A botnet's life cycle is depicted in the diagram below in Fig 1. These "zombie" machines are subsequently exploited for malevolent purposes such as identity fraud, network assaults (DDoS), phishing, spamming, and impersonating domain names.

Botnets like Mirai are generally built in multiple operational phases [49], including transmission, infiltration, C&C interaction, and offensive execution.

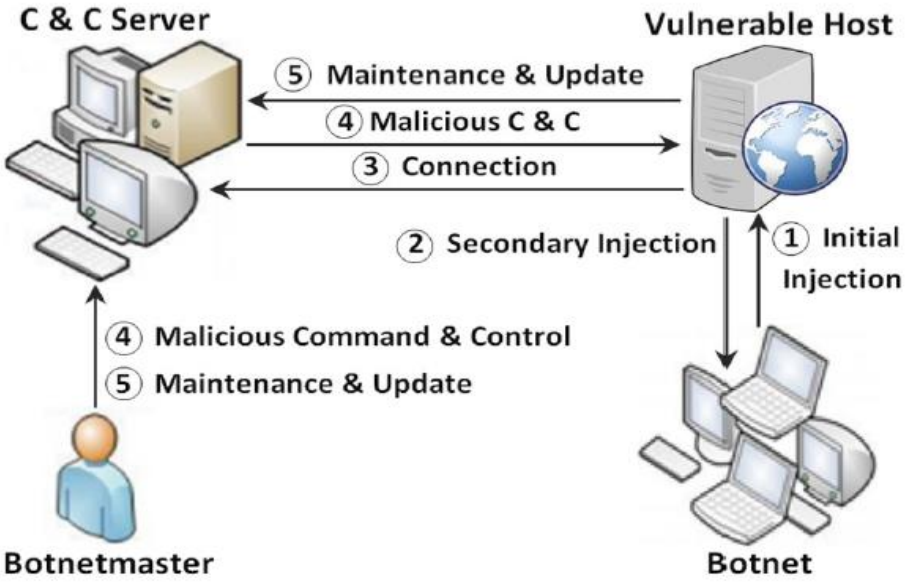


Fig 1. Life cycle of botnet
Source: [49]

1.4 Working of bots

The Botnet is a cluster of devices controlled by the botmasters and used for nefarious purposes such as theft, spam, DDoS assaults, and surveillance. Botnets are created in large part by worms that propagate over networks and connect machines to a central (C&C) command and control. The botmasters - malevolent people who dominated botnets - aspire to gather tens of billions of network nodes so that they may attack targets whenever they want. Fig 2 depicts botnet operations. There are multiple operational phases in a botnet, including propagation, infiltration, C&C interaction, and offensive execution. The next section goes over each of these steps.

1.5 Stages of Botnet

The Infection, expansion, communication, and assault launch are all part of the botnet's life cycle. The Botnet first infects a machine by abusing known or unknown

vulnerabilities. It monitors the network for other possible hosts once it has infected the system. Botnet begins interacting with the command and control center through encrypted HTTP, IRC, or other network protocols after infection. When the botmaster issues a command to begin an assault on a specific target, the botnet obeys and performs the assigned mission, such as DDOS, mining, or data breach.

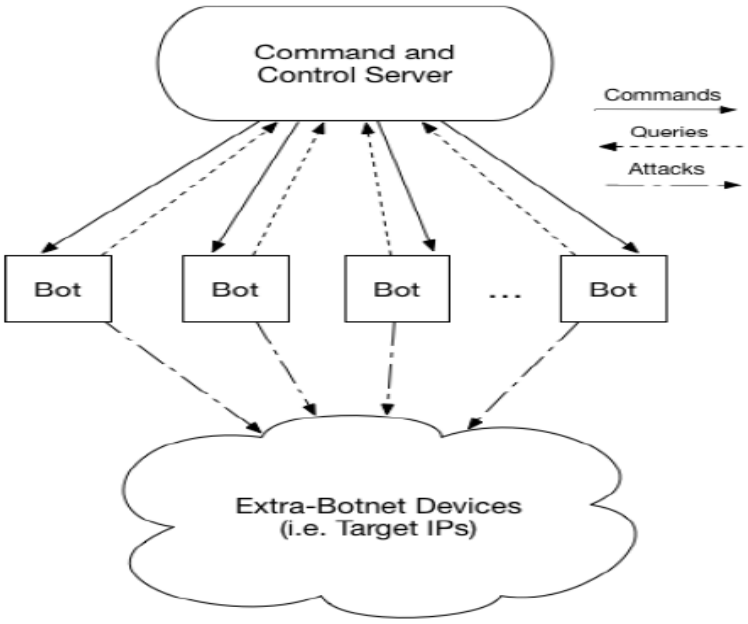


Fig 2. Botnet Structure

Source: [49]

1.5.1 Propagation

The botnets' primary goal is to propagate as many nodes as possible, resulting in more effective assaults. The main issue that the botnet developer faces is the propagation of the botnet without identification. Certain threats propagate by social engineering, such as persuading a user to click a virus email that leads to a malware webpage, while others may be found on file-sharing websites. Human spread limits the rate of germination, despite the fact that it is a successful strategy. Without

human involvement, a more effective spread might be achieved by scanning for unprotected IP addresses and attacking them. The first malware was aimed at any device. It's used in random number generators to produce a list of IP addresses and attempt to exploit them.

1.5.2 Infection

Typically, the attackers target specific flaws in the equipment. This may be anything, and in most cases, zero-day vulnerability, such as servers and PCs, which are common botnet targets, have fairly adequate security. In the context of Mirai and BASHLITE, security implementation is not safe due to the usage of default credentials, which has been attacked. Intruders will install binaries on the target network elements after they have gained access to them.

1.5.3 Command and Control

Command & Control are two terms that are used interchangeably. Botnets are meaningless unless the cyber attackers can control them and carry out the commands. Bot connects to the centralised C&C server in most cases. This is the situation with Mirai. The addresses of such servers were transferred sparingly at the start of the botnet to make it easier for bots to join and make it less probable that they would become orphans. However, the longer the IP stays statics, the easier it will be for law authorities to locate and shut down the server C&C. Botmasters frequently change servers, either because they are too common to be accurately monitored – or because they are worth shutting down – requiring the bot to be more successful in connecting to and updating C&C locations.

1.5.4 Attacks

Botnet has been used to achieve a broad range of horrible goals. One of the applications is to launch a DDoS assault. DDoS cyberattacks are on rise, according

to the security community. Previously, adversaries who wanted to carry out DDoS assaults had to establish their own bots. A large DDoS assault, similar to the one observed on Dyn, has happened.

The fundamental question is why the bots are undetectable by the victim machines? Bots employ stealth tactics to elude the notice of users on victim systems, which is a straightforward answer to this question. In the next section, certain bot stealth strategies are briefly addressed.

1.6 Stealth Tactics of Botnet

One of several key components of the IoT botnet is in charge of maintaining persistence on victim devices in order to avoid detection by users. Some of the strategies are explained in this section.

1.6.1 Modifying Registry Values:

[16] found that the Retadup Botnet obtains persistence and bypasses UAC either by generating a registry value in **"HKCU/ Software/Microsoft/Windows/CurrentVersion/Run"** that runs the bot at boot time or by creating a registry value in **"HKCU/Software/Classes/mscfile/shell/open/command"** and then running windows eventvwr.exe system file which gives the bot admin-level privileges. To ensure persistence on the target PC with windows 10, Retadup leverages the registry value **"HKCU/Software/Classes/ms-settings/shell/open/command"** on Windows 10. Fig 3 depicts retadup's deobfuscated code for stealth.

1.6.2 Task Scheduler:

Task Scheduler function makes it possible to run automated operations on a specific machine for legal administrative needs. Retadup uses the schtasks.exe application to

generate the scheduled task which is set to run every minute according to [16]. Hardcoded registry values are common in AutoIt Retadup variations whereas AutoHotkey variants employ both registry variables and scheduled activities with randomly created names. Fig 4 depicts the scheduler used in retadup botnet.

```
Func itvsltytihsxsrsoptmlwkwkhmwiu()
    If StringInStr(@OSVersion, "7") OR StringInStr(@OSVersion, "8") Then
        If NOT Execute("IsAdmin()") Then
            RegWrite("HKCU\\Software\\Classes\\mscfile\\shell\\open\\command", "",
                     "REG_SZ", @AutoItExe)
            ShellExecute("eventvwr")
            Exit
        EndIf
    ElseIf StringInStr(@OSVersion, "10") Then
        If NOT Execute("IsAdmin()") Then
            DllCall("kernel32.dll", "boolean", "Wow64EnableWow64FsRedirection",
                     "boolean", "0")
            RegWrite("HKCU\\Software\\Classes\\ms-settings\\shell\\open\\command",
                     "DelegateExecute", "REG_SZ", "Null")
            RegWrite("HKCU\\Software\\Classes\\ms-settings\\shell\\open\\command",
                     "REG_SZ", @AutoItExe)
            ShellExecute("fodhelper")
            Exit
        EndIf
    EndIf
EndFunc
```

Fig 3. Retadup's deobfuscated code for stealth [16].

Source: Adapted from [16]

1.6.3 Process Hollowing:

Botnets frequently employ Process Hollowing to gain persistence by transforming a genuine programme into a malicious one. Botnets replace the code portion of other running programmes' Portable executables in memory with their malicious code. Some instances of exploiting process hollowing approach and avoiding defences by leveraging functions exposed from unhooked copy of ntdll files are agent tesla and miner payload in retadup [16].

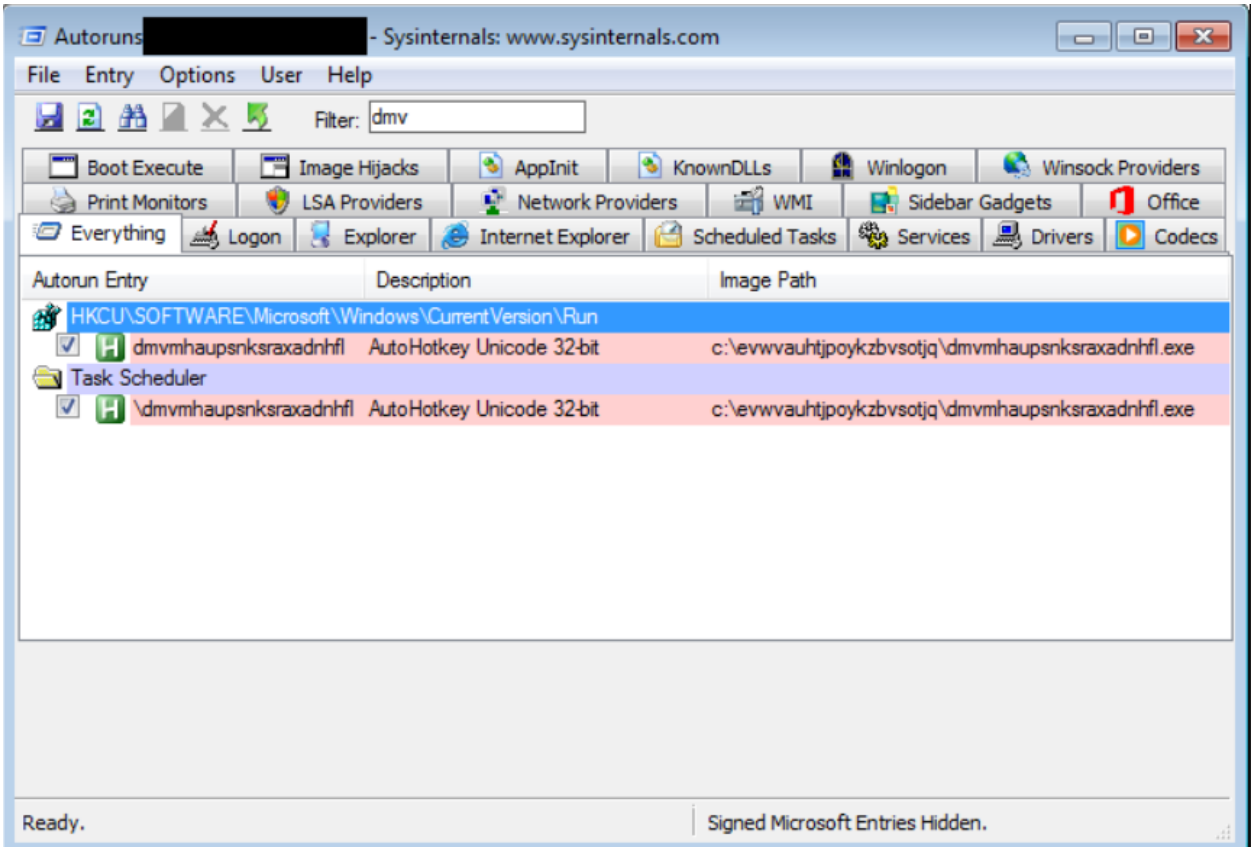


Fig 4. Retadup's scheduled task [16].

Source: Adapted from [16]

1.6.4 DLL Injection:

Botnets often implant DLLs into processes to circumvent process based protections and possibly raise privileges. DLL injection can run malicious scripts in another active process' address space. It is achieved by putting the DLL path in target process' virtual address space by initiating a new thread Using Win32 API function **'CreateRemoteThread'** which in turn calls **'LoadLibrary'** API that is responsible for DLL loading. Fig 5 depicts how DLL Injection works.

1.6.5 File-less Malware Approach:

The RDATA part of a disguised application's Portable Executable is where

fileless botnet payloads are frequently hidden (PE). During execution, the stub pulls the malicious code from RDATA and runs it as a separate process in memory, rather than dumping the malicious executable on secondary storage.

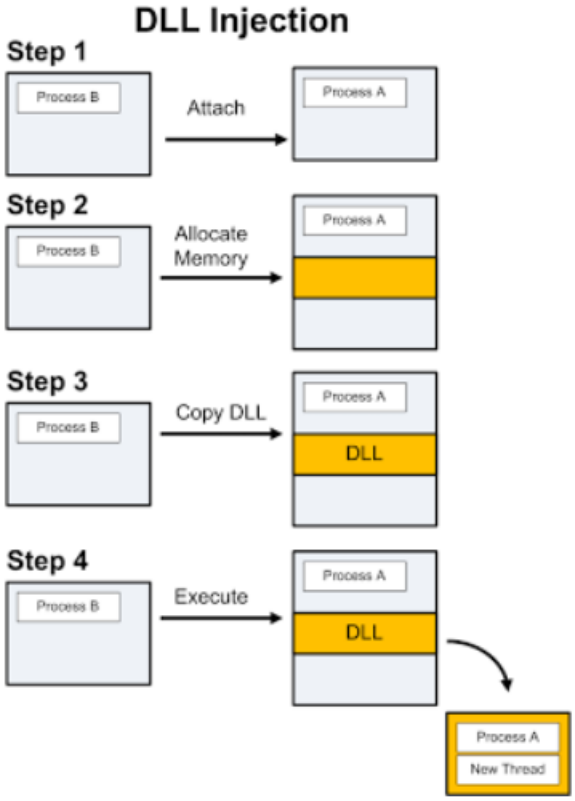


Fig 5. DLL Injection

1.7 Objective of Project

The following objectives must be met in this project:

1. To find, evaluate, and present a comprehensive overview of experimental works on IoT botnet detection research.

2. Examine IoT botnet identification techniques, stages, and stealth tactics.

3. Obtaining the UNSW dataset in order to train & test the deep learning model.

4. Putting the deep learning model through its paces and evaluating its results.
5. Using different deep learning models to compare their performance.
6. Establish a benchmark for future improvements to deep learning models.

Chapter 2

Related Work

2.1 Literature Review

This section addresses any past research on the subject that is closely related to the topic. IoT botnet detection systems using Domain Name Space DNS detection were carefully reviewed by Singh et al [7]. Their research provides a complete review of each approach as well as a unique classification framework for strategies to detect botnets based on DNS. Koroniotis et al. [8] reviewed and examined the existing methods based on forensic techniques and deep learning for IoT botnets. The researchers also looked at how deep learning algorithms may be used in network forensics. Future research directions were also highlighted. The application of machine learning for the detection of IoT botnet anomalies was examined by [9]. The researchers looked at the viability of utilizing autoencoder algorithms for botnet identification and proposed some future research topics for machine learning methods in this field. [11] conducted a demographic analysis of IoT botnets grouped the techniques into detection and avoidance and offered suggestions for further studies into resistance methods. S Dange et al. [10] focussed on machine learning based approaches reviewing the numerous forms of probable IoT assaults and assessing the importance of each category for botmasters. Similarly, J Sengupta et al reviewed threats and security problems in blockchain and industrial IoT in [12]. They concentrated on blockchain based solutions since they saw it as a potential technique for IoT botnet identification. Most researchers only gave a cursory look into IoT botnet detection without going into detail about each approach. Ali et al. [11] conducted a demography study of botnet assaults on IoT devices. Furthermore, several assessments concentrated just on one form of IoT botnet malware, with no

mention of the necessary detection procedures. Y Ji et al. in [14] did a review assessing and investigating Mirai Botnet's malware. They looked at the Mirai Botnet's architectural design and components in-depth as well as the attack tactics and botnet propagation model's effect factor. M. Salim et al. [13] dealt with the identification of a specific type of attack caused by an IoT botnet, and they looked at the distributed denial of service (DoS) attacks and responses in the context of IoT. They highlighted the reasons why botmasters preferred to deploy DDoS assaults against IoT devices, as well as the primary defense mechanisms against DDoS attacks. Moreover [7] addressed detection approaches in IoT contexts based on particular protocols DNS with a full analysis of each strategy. This research introduced a new classification system for botnet detection approaches based on DNS. Table I compares the findings of past research on this subject.

REF	**Year**	**Detection Approach**	**Stealth Tactic**
[11]	2020	Demographic Review	NO
[12]	2020	Blockchain Based	NO
[7]	2019	DNS Based Detection	NO
[8]	2019	Deep Learning And Network Forensics	NO
[9]	2019	AutoEncoder Based	NO
[10]	2019	Machine Learning Based Detection	NO
[13]	2019	DDOS Classification	NO
[56]	2019	RNN, SVM, LSTM-RNN	NO
[50]	2019	C5 Decision Tree, One Class SVM	NO
[14]	2018	Mirai Review	NO
Our method	------	Review All detection approaches	YES

Table I

Previous Studies on IoT Botnet

Honeynet-based detection techniques and Intrusion Detection methods are the two primary types of botnet detection approaches (IDSs). IDSs are typically primarily signature-based or behavioural, however hybrid IDSs combining aspects from both kinds of IDs have indeed been proposed by researchers [50], [51], [52], [53]. Whilst signature-based IDSs can detect bots that fit established patterns but behavioural IDSs can detect bots that aren't visible by analysing network traffic and looking for abnormalities such as excessive network latency, traffic on uncommon ports, and high network volume, to name a few [54].

Outlier or Anomaly-based detection systems can be either: network-based or host-based. Detection in host-based IDSs focuses on evaluating the activity of individual computers, whereas detection in network-based IDSs focuses on monitoring the aggregated network packets [55].

Another researcher [56] used three machine-learning techniques to analyse the Bot-IoT sub-dataset, including RNN (Recurrent Neural Network), SVM (Support Vector Machines) and LSTM-RNN (Long-Short Term Memory Recurrent Neural Network) [56]. When all of the characteristics were included in the experiment, the SVM classifier took the longest to train, but had the greatest accuracy and recall rates.

[50] Proposed a hybrid IDS (HIDS) with the goal of boosting the accuracy of detecting IoT intrusions. This solution incorporates a signature-based IDS for detecting well-known assaults with a behavioural IDS for detecting zero-day threats. The Boosting method was used to combine the classification performance from both mechanisms, with the signature-based part consisting of a C5 decision tree and the behavioural part consisting of a One-Class SVM, with only 13 out of the original 46 features from the Bot-IoT datasets while performing binary classification. The detection accuracy was 93.30 percent when just the signature-based component was

used, and 92.50 percent when only the behavioural part was used. The consolidated accuracy from both, was observed as 99.97 percent.

2.2 Publicly Available Datasets

2.2.1 MedBIoT

Three genuine IoT devices and 80 simulated IoT devices were employed in the study. Two smart switches and one smart light bulb were the actual gadgets. When the devices were in regular operation and after being infected with malware, traffic was collected. The viruses Mirai, BashLite, and Torii were employed.

There are two versions of the dataset: one with no processing (pcap files) and one with 115 characteristics extracted.

2.2.2 IoT-23

Three genuine IoT devices were used, as well as a Raspberry Pi. One smart doorbell, one smart bulb and one smart speaker or virtual assistant were the actual gadgets. Whenever the Connected devices were running normally and when the Raspberry Pi was infected with malware, traffic was recorded. Throughout 20 distinct samples, 11 malicious variants were employed, notably Mirai, Torii, and Gagfyt (aka BashLite).

2.2.3 UNSW-NB15:

2015 saw the dissemination of the NB15 dataset by UNSW. This dataset contains a total of 2540044 records of realistic and abnormal often popularly referred to as attack network events. IXIA's traffic generator made use of three different virtual servers in order to compile these data one server was conFigd to generate unusual network flow while the first two were set up to execute typical network flow. The

dataset contains a wide variety of attacks including Reconnaissance, DoS, Worms, Fuzzers, Generic and Shellcode amongst others. The dataset was searched using a variety of techniques such as Bro IDS and 49 characteristics were extracted.

2.2.4 N-BaIoT:

N-BaIoT is a freely accessible dataset that is commonly used to identify botnets. There were nine IoT devices utilised in this study. Two smart doorbells, one smart thermostat, one smart baby monitor, four security cameras, and one webcam were among the gadgets. When the gadgets were in regular operation and after being infected with malware, traffic was collected. Malware such as Mirai and BashLite were employed.

2.3 Detection Methods

Researchers have presented many Botnet Detection approaches in recent studies Fig 6 depicts all of the potential solutions to this problem.

2.3.1 Supervised Learning Based:

[15] proposed a method for identifying IoT botnets during their spread. The study contains a well-known logistic regression model. It may be used to determine the chance of a bot to be launched by a system starting a connection. There is additional information on network protocols used to gain unauthorized access to systems and obtain orders from a c&c server. The technique shown here is suitable for detecting botnets spread by brute force assaults on SSH and Telnet services. To identify zero day attacks the authors of [27] employed supervised machine learning approach to discover trends and discriminate abnormalities in an IoT context. They used random forest classifier with only four kind of threats included in the training phase and ten

types of assault in the test dataset. When it came to identifying new threats the suggested model performed admirably with a TPR of 99 percent a TNR of 100 percent and close to zero false warnings.

2.3.2 Un-supervised Learning-Based:

Researchers in [30] suggested a strategy based on Grey Wolf Optimization for optimizing one class SVM hyper-parameters while controlling the selection feature simultaneously. The efficacy of the suggested method is demonstrated by experimental results utilizing GWO on NN-BaIoT dataset. The researchers of [43] proposed a methodology based on association rule learning to determine the patterns and regularities of assaults using data gathered on a broad scale from darknets with a significant stream. By finding similarities in IoT traffic, such as destination ports, TCP window size, and operation type they were able to determine the behaviours of striking hosts associated with known malware groups.

2.3.3 Deep Learning-Based:

Several deep neural network-based algorithms have been deployed as IoT classification frameworks to discriminate normal IoT connections and assaults using a smart intrusion detection system. The outcomes of the performance evaluations show that these tactics are effective. McDermott et al [17] proposed to use a bidirectional LSTM RNN for identification of botnet in conjunction with word embedding. Results were compared to unidirectional LSTM RNN and both models were found to be highly accurate. [20] proposed a unique technique for detecting Linux IoT botnets that included a PSI graph with a CNN classifier. 10033 ELF files have been employed in the experiment According to the results of the test the PSI graph-based CNN classifier had a precision of 92 percent.

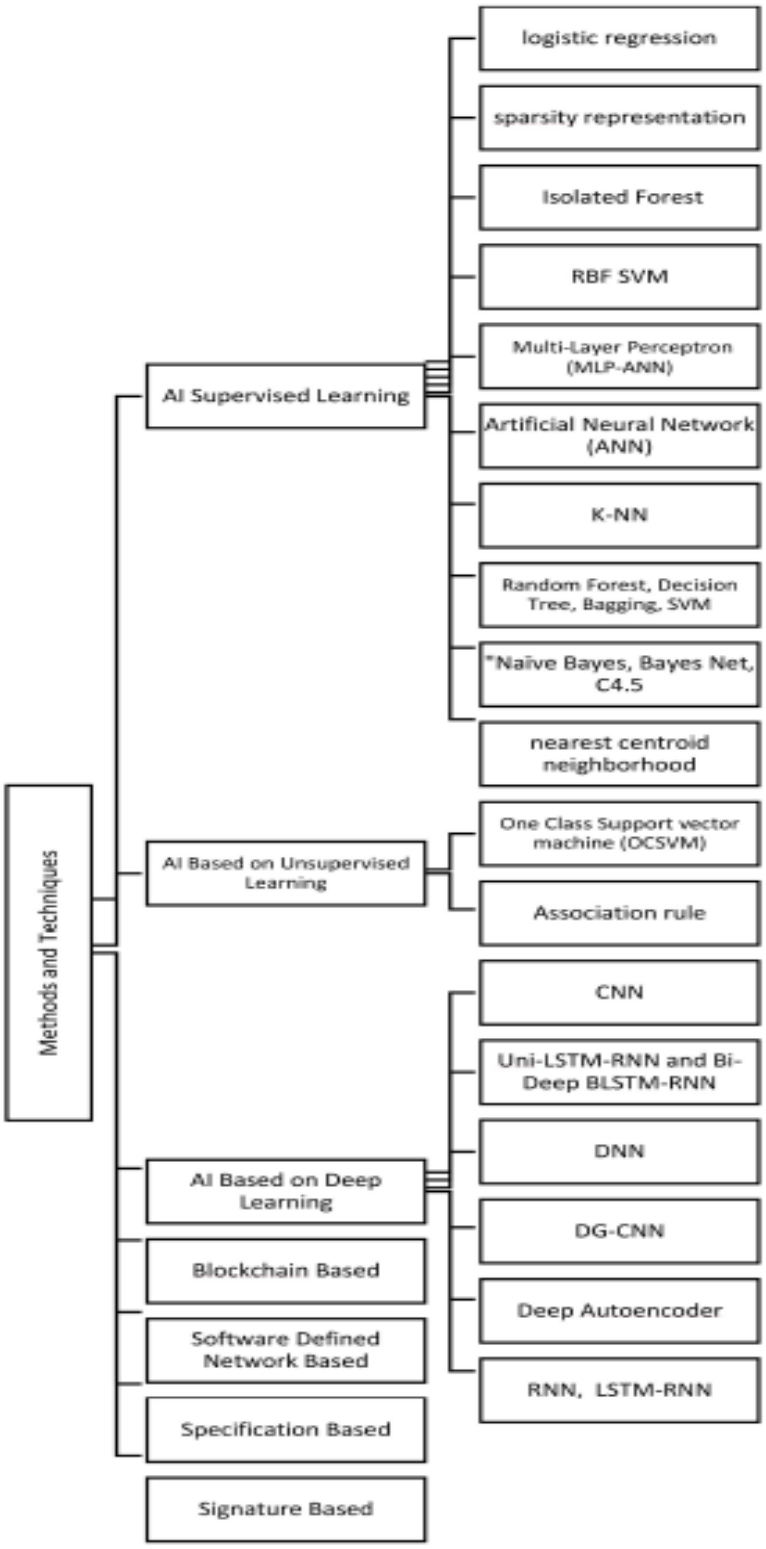


Fig 6. Detection Methods

Source: Adapted from [47]

2.3.4 Blockchain-Based:

AutoBotCatcher [41] was created with the objective of dynamically analyzing IoT device communities based on network traffic flow among devices. The researchers designed a P2P botnet detection and prevention framework relying on blockchain that conducts group classification on IoT network flows. In this framework gateway-edge devices became peers of the BFT blockchain where system suppliers and security regulators act as block producers and participate in the consensus process. Blockchain was utilized to create snapshots of the reciprocal communication graphs of IoT devices.

2.3.5 SDN-Based (Software-Defined-Networking):

SDN is a contemporary networking technology that allows the complete behavior of the network to be handled by a single application known as the SDN controller. SDN controller enables the rapid reaction to security threats, traffic filtering and execution of complicated security rules [53]. [54] investigated the Mirai botnet and advocated using SDN to apply multiple flow rules and proactively renew them as necessary. There are practical issues with integrating this approach into IoT networks that are typically heterogeneous and poorly managed.

2.4 Analysis

The purpose of this section is to compare various botnet identification techniques offered by researchers. The comparison of various strategies is shown in Table II. We gathered the data in Table II from different researches mentioned in the table to compare various detection methods and studies discussed in this report. As can be seen, experiments that combined deep learning techniques with traditional machine learning approaches did well. IoT devices have limited processing power and resources, and dimensionality reduction approaches aid in the reduction of

characteristics, making botnet detection fast and accurate. [24] obtained 98 percent accuracy using dimensionality reduction along with Decision tree. Deep autoencoders outperformed all other detection methods listed in this research in terms of false alarm rate, achieving nearly zero false alarms.

Ref	**Dataset**	**Detection Approach**	**Evaluation Measurement**
[15]	Collected from 100 Botnets	Logistic Regression	97.3%
[17]	Collected from 3 servers and 2 cameras	LSTM-RNN and BLSTM-RNN	99%
[18]	Collected from honeypots	Machine Learning	Not Mentioned
[19]	N-BaIoT	sparsity representation	Better than single layer autoencoder
[20]	Collected from benign and IOTPOT	DG-CNN	92%
[2]	N-BaIoT	Deep Autoencoders	FPR 0
[21]	Balanced N-BaIoT	SVM and isolated forest	90%
[22]	Stored scanned pattern	KNN, Random Forest, Gaussian Naive Bayes	94%, 77.5%, 88.5%
[23]	N-BaIoT	CNN	99.5%
[24]	N-BaIoT	KNN, Decision Tree, Dimensionality reduction	94%, 98%
[25]	Collected DNS query log data	Adaboost, Bagging, KNN, Naive Bayes	KNN, Precision, F-Measure, ROC Area = 1
[26]	VirusShare and IoTPOT	Random Forest, Bagging, Radial Basis Function SVM, Decision Tree	Accuracy: 97%, F-Measure = 98%

[27]	Subset of N-BaIoT	Naive Bayes, KNN, Random Forest	99%, Almost 0 False Alarms
[28]	Not Mentioned	Deep Learning and SDN	Not Mentioned
[29]	11200 ELF Files	PSI Graph, graph2vec using CNN	98.7%
[30]	N-BaIoT	One Class SVM, Grey Wolf Optimization	96 to 99%
[31]	Subset of N-BaIoT	Multi layer perception ANN	100%
[32]	USNW-NB15, KDD99	Decision Tree, Association Rule Mining ANN	93%
[33]	Collected 100 pcap from repositories	CNN	96%
[34]	Subset of N-BaIoT	KNN	100%
[35]	Collected power consumption data	CNN	96.5%
[36]	BoT-IoT	SVM, LSTM-RNN	99%
[37]	BoT-IoT	Naive Bayes, Random Forest	99.9%
[38]	Collected data from 34974 IoT devices and 7193 non-IoT devices	CNN	Number of compromised devices = 400
[39]	IoTPOT	Random Forest	95%
[40]	Not Mentioned	Nearest centroid neighbourhood	99%
[41]	Not Mentioned	Blockchain	Not Mentioned
[42]	Not Mentioned	Neural Network, Blockchain	Not Mentioned
[43]	Collected From Darknet	Association Rule	Not Mentioned
[44]	Captured 1,840,973,403 Packets	Association Rule	Not Mentioned

[45]	Not Mentioned	SDN	Not Mentioned
[46]	DNS traffic generated by 19 different botnets	Threshold Random Walk	94%

Table II

Comparison of Detection Methods.

Chapter 3

Technology Used

3.1 Deep Learning

Artificial neural networks and feature learning are the core of deep learning, a subfield of machine learning. Instead of employing a set of pre-programmed commands, DL algorithms may learn from enormous volumes of data. A Variety of frameworks relying on deep learning have been implemented, with competitive results in areas like as speech recognition and computer vision, and medical image processing, among others. The potential uses of deep learning in network traffic inspection and intrusion detection are becoming more apparent. DL models for security of network device have come a long way in the last few years. Fig 7 below depicts various DL Methods.

3.2 Convolutional Neural Network

Another deep learning based technique, named CNN is an intelligent technique that is used to create a fast picture categorization system. Likewise, the CNN model can aid in the construction of secure systems. The CNN method is comparable to a traditional neural network in that it has four layers: an input layer, a convolutional layer, a pooling layer, and a fully connected layer [58].

3.2.1: Convolutional Layer:

The convolutional layer which includes several convolution kernels explores and filters the training sample. The weight matrix of the feeded data set is produced by the convolutional layer which also regenerates “weighted summation kernel layer”. Integer values are used to reduce the input size in the filter. Filter size, zero padding

and stride are three important hyperparameters that play a role in improving the performance of convolutional kernels. Choosing optimal values can assist minimising the complexities of the network thereby improving the accuracy. Fig 8 depicts the CNN layers in detail.

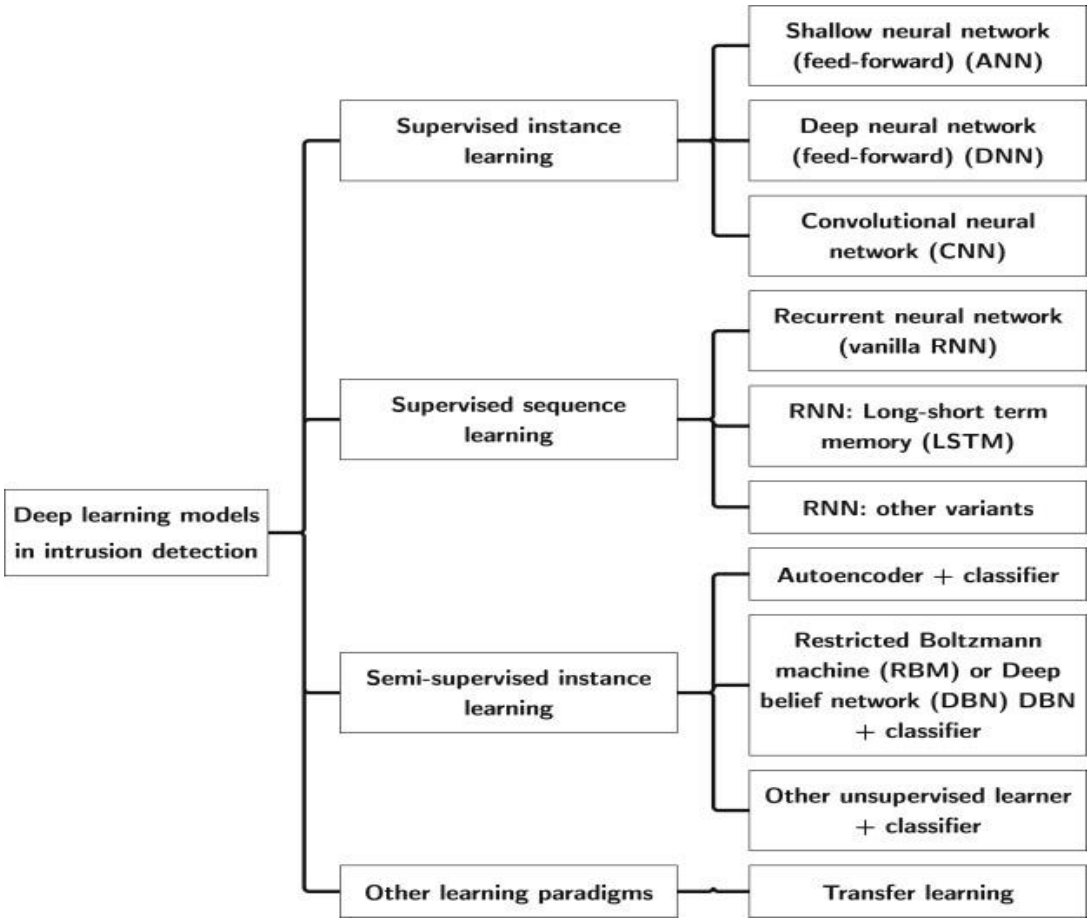


Fig 7. Various DL Methods [57]

Source: Adapted from [57]

Equation 1 depicts the operation of convolutional layers where w represents the weights, x represents the input, b represents the basis and f is an activation function.

$$x_i = f\left(w_i \otimes x_{i-1} + b_i\right)$$

Equation 1: operation of convolutional layer.

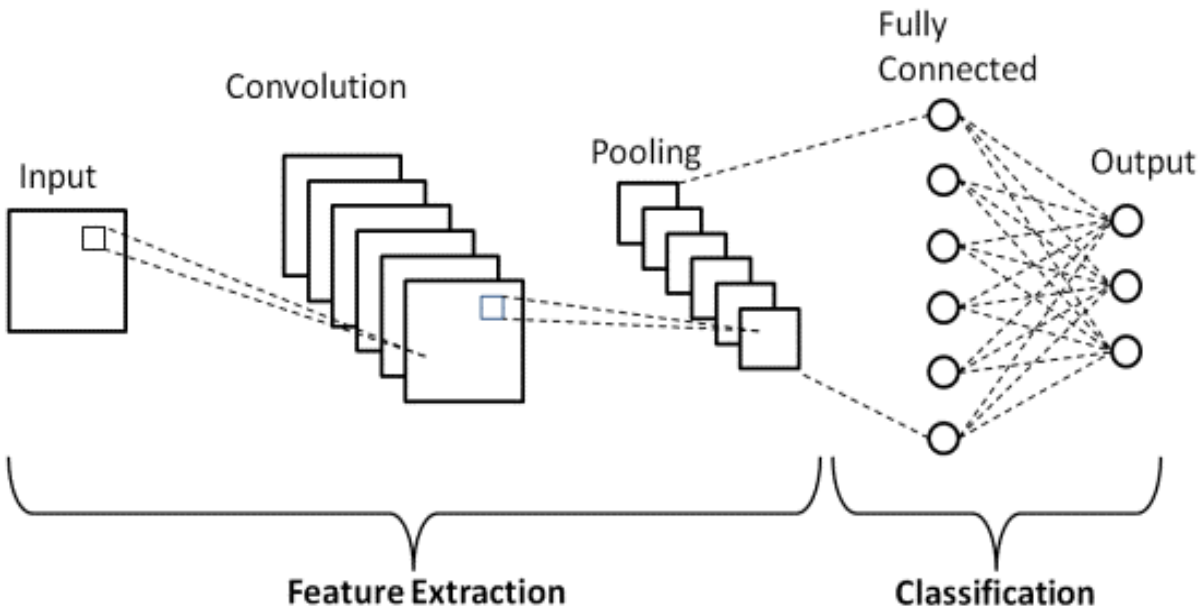


Fig 8. CNN Layers

3.2.2: Polling Layer:

By picking the peak value region by region, the pooling layer is utilised for building a fit matrix and minimising the number of characteristics in the feature map. The following layer processes this matrix. pooling size is configurable and we set it at 5 for our study. The max function or the average function can be used to implement polling. To limit the amount of characteristics, max polling selects the peak value, whereas average polling computes the average region by region. Fig 9 depicts the polling layer architecture.

3.2.3: Fully-Connected-Layer:

The fully connected layer represents the final layer of the convolutional neural network. All unit nodes of layer M in the completely connected layer is directly linked to all unit node in layers (M -1) and (M + 1). Unlike typical ANN , there is no link between nodes in the same layer. As a result, this layer need extensive training and testing. Fig 10 depicts the FC layer of CNN.

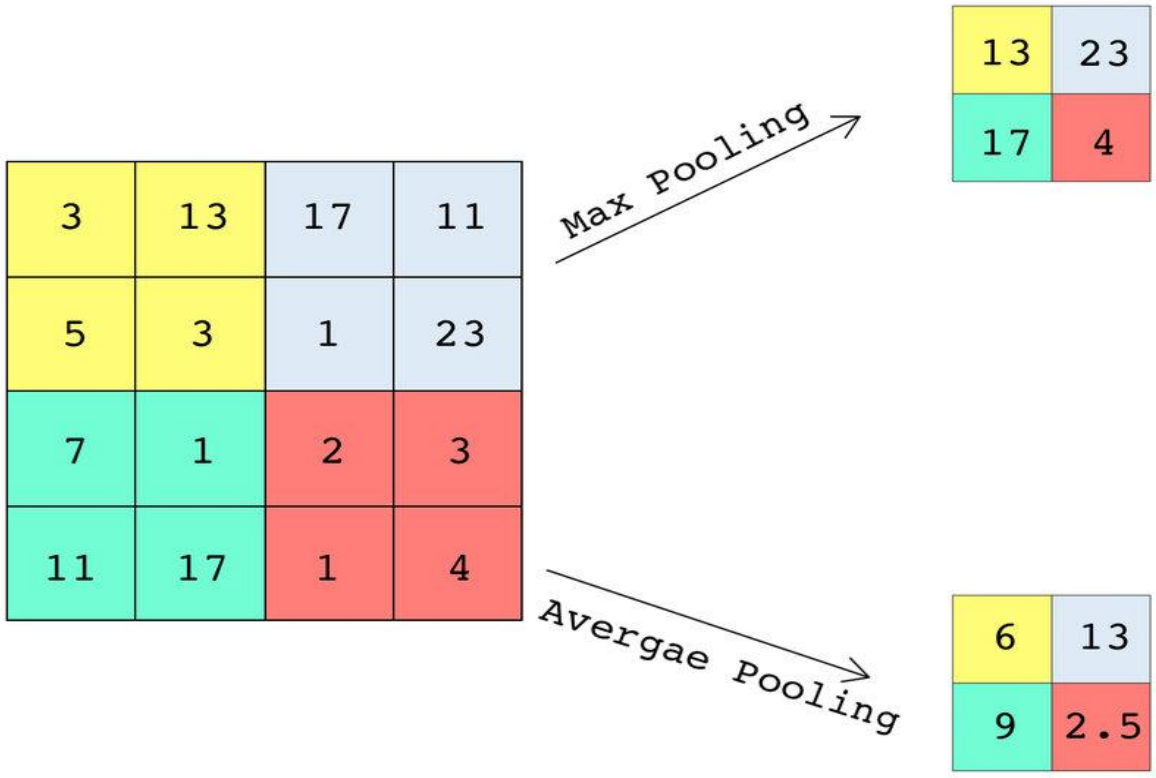


Fig 9. Polling Layer

3.3 LSTM (Long-Short-Term-Memory)

Among deep learning models utilised in tremendous realistic applications is the RNN method. Fig 11 depicts the construction of a RNN model wherein x represents input and y represents categorized output. One form of RNN is the LSTM model. Unlike traditional feedforward neural networks, the LSTM is utilised to analyse sequence data using feedback connections. In the same way that when you watch a film you could remember what happened in the previous episode, recurrent neural networks (RNNs) also retain previous knowledge and incorporate it into how they process new data. RNNs are unable to remember long-term context as a result of the diminishing gradient. When employing LSTMs, one must take great care to avoid worrying about the context over the long run.

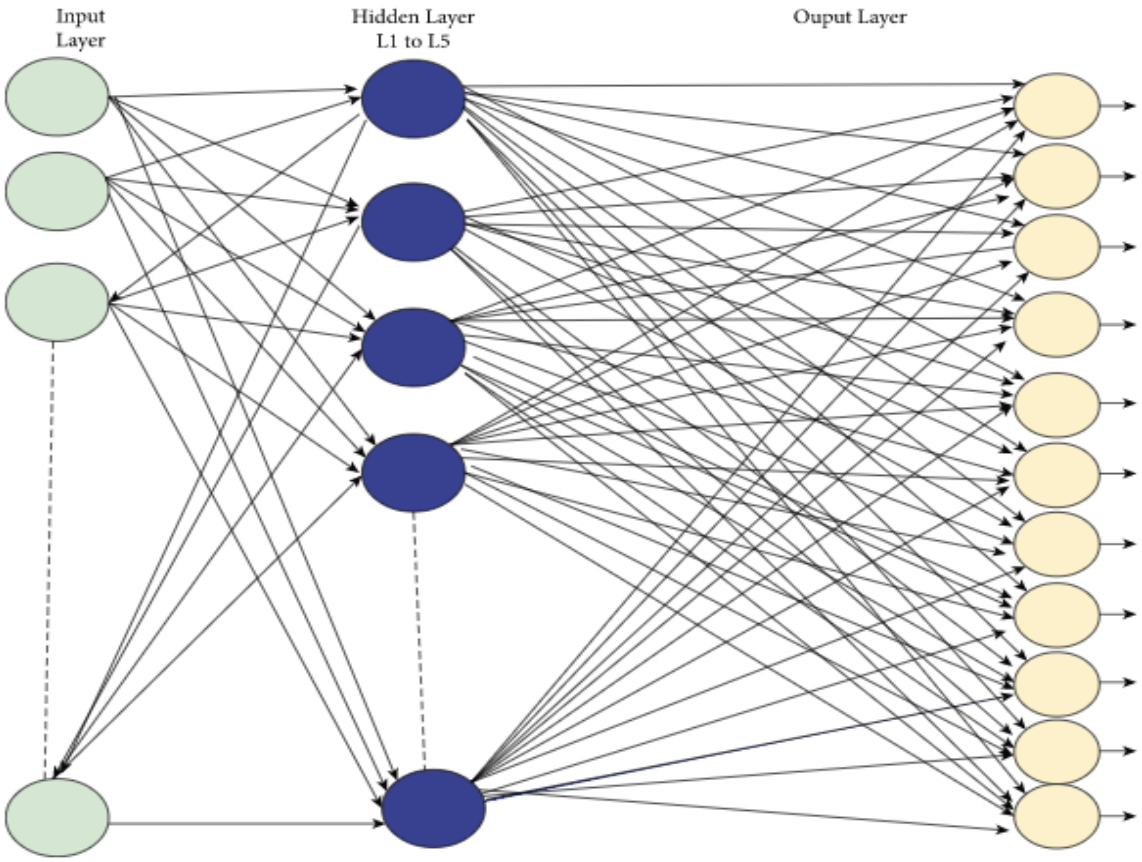


Fig 10. Fully Connected Layer

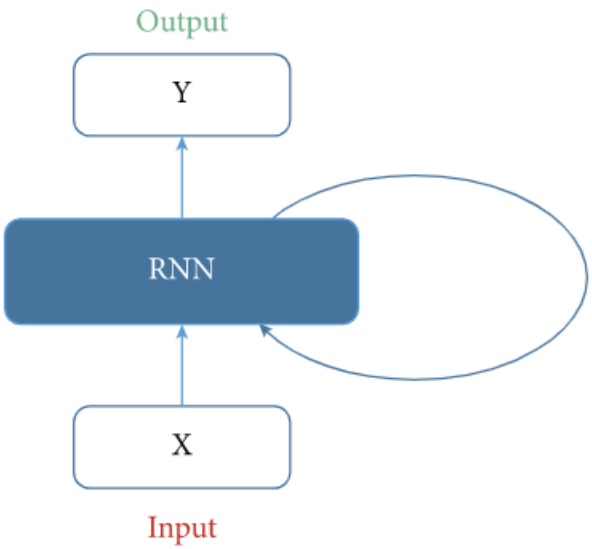


Fig 11. RNN Architecture

Three gates namely output forget and input gate are the primary gates of the LSTM The training data is stored in long term memory via the input gate The short term memory is initialised from the previous iteration whereas the long term memory is initialised from the present inputs The input gate contains filters that helps extracting training data and eliminate useless data while the valuable data travels through to the sigma function. Output 1 in sigma represents extremely significant values whereas the 0 value represents inconsequential ones. The input layer's output is kept in long term memory. Multiplying the values in forget vector by the current input gate another gate called forget gate determines to save or reject the information. The forget gate's output will be transferred to the succeeding cell which will get a fresh data from long term memory. The construction of LSTM is depicted in Fig 12.

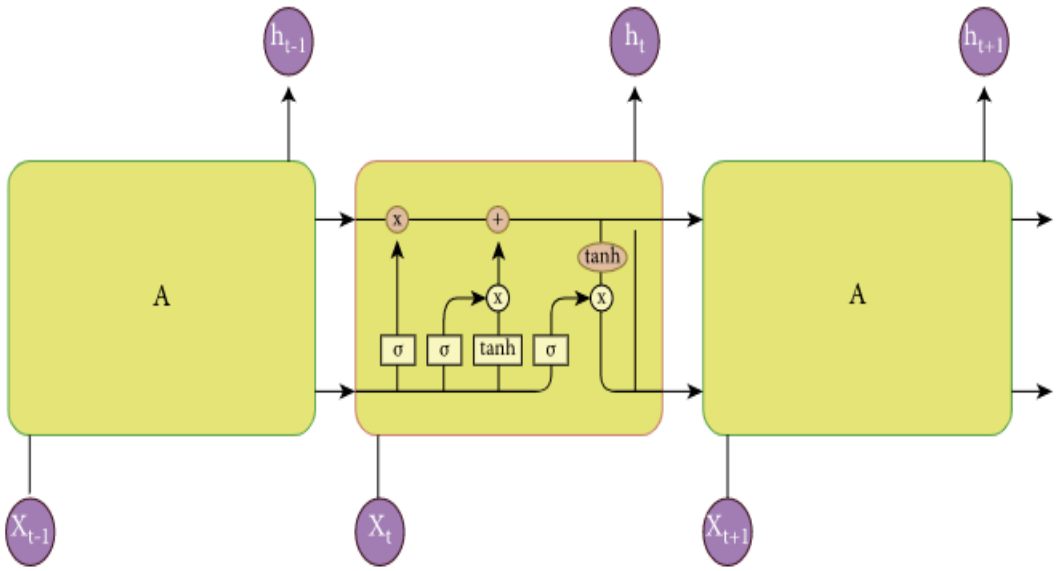


Fig 12. LSTM Architecture

Chapter 4

Methodology

4.1 Dataset Used

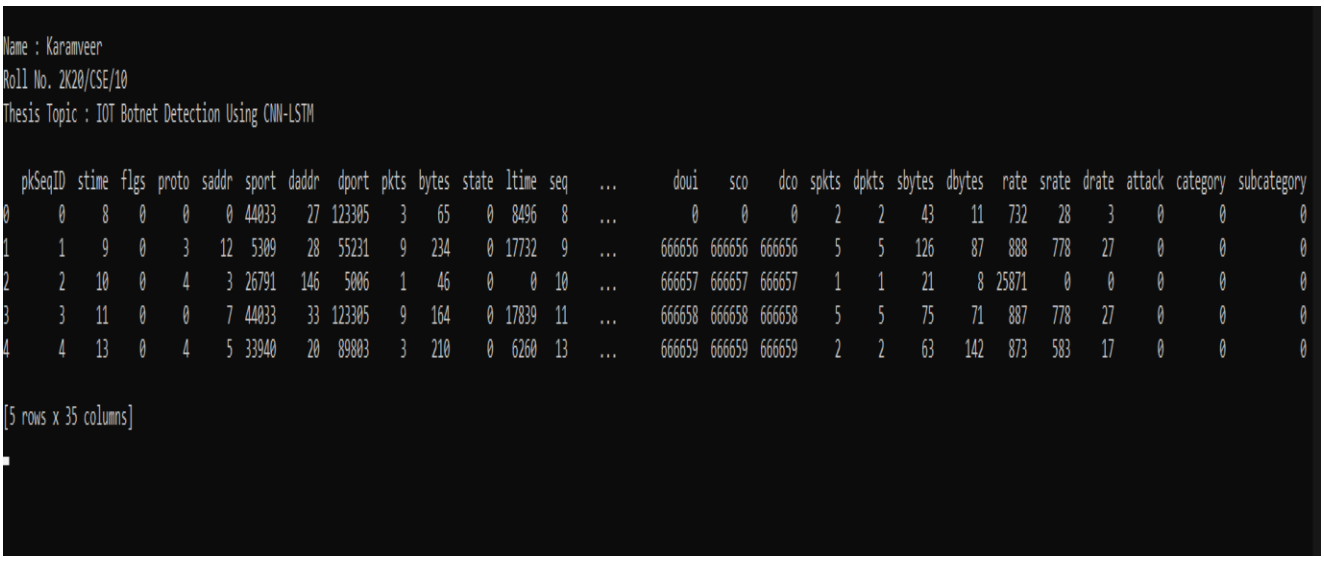
2015 saw the dissemination of the NB15 dataset by UNSW. This dataset contains a total of 2540044 records of realistic and abnormal often popularly referred to as attack network events. IXIA's traffic generator made use of three different virtual servers in order to compile these data one server was conFigd to generate unusual network flow while the first two were set up to execute typical network flow. The dataset contains a wide variety of attacks including Reconnaissance, DoS, Worms, Fuzzers, Generic and Shellcode amongst others. The dataset was searched using a variety of techniques such as Bro IDS and 49 characteristics were extracted.

Ser No.	Feature	Description
1	Stime	Start time of connection
2	Proto	Protocol being used in the flow
3	proto number	'Proto' feature representation in numerical form
4	saddr	IP of source
5	sport	Port number of source
6	daddr	IP address of destination
7	dport	Port number of destination
8	pkts	No. of packets involved in current flow
9	bytes	No. of bytes involved in current flow
10	state	State of current transaction
11	state number	'state' feature representation in numerical form
12	ltime	Finish time of connection
13	dur	Total record duration
14	spkts	No. of packets involved in current flow from source-to-destination
15	dpkts	No. of packets involved in current flow from destination-to-source
16	sbytes	No. of bytes involved in current flow from source-to-destination
17	dbytes	No. of bytes involved in current flow from destination-to-source
18	TnBPSrcIP	Total 'bytes' of a 'saddr' in 100 connections
19	TnBPDstIP	Total 'bytes' of a 'daddr' in 100 connections
20	TnP_PSrcIP	Total 'pkts' of a 'saddr' in 100 connections
21	TnP_PDstIP	Total 'pkts' of a 'daddr' in 100 connections
22	TnP_PerProto	Total 'pkts' of a 'proto' in 100 connections
23	TnP_Per_Dport	Total 'pkts' of a 'dport' in 100 connections
24	AR_P_Proto_P_SrcIP	Avg 'pkts' / 'dur' per 'saddr' in 100 connections
25	AR_P_Proto_P_DstIP	Avg 'pkts' / 'dur' per 'daddr' in 100 connections
26	N_IN_Conn_P_SrcIP	Connections with same 'saddr' in 100 connections
27	N_IN_Conn_P_DstIP	Connections with same 'daddr' in 100 connections
28	AR_P_Proto_P_Sport	Avg 'pkts' / 'dur' per 'sport' in 100 connections
29	AR_P_Proto_P_Dport	Avg 'pkts' / 'dur' per 'sport' in 100 connections

Fig 13. Selected Features

35 most significant labelled characteristics were retrieved from network traffic in which last three are dependent variables. The characteristics retrieved from the dataset that have been used in this work are depicted in Fig 13.

Fig 14 and Fig 15 depicts the snapshot of the data and the dataset's distribution by class respectively. The dimension of the dataset used is (2,540,044 x 35).



```
Name : Karamveer
Roll No. 2K20/CSE/10
Thesis Topic : IOT Botnet Detection Using CNN-LSTM

   pkSeqID  stime  flgs  proto  saddr  sport  daddr  dport  pkts  bytes  state  ltime  seq  ...    doui    sco    dco  spkts  dpkts  sbytes  dbytes  rate  srate  drate  attack  category  subcategory
0        0      8     0      0      0  44033     27 123305     3     65      0   8496    8  ...       0      0      0      2      2      43      11   732     28      3       0         0            0
1        1      9     0      3     12   5309     28  55231     9    234      0  17732    9  ...  666656 666656 666656      5      5     126      87   888    778     27       0         0            0
2        2     10     0      4      3  26791    146   5006     1     46      0      0   10  ...  666657 666657 666657      1      1      21       8 25871      0      0       0         0            0
3        3     11     0      0      7  44033     33 123305     9    164      0  17839   11  ...  666658 666658 666658      5      5      75      71   887    778     27       0         0            0
4        4     13     0      4      5  33940     20  89803     3    210      0   6260   13  ...  666659 666659 666659      2      2      63     142   873    583     17       0         0            0

[5 rows x 35 columns]
```

Fig 14. Dataset Snapshot

Class	No. of records	% of total data
Non-anomalous (normal)	2,218,761	87.35
Exploits	44,525	1.75
Reconnaissance	13,987	0.55
DoS	16,353	0.64
Generic	215,481	8.48
Shellcode	1,511	0.06
Fuzzers	24,246	0.95
Analysis	2,677	0.11
Backdoor	2,329	0.1
Worms	174	0.01
Total	**2,540,044**	

Fig 15. Distribution of Data

4.2 Proposed system

To identify botnet assaults from different sorts of Connected devices we used CNN and LSTM classifiers in this work. A Hybrid CNN-LSTM based method employed in our research has a common pattern as shown in Fig 16. The input data is sent into a CNN network which selects characteristics and then the CNN output is fed into an LSTM model which classifies the data as ordinary or hostile.

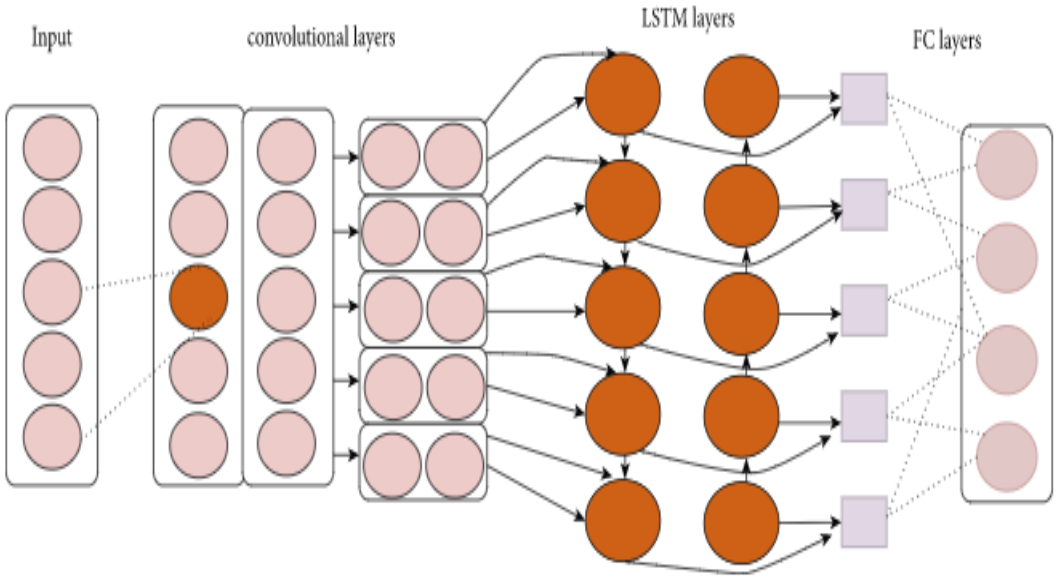


Fig 16. CNN-LSTM Method

4.2.1 Data Collection

In this project, the outlier detector is trained using the dataset UNSW-NB15, as detailed in section 4.1. To train our outlier detection, we employed 35 characteristics as in Fig 13.

4.2.2 Feature Extraction

We obtain a behavioural traffic snapshot of the gadgets and protocols that transmitted this payload whenever a packet is received. The snapshot extracts 35

traffic metrics across several sequential windows to summarise all packets that are emerged from same IP, emerged from same source MAC and IP address, previously been submitted between sender & receiver IPs, or previously been submitted between sender & receiver TCP/UDP sockets (socket). These characteristics may be calculated quickly and gradually, allowing for real-time identification of malicious nodes. Furthermore, while these traits are general, they can capture unique characteristics of Mirai's assaults, such as source IP spoofing [2]. Whenever an exploited IoT gadget parodies an IP address, for example, the characteristics aggregated by the Sender MACIP, Sender IP, and Channel will instantly show a big abnormality owing to the undetected behaviour emanating from the faked IP address.

4.2.3 Training and Testing

We divided the dataset into two parts: 80 percent for training and 20 percent for testing. We utilised the conv layer with kernel size 2 and filters 8 in the first layer. In this study, the Relu activation function is employed. None,1,32,1 is the input shape to the conv layer. In the polling layer, max-polling is utilised with pool size 2 and the output is flattened before being sent to the FC layer. Once the characteristics are chosen by the CNN layer, the output is sent to the LSTM. As an activation function, the sigmoid function is applied here. Adams optimizer was utilised for optimization. To begin, a number of observations are examined in order to divide the characteristics into distinct groups. We used 10 epoch during training.

The network learns routine behaviour patterns from datasets of different kinds. The experimental results for the proposed study are discussed in the next chapter.

Chapter 5

Experimental Results

5.1 Evaluation Metrics

Metrics like accuracy, precision, F1 score, and recall were used to Fig out how well the method for finding botnet intrusions worked. Fig 17 depicts the following equations for true positive, false positive, true negative, and false negative.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{FP} + \text{FN} + \text{TP} + \text{TN}} \times 100\%,$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%,$$

$$F1 - \text{score} = 2 * \frac{\text{precision} \times \text{sensitivity}}{\text{precision} + \text{sensitivity}} \times 100\%,$$

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%,$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%,$$

Fig 17. Equations

5.2 Results

Preliminary tests indicate that determining whether such a gadgets' packet flow is aberrant or not based on individual occurrence allows for extremely precise identification of IoT-based botnet assaults (high TPR). However, ordinary cases were mistakenly labeled as aberrant too but in small numbers. (about 0.3- 0.4 percent of the time).

Keras was utilised for training and refinement. Every node in convolution layer has an input vector with the same dimension as the dataset's number of features (i.e., 35). CNN successfully execute dimensionality reduction automatically, so that the polling layer in between effectively compresses and replicates the input layer's important properties. In our tests, we divided the dataset into two parts: 80 percent for training and 20 percent for testing. We utilised the conv layer with kernel size 2 and filters 8 in the first layer. In this study, the Relu activation function is employed. (None,1,32,1) is the input shape to the conv layer. In the polling layer, max-polling is utilised with pool size 2 and the output is flattened before being sent to the FC layer. Once the characteristics are chosen by the CNN layer, the output is sent to the LSTM. As an activation function, the sigmoid function is applied here. Adams optimizer was utilised for optimization. To begin, a number of observations are examined in order to divide the characteristics into distinct groups. We used 10 epoch during training.

Our approach detected each successful attack conducted by each and every infected Iot system, resulting in an accuracy of 99.98%, F1-Score of 99.9% using CNN-LSTM whereas it was 96.05% and 97.98% for CNN. For LSTM accuracy was recorded as 96.99% and F1-Score as 98.4%. Fig 18 shows the output results for same. Hybrid model performed substantially superior than that of the CNN and LSTM models used separately. In addition, our strategy generated the minimal false alerts. It had a shorter and much more stable average FPR of 0.5 lower than CNN and LSTM.

5.3 Outputs

5.3.1 Performance Evaluation

Fig 18 illustrates the performance of each model.

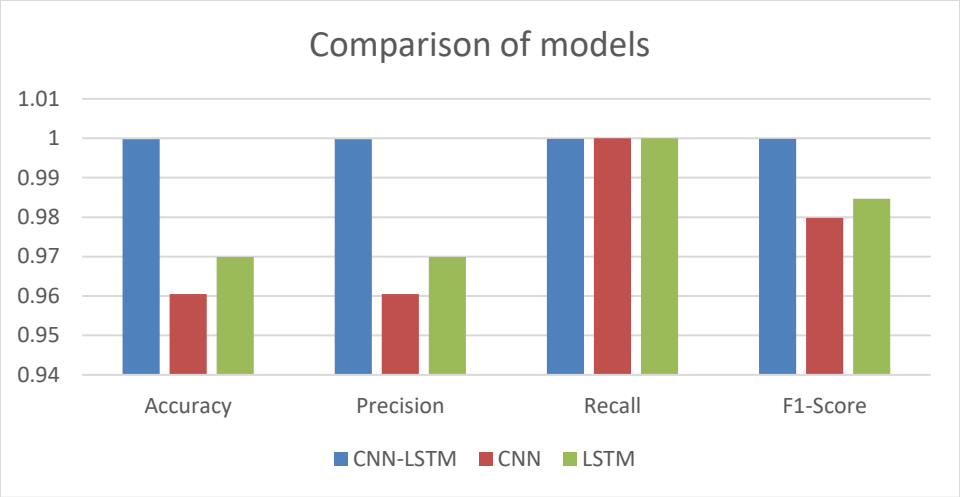


Fig 18. Performance comparison

5.3.2 Confusion Matrix

Fig 19 illustrates the confusion matrix.

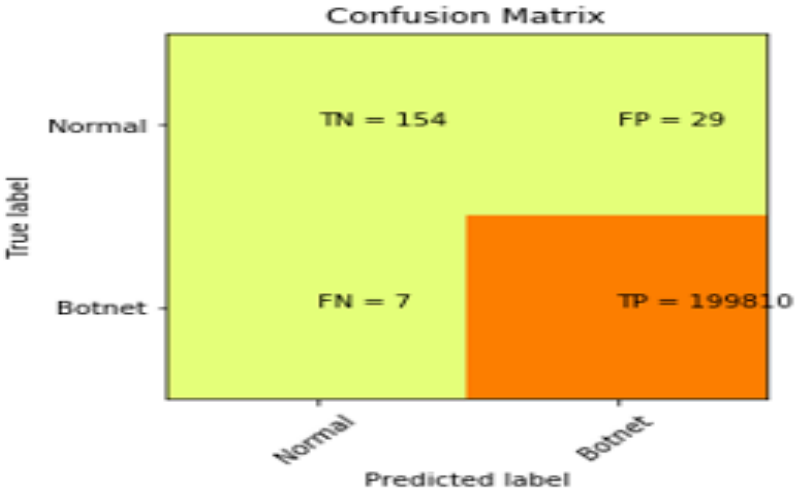


Fig 19. Confusion Matrix

5.3.3 Training Outcome

Fig 20 illustrates the training phase.

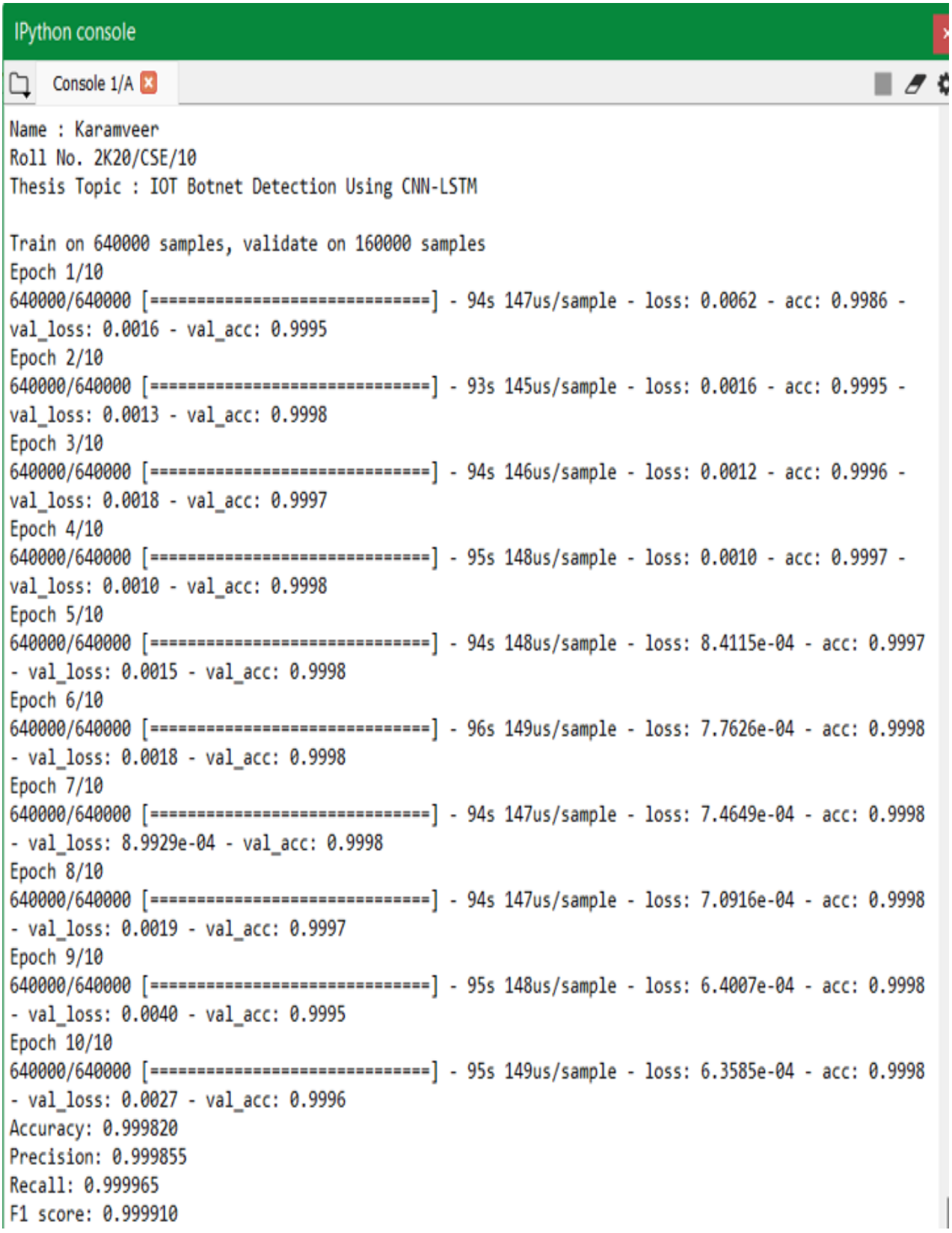


Fig 20. Training Outcome

5.3.4 Accuracy vs Epoch

Fig 21 illustrates the accuracy-epoch graph.

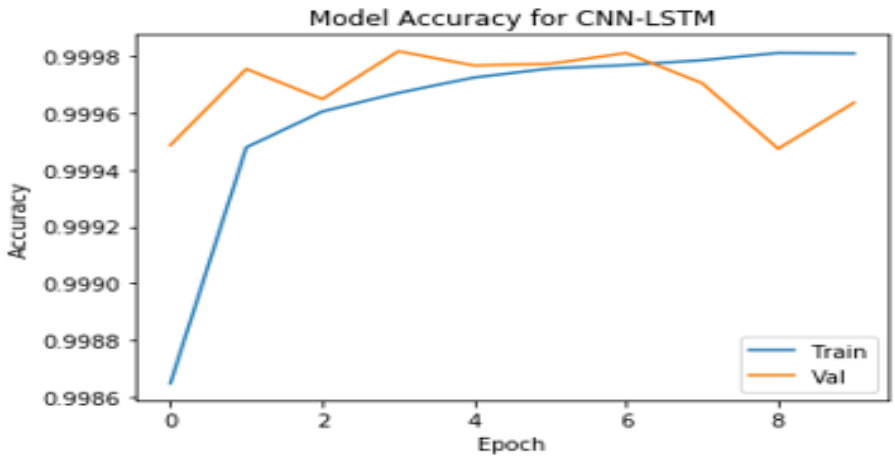


Fig 21. Accuracy-Epoch Graph

5.3.5 Model Loss

Fig 22 illustrates the accuracy-epoch graph.

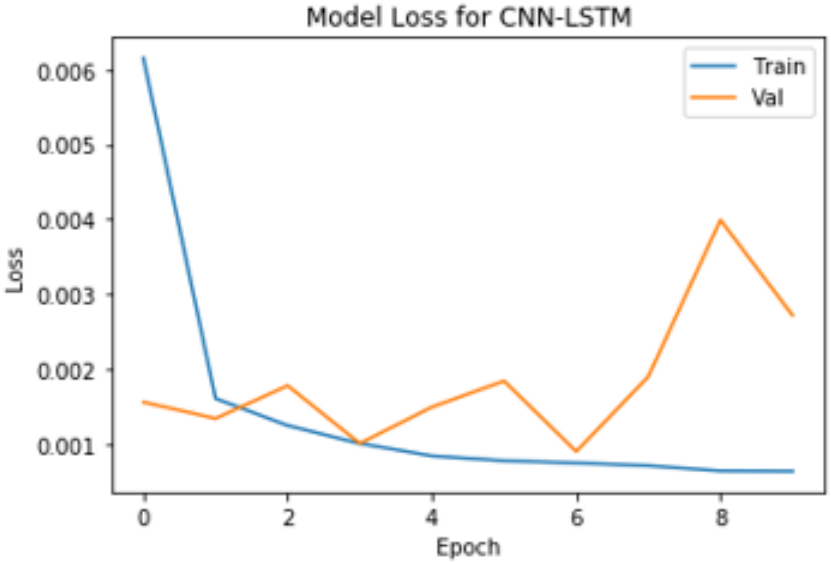


Fig 22. Model Loss

Chapter 6

Conclusion

Despite the fact that IoT gadgets have now become a vital element of the Internet, they are plagued with security flaws and vulnerabilities. In most commonly used Iot nodes, security has never been a design focus. As a result, numerous cyberattacks were successful in penetrating these machines and recruiting them to carry out serious strikes.

The most modern techniques and tactics for identifying IoT botnets were extensively examined in this thesis. The count of publications has continuously grown indicating that this subject is being explored and will gain further attention in the future All of these papers were published in renowned journals and at prestigious conferences. We discovered that most researchers concentrated on discovering the botnet at a late stage when they launched assaults after receiving order from the botmaster. Early identification of botnets is required to minimise the damage caused by such illegal instruments used against the public. We reviewed several botnet stealth strategies that had not been covered in earlier research on the subject. This study will assist researchers in gaining an indepth understanding of ongoing work on this issue and digging deeper to contain these illicit instruments as soon as feasible.

The suggested hybrid deep learning CNN-LSTM approach in this document efficiently detects botnet assaults and may be utilised to increase NN efficiency by manipulating hidden layers. The suggested model's excellent performance depends on the usage of a trustworthy dataset. This research shows that using deep learning to detect botnets yields accuracy of about 99.8%, which is the greatest across SVM, NB, CNN, LSTM and backpropagation techniques.

References

[1] McKinsay Global Institute, ”Unlocking the Potential of the Internet of Things”, June 2015, Source:http://www.mckinsey.com/industries/hightech/our-insights/the-internet-of-things-the-value-of-digitizing-thephysical-world.

[2] Meidan, Yair, Michael Bohadana, Yael Mathov, Yisroel Mirsky,Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. ”N-BaIoT Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders.” IEEE Pervasive Computing 17, no. 3 (2018): 12-22.

[3] Ahmad, U., Song, H., Bilal, A., Alazab, M., \& Jolfaei, A. (2019).Securing smart vehicles from relay attacks using machine learning. The Journal of Supercomputing, 1-18.

[4] Kaur, M., Kaur, G., Sharma, P. K., Jolfaei, A., \& Singh, D. (2019).Binary cuckoo search metaheuristic-based supercomputing framework for human behavior analysis in smart home. The Journal of Supercomputing, 1-24.

[5] Alazab, M. (2015). Profiling and classifying the behavior of malicious codes. Journal of Systems and Software, 100, 91-102.

[6] Huda, S., Abawajy, J., Alazab, M., Abdollalihian, M., Islam, R., \&Yearwood, J. (2016). Hybrids of support vector machine wrapper and filter based framework for malware detection. Future Generation Computer Systems, 55, 376-390.

[7] Singh, M.; Singh, M.; Kaur, S. Issues and challenges in DNS based botnet detection: A survey. Comput. Secur. 2019, 86, 28–52.

[8] Koroniotis, N.; Moustafa, N.; Sitnikova, E. Forensics and deep learning mechanisms for botnets in Internet of Things: A survey of challenges and solutions. IEEE Access 2019, 7, 61764–61785.

[9] Alhajri, R.; Zagrouba, R.; Al-Haidari, F. Survey for anomaly detection of IoT botnets using machine learning auto-encoders. Int. J. Appl. Eng. Res. 2019, 14, 2417.

[10] Dange, S.; Chatterjee, M. IoT Botnet: The Largest Threat to the IoT Network. In Advances in Intelligent Systems and Computing; Springer: Singapore, 2019; pp. 137–157.

[11] Ali, I.; Ahmed AI, A.; Almogren, A.; Raza, M.A.; Shah, S.A.; Khan, A.; Gani, A. Systematic literature review on IoT-based botnet attack. IEEE Access 2020, 8, 212220–212232.

[12] Sengupta, J.; Ruj, S.; Das Bit, S. A Comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT. J. Netw. Comput. Appl. 2020, 149, 102481.

[13] 14alim, M.M.; Rathore, S.; Park, J.H. Distributed denial of service attacks and its defenses in IoT: A survey. J. Supercomput. 2019, 76, 5320–5363.

[14] Ji, Y.; Yao, L.; Liu, S.; Yao, H.; Ye, Q.; Wang, R. The study on the botnet and its prevention policies in the internet of things. In Proceedings of the 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design (CSCWD), Nanjing, China, 9–11 May 2018; pp. 837–842.

[15] Prokofiev, A.O.; Smirnova, Y.S.; Surov, V.A. A method to detect Internet of Things botnets. In Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), St. Petersburg, Russia, 29 January–1 February 2018; pp. 105–108.

[16] Avast Threat report, Available online: https://decoded.avast.io/janvojtesek/putting-an-end-to-retadup-a-malicious-worm-that-infected-hundreds-of-thousands/

[17] McDermott, C.D.; Majdani, F.; Petrovski, A.V. Botnet detection in the internet of things using deep learning approaches. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.

[18] Vishwakarma, R.; Jain, A.K. A Honeypot with machine learning based detection framework for defending iot based botnet DDoS attacks. In Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 23–25 April 2019; pp. 1019–1024.

[19] Tzagkarakis, C.; Petroulakis, N.; Ioannidis, S. Botnet attack detection at the IoT edge based on sparse representation. In Proceedings of the 2019 Global IoT Summit (GIoTS), Aarhus, Denmark, 17–21 June 2019; pp. 1–6.

[20] Nguyen, H.-T.; Ngo, Q.-D.; Le, V.-H. IoT Botnet Detection Approach Based on PSI graph and DGCNN classifier. In Proceedings of the 2018 IEEE International Conference on Information Communication and Signal Processing (ICICSP), Singapore, 28–30 September 2018; pp. 118–122.

[21] Nomm, S.; Bahsi, H. Unsupervised anomaly based botnet detection in IoT networks. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 1048–1053.

[22] Kumar, A.; Lim, T.J. Edima: Early detection of IoT malware network activity using machine learning techniques. In Proceedings of the 2019 IEEE 5thWorld Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019; p. 289.

[23] Liu, J.; Liu, S.; Zhang, S. Detection of IoT botnet based on deep learning. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8381–8385.

[24] Bahsi, H.; Nomm, S.; La Torre, F.B. Dimensionality reduction for machine learning based IoT botnet detection. In Proceedings of the 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 18–21 November 2018; pp. 1857–1862.

[25] Li,W.; Jin, J.; Lee, J.-H. Analysis of botnet domain names for IoT cybersecurity. IEEE Access 2019, 7, 94658–94665.

[26] Nguyen, H.-T.; Nguyen, D.-H.; Ngo, Q.-D.; Tran, V.-H.; Le, V.-H. Towards a rooted subgraph classifier for IoT botnet detection. In Proceedings of the 2019 7th International Conference on Computer and Communications Management, Bangkok, Thailand, 27–29 July 2019; pp. 247–251.

[27] Alazzam, H.; Alsmady, A.; Al Shorman, A. Supervised detection of IoT botnet attacks. In Proceedings of the Second International Conference on Data

Science, E-Learning and Information Systems, Dubai, United Arab Emirates, 2–5 December 2019; p. 42.

[28] Salim, M.M.; Park, J.H. Deep Learning based IoT re-authentication for botnet detection and prevention. In Advanced Multimedia and Ubiquitous Engineering; Springer: Singapore, 2019; p. 239.

[29] Nguyen, H.-T.; Ngo, Q.-D.; Le, V.-H. A novel graph-based approach for IoT botnet detection. Int. J. Inf. Secur. 2019, 19, 567–577.

[30] Al Shorman, A.; Faris, H.; Aljarah, I. Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection. J. Ambient Intell. Humaniz. Comput. 2020, 11, 2809–2825.

[31] Javed, Y.; Rajabi, N. Multi-layer perceptron artificial neural network based IoT botnet traffic classification. In Advances in Intelligent Systems and Computing; Springer Science and Business Media LLC.: Cham, Switzerland, 2019; pp. 973–984.

[32] Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Slay, J. Towards developing network forensic mechanism for botnet activities in the IoT based on machine learning techniques. In Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering; Springer Science and Business Media LLC.: Cham, Switzerland, 2018; pp. 30–44.

[33] Shire, R.; Shiaeles, S.; Bendiab, K.; Ghita, B.; Kolokotronis, N. Malware squid: A novel iot malware traffic analysis framework using convolutional neural network and binary visualisation. In Internet of Things, Smart Spaces, and Next Generation Networks and Systems; Springer: Cham, Switzerland, 2019; Volume 11660, pp. 65–76.

[34] Habib, M.; Aljarah, I.; Faris, H.; Mirjalili, S. Multi-objective Particle Swarm Optimization for Botnet Detection in Internet of Things. In Algorithms for Intelligent Systems; Springer Science and Business Media LLC.: Singapore, 2019; pp. 203–229.

[35] Jung, W.; Zhao, H.; Sun, M.; Zhou, G. IoT botnet detection via power consumption modeling. Smart Health 2020, 15, 100103.

[36] Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. Future Gener. Comput. Syst. 2019, 100, 779–796.

[37] Shafiq, M.; Tian, Z.; Sun, Y.; Du, X.; Guizani, M. Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city. Future Gener. Comput. Syst. 2020, 107, 433–442.

[38] Pour, M.S.; Mangino, A.; Friday, K.; Rathbun, M.; Bou-Harb, E.; Iqbal, F.; Samtani, S.; Crichigno, J.; Ghani, N. On data-driven curation, learning, and analysis for inferring evolving internet-of-Things (IoT) botnets in the wild. Comput. Secur. 2020, 91, 101707.

[39] Karanja, E.M.; Masupe, S.; Jeffrey, M.G. Analysis of internet of things malware using image texture features and machine learning techniques. Internet Things 2020, 9, 100153.

[40] Spaulding, J.; Park, J.; Kim, J.; Nyang, D.; Mohaisen, A. Thriving on chaos: Proactive detection of command and control domains in internet of things-scale botnets using DRIFT. Trans. Emerg. Telecommun. Technol. 2018, 30, e3505.

[41] Sagirlar, G.; Carminati, B.; Ferrari, E. AutoBotCatcher: Blockchain-based P2P botnet detection for the internet of things. In Proceedings of the 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), Philadelphia, PA, USA, 18–20 October 2018; pp. 1–8.

[42] Falco, G.; Li, C.; Fedorov, P.; Caldera, C.; Arora, R.; Jackson, K. Neuromesh: Iot security enabled by a blockchain powered botnet vaccine. In Proceedings of the International Conference on Omni-Layer Intelligent Systems, Crete, Greece, 5–7 May 2019; pp. 1–6.

[43] Ozawa, S.; Ban, T.; Hashimoto, N.; Nakazato, J.; Shimamura, J. A study of IoT malware activities using association rule learning for darknet sensor data. Int. J. Inf. Secur. 2019, 19, 83–92.

[44] Hashimoto, N.; Ozawa, S.; Ban, T.; Nakazato, J.; Shimamura, J. A darknet traffic analysis for IoT malwares using association rule learning. Procedia Comput. Sci. 2018, 144, 118–123.

[45] Özçelik, M.; Chalabianloo, N.; Gür, G. Software-defined edge defense against IoT-based DDoS. In Proceedings of the 2017 IEEE International Conference on Computer and Information Technology (CIT), Helsinki, Finland, 21–23 August 2017; p. 308.

[46] Yin, L.; Luo, X.; Zhu, C.; Wang, L.; Xu, Z.; Lu, H. ConnSpoiler: Disrupting C\&C communication of IoT-based botnet through fast detection of anomalous domain queries. IEEE Trans. Ind. Inform. 2019, 16, 1373–1384.

[47] Wazzan, M.; Algazzawi, D.; Bamasaq, O.; Albeshri, A.; Cheng, L. Internet of Things Botnet Detection Approaches: Analysis and Recommendations for Future Research. Appl. Sci. 2021, 11, 5713. https://doi.org/10.3390/app11125713.

[48] Al-Hayajneh, A.; Bhuiyan, Z.A.; McAndrew, I. Improving Internet of Things (IoT) security with soft-ware-defined networking (SDN). Computers 2020, 9, 8.

[49] Saadi, Weam & Ibrahim, Hassan & Shyaa, Methaq & Stephan, Jane. (2020). IoT Botnet Detection: Challenges and Issues. Test Engineering and Management. 83. 15092 - 15097.

[50] K. Ansam, G. Iqbal, V. Peter, K. Joarder and A. Ammar, "A Novel Ensemble of Hybrid Intrusion Detection System for Detecting Internet of Things Attacks," Electronics, vol. 8, no. 11, p. 1210, 2019.

[51] A. N. Devi and K. P. M. Kumar, "Intrusion detection system based on genetic-SVM for DoS attacks," International Journal of Engineering Research and General Science, vol. 3, pp. 107-113, 2015.

[52] M. Almseidin, M. Alzubi, M. Alkasassbeh and S. Kovacs, "Applying Intrusion Detection Algorithms On The Kdd-99 Dataset," Production Systems and Information Engineering, vol. 8, p. pp. 51–67, 2019.

[53] I. Obeidat, N. Hamadneh, M. Alkasassbeh, M. Almseidin and M. AlZubi, "Intensive Pre-Processing of KDD Cup 99 for Network Intrusion Classification

Using Machine Learning Techniques," International Journal of Interactive Mobile Technologies, vol. 13, no. 1, 2019.

[54] C. Li, W. Jiang and X. Zou, "Botnet: Survey and case study," in 2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC), 2009.

[55] M. Alauthman, N. Aslam, M. Al-kasassbeh, S. Khan and K.-K. R. Choo, "An efficient reinforcement learning-based Botnet detection approach," Journal of Network and Computer Applications, vol. 150, no. 15, 2020.

[56] N. Koroniotis, N. Moustafa, E. Sitnikova and B. Turnbull, "Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset," Future Generation Computer Systems, vol. 100, pp. 779-796, 2019.

[57] Sunanda Gamage, Jagath Samarabandu, Deep learning methods in network intrusion detection: A survey and an objective comparison, Journal of Network and Computer Applications, Volume 169, 2020, 102767, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2020.102767.

[58] W. Jo, S. Kim, C. Lee, and T. Shon, “Packet Preprocessing in CNN-based network intrusion detection system,” Electronics, vol. 9, no. 7, p. 1151, 2020.

List of Publications

Paper	Title. Author list. Conference/Journal	Status
Paper 1	**A survey on IoT Botnets and their Detection Approaches.** Karamveer, Rajesh Kumar Yadav. International Conference on Advances in Computing, Communication Control and Networking (ICAC3N–22)	Accepted