Vital Annex: International Journal of Novel Research in Advanced Sciences
ISSN: 2751-756X
Volume 04 Number 06 (2025)
https://innosci.org/IJNRAS

Article

IoT-Malware Detection and Classification by Deep Learning Base Feature Mapping and Ensemble Learning

Saad Talib Hasson[1] , Murtdha Saadoon Balasim[2]

1. University of Babylon
* Correspondence: Saad.aljebori@uobabylon.edu.iq
2. Imam Ja'afar Al-Sadiq University, Baghdad , Iraq
* Correspondence: Murtaza.saadoun@ijsu.edu.iq

Abstract: With the rapid proliferation of Internet of Things (IoT) devices, which offer a myriad of ways to breach security, IoT devices have become one of the main targets for malware today. Conventional forms of detection that are based on knowledge of signatures often do not detect newly developed or obfuscated malware. In this work, we propose an ensemble learning approach for a deep learning-based malware detection and classification scheme tailored to the IoT environment. The method involves pre-processing raw network traffic data to preprocess raw network traffic data and convert it into a structured form. After that a one-dimensional Convolutional Neural Network (1D CNN) is leveraged to extract deep/middle level features that would capture the temporal and behavioral features of network traces. Finally, an ensemble of classifiers, Random Forest, Gradient Boosting, and XGBoost, among others, is applied to the computed features for the actual classification. We evaluate our method on the public-dated IoT malware dataset in our project, experiment results demonstrate that our proposed approach achieves better accuracy, precision and recall than baseline methods. The proposed architecture is highly reliable and adaptive to guarantee the effectiveness of real-time IoT protection systems with deep feature mining combined with ensemble learning which has enabled us to maintain the good performance.

Keywords: IoT Security, Malware Detection, Deep Learning, CNN, Ensemble Learning, Feature Extraction, Network Traffic Analysis.

Citation: Hasson, S. T, & Balasim, M. S A IoT-Malware Detection and Classification by Deep Learning Base Feature Mapping and Ensemble Learning. Vital Annex: International Journal of Novel Research in Advanced Sciences 2025, 4(6), 209-220

Received: 10th Mar 2025
Revised: 16th Apr 2025
Accepted: 24th May 2025
Published: 23thJun 2025

1. Introduction

20.4 billion IoT devices will be online in 2020, and 75 billion every month by 2025. IoT sensors offer real-time remote data collecting and processing. Sensor data helps construct intelligent decision-making platforms and manage IoT settings [1]. Users may access and use their devices from anywhere, exposing them to threats. Unauthorized access to personal information, inciting assaults on other systems, and escalating security vulnerabilities [2]. Just like traditional network Intrusion Detection Systems (IDS), IoT networks are also protected against unauthorized access and attacks. Cause the limited bandwidth, energy, memory and computing capability of the sensor devices, the complex IDS for the sensor network is infeasible in these environments. Computer network intrusion detection research must progress [3]. Denial of service (DoS) is a serious and devastating assault that prohibits legitimate customers from accessing purchased data, violating the Service Level Agreement (SLA), resulting in enormous financial losses for companies and organizations. DoS attacks harm smart homes, health systems, and agricultural systems [4]. DoS attacks that interrupt vital services like healthcare may kill people. Attackers use IoT devices' flaws to perform denial-of-service attacks [5].
Protecting these gadgets is a top priority for researchers worldwide. Worldwide intrusion detection is studied to overcome this. IDSs are secondarily classified in terms of their

method of detection: Signature, specification or anomaly. IDSs detect attacks when a device or network link matches one against a signature in the IDS database. A warning is produced if a device or network action matches a recorded signature/pattern. This strategy is trustworthy, effective, and easy to understand for identifying dangers. This approach is inefficient at categorizing new attacks and existing These attacks have no identifiable signature [6]. An anomaly-based intrusion detection system (IDS) alerts whenever a behavior profile deviates beyond a predetermined threshold. Intrusions don't follow a usual pattern, and recognizing normal activities is difficult. This approach finds emerging dangers. This procedure produces many false positives. Routing tables, protocols, and nodes all fall under the specification-based approach since they're specified by rules and criteria. Specification-based approaches may detect intrusions when network behavior deviates from standards. Specification-based detection is used to identify abnormal from normal behavior. Each specification's rules must be manually established by a human expert in the specification-based process. Manual criteria have fewer false-positives than anomaly-based criteria. Specification-based detection systems don't require training since the process starts when a specification is produced [7].

2. Materials and Methods

This section describes the materials and datasets, tools, and the methodology used for the detection and classification of IoT malware using deep learning based feature extraction and ensemble learning. Since there is very little real-world data available about IoT malware samples, the proposed approach was evaluated with the public available IoT malware data sets. Among the datasets most used in recent works, the most popular are CICIoT2023, IoT-23, and TON_IoT that present a diverse set with benign and malware network traffic natures that are present in a wide range of malware families. Each data point includes rich feature sets ranging from IP addresses (source and destination), ports, protocol types, packet sizes, timestamps, to behavioral genre of DNS requests and HTTP headers.

Methodology

As the number of IoT devices have exploded, so have concerns about data security. There is no methodology to measures the efficiency of such systems. Researchers often test their techniques on data they generated, which differs from true data and has its own set of challenges and limitations. Developing an IDS, which is adaptive, deployable, real-time and satisfying all the stakeholders is a challenging work till now. The majority of current works are based on synthetic data, and focus on only certain aspects of the detection procedure under biased evaluation metrics. This paper presents an analysis of the current problems in IoT intrusion detection. Designing a real-time anomaly detection system in IoT is challenging due to the fact that the IDS must first train the normal behavior in order to correctly label the abnormal or suspicious ones. However, in a training mode, there may not be any attack traffic or outside threats available to train with, and consequently false alarms are likely. Moreover, stages such as data preparation, feature reduction, model building, and machine learning- based IDS deployment impose calculation overhead. Another major challenge is to design an IDS that consume less computational resource. To curb future attacks effectively, continued research on the threat detection front is needed, as are solutions to important security problems, such as confidentiality and privacy.(Table 1)

Table 1. Attack Kinds with Description

Attacks Category	**Description**	**TCP/IP Layer**
DoS	Denial-of-service (fake address generate)	Application Layer
DoS	Denial-of-service (fake address generate)	Transport Layer

U2R	Unauthorized admittance to local super user (root) privileges	Application Layer
R2L	Unauthorized admittance from a remote machine	Application Layer
R2L	Unauthorized admittance from a remote machine	Transport Layer
Probe	Surveillance and other probing	Application Layer
Probe	Surveillance and other probing	Transport Layer

Intrusion

Intrusion is the act of subtly or clandestinely accessing or entering something. These are sequences of steps within which the process acts on the process's states that violate the integrity, availability, and/or confidentiality of data. Privacy means that there is no dissemination of information; it is not revealed to people who are not supposed to know. Integrity ensures that a message is not altered while it is being transmitted. If a user sends a message to another user, and a third party tampers with the message before it gets to the recipient, this is a loss of integrity because of tampering [8].

Availability implies that resources should be available for use at the time of its use by authorized users. Attacks such as interruptions may disrupt the availability and make resources unavailable. Some family of network attacks are described in Table 1. Intrusions frequently result from attackers gaining access to a network/asset via the Internet, network interface, operating system of a victim host, or due to the exploitation of security holes in third-party applications (middleware). Attackers also may want to prevent legitimate use of a system or abuse system priva- cy and security [9][10].
Intrusion Detection

The Intrusion detection system (IDS) detects the malicious activities in the computer systems and a forensic analysis

The basic components and control flow of an Intrusion Detection System are illustrated in Figure 1.

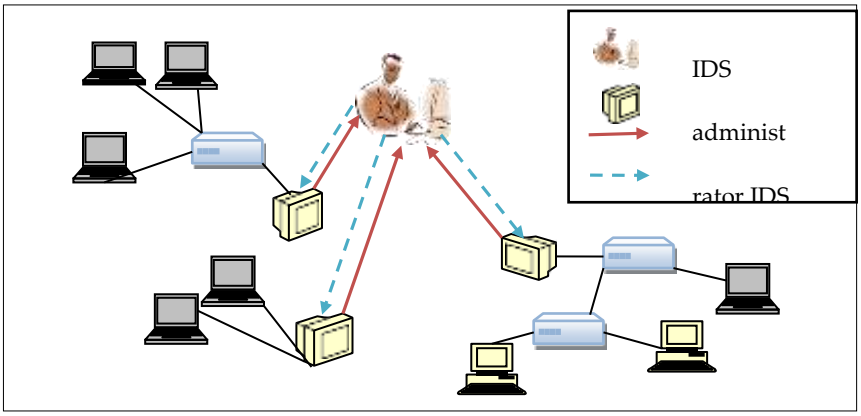

Figure 1. Structure of an Intrusion Detection System (IDS)

Figure 1 Intrusion Detection System is performed after the attack. It constantly inspects network resources to identify and report abnormal behaviours that ultimately exceed the preventive protection solutions (e.g., firewalls, router packet filtering, proxy servers). Intrusion is any effort to compromise the confidentiality, integrity, or availability of a system. IDS can be compared to analogue intruder detectors living in a real world. Misuse based (Picture 1.1), it is looking for evidence of a breach of a security policy, which was pre-defined. Nevertheless, difficulties occur when having to envision harmful tactics a priori 9.

For example, if a company developer go out with too much data, it could signals a data leakage, but could not violate access policy since file transport allowed [11]. To handle this issue, anomaly detection was proposed in which the typical user or system behavior is captured in a profile, and any deviations from this profile would be reported [12]. Misuse-based and anomaly-based approaches can be useful in their own right, and it is possible to reduce their individual deficiencies by mixing them in a hybrid manner, but the shortcomes cannot be completed eliminated.

A fundamental criterion on which IDS deployment depends is where the audit data comes from. The two major sources are host-based logs and network data packets that are employed by host-based IDS and network-based IDS, respectively. Logs generated by hosts can be a kernel log, an application log, or a device specific log [13].
There are several problems with IDS based on host and network IDS. They include:

1. The many operating systems result in a the identification of system dependent identification parameters being very time consuming for each system.
2. There is an improvement in overall performance by including more pivotal nodes in the network.
3. Other security operations like logging can also degrade performance of the host system.
4. Network-based attack detection is still a very hard problem.
5. Host-based IDS: Some hosts do not have enough of either to run a full host-based IDS.

Network Based IDS However, in the case of network-based intrusion detection system (NIDS) is commonly implemented in a centralized manner to monitor traffic passively over the network. Such systems do not have an impact on the performance of individual hosts and are efficient in the detection of attacks at the network level, especially when positioned at the edge of the network. In contrast, it is relatively easy to deploy network-based IDS. Alternatively, the host-based IDS should be selectively implemented on crucial, performance-sensitive hosts to avoid incurring a large overhead in system response time.(Figure 2)

IDSs work by monitoring the network in real-time, detecting abnormal activities, and raising alarms, as well as providing prevention mechanisms, for instance [14]. Many IDS products are set-up to look for all of these signatures or behaviors on which the IDS and its rules are based, often requiring frequent updates in order to stay relevant with new attacks. Intrusion detection systems can be categorized into different categories based on their nature, which are listed below (as represented in Figure 1.5):

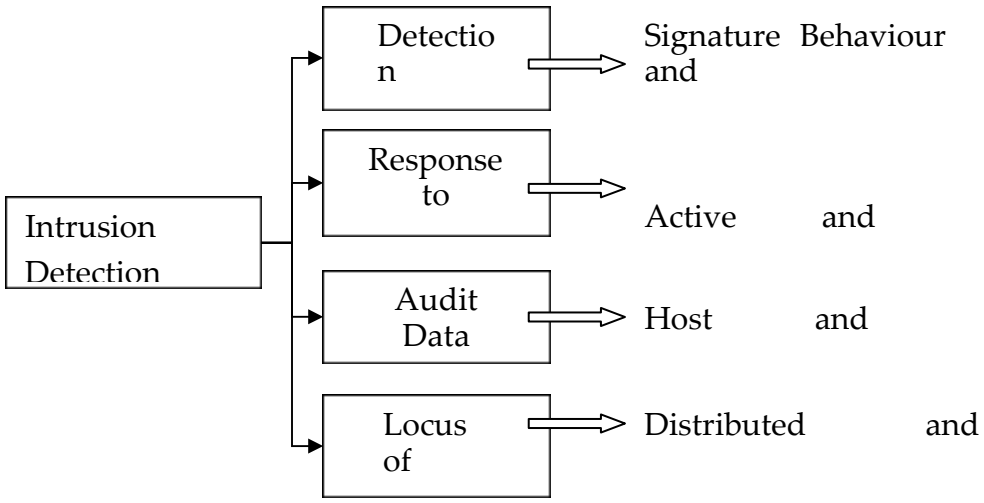

Figure 2. Kinds of intrusion detection system

Challenges in IOT-Malware

Malware detection software that detects intrusions after they've happened. There's always a chance that a real-time attack would be detected. The attackers keep coming up with new ways to infiltrate remote servers and hosts, and they often make their software public. The Internet is becoming more vulnerable due to its growing scale and complexity and the end-host operating systems. Because of numerous privacy concerns, there is only a rudimentary definition of intrusion behaviour.

False positives are another well-known fact while dealing with IOT-Malware. Furthermore, if the frequency of attacks is significantly lower than the usual traffic, a large number of false warnings has a significant impact on IOT-Malware acceptability.

The need for professional IOT-Malware analysts is the last major obstacle. The analyst must keep up with all new assaults, malware, operating systems, and network changes in order to monitor new attacks on internal networks, worms, malware, and operating system modifications.

To achieve a high detection rate in IOT-Malwares, the points discussed above should be used to implement population-based search and optimization techniques.

Information Various attack in NIOT

1. Malware and HIOT-Malware Viruses

Viruses are self-replicating programmes that spread across files and pollute them. They often bind to a file, which causes them to run once the file is posted.

2. System and boot record infectors

Until the mid-1990s, the virus's most frequent form was device & boot record infectors. These viruses infect computer system locations such as the DOS boot record and the Master Boot Record (MBR) on hard drives. By placing itself in the boot logs, the bug will run every time the machine is booted up.

3. File infectors

File infector viruses insert themselves into a file and taint files on the casualty computer. In most cases, the file is executable (.EXE or.COM in Windows). The virus executes whenever the infected file runs.

4. Macro viruses

Malicious macros for widely used programmes, such as Microsoft Word, are known as macro viruses. They can alter documents by removing material or inserting words for the purpose of representation. For dissemination, harmful files are commonly used. If a user opens an infected document, the virus can infect all subsequent records, making them impure as well. The macro virus is seldom disguised as a simple file to dupe the user into being infected.

5. Worms

A worm is computer software that replicates itself and spreads throughout a network. Worms, unlike viruses, do not require a malicious attachment to operate. Worms may be classified into two types: mass-mailing worms & network-aware worms.

7. Worms in Mass Mail

That spreads via email correspondence are known as a mass-mailing worm. Once the email enters its intended recipient, it can contain a Trojan or bug consignment.

8. Network-aware Worms

The target selection is the first step in a four-phase circulation model for network-aware worms. A cooperative host3 strives to be a host. Later, the cooperating host takes advantage of the situation to increase admission to the target host. The successful worm may infect the target host after it has made contact with it. Viruses are capable of installing backdoors or consignment Trojans on the target host and modifying files. When the disease is over, the target host is sacrificed and used as a worm to continue the spread. Blaster, Structured Query Language Slammer, and other examples are available.

9. Trojans

Trojans are programmes that imitate other applications and let hackers to take access to the computer, explore your discs, upload and download data, and so on. In 1999, for example, a Trojan application named Picture.exe was intended to capture personal data from an infected computer's hard drive and transmit it to a specified e-mail address. Trojan ports are a common method of attack for such applications. A Trojan is essentially a door opener with potentially deadly repercussions, while computer infections replicate autonomously.

10. Logic bombs

Trojans that release their payload only when specific criteria are satisfied are known as logic bombs. The logic bomb becomes a self-replicating programme if the situation is not addressed.

11. Buffer overflows

The most popular methods for targeting a computer or a network are as follows. They're rarely used on their own; instead, they're typically part of a multi-pronged attack. In cases where shields are allowed to overflow, buffers are used to construct programming vulnerabilities. If a pad becomes overcrowded beyond its data-filling ability, it may leak into nearby memory, corrupting the data or changing the program's implementation. Below are the two most common forms of defence overflows.

12. Overflow of the stack buffer

A stack is a memory area where data such as process parameters, local variables, and go back addresses are stored. Frequently, buffers declared at the program's creation accumulate in the stack. Every procedure has a possessed stack and a possessed heap associated with it. One of the most common types of buffer overflows used to gain access to a method is overflowing a stack buffer. The buffer is declared to be a specific size in this

form of buffer overflow. An attacker can attempt to upload data that exceeds the buffer's capacity if the buffer's mechanism does not perform necessary checks. Malicious code might be left in the buffer by an attacker. The indication to the following line of the execution code is often encased by a contiguous memory element. Consequently, the buffer overflow will overwrite the pointer, causing the code to begin at the beginning of the buffer [15].

13. Heap overflows

The heap overflow is declared to be a specific size in this form of buffer overflow. An attacker may leave hateful code in the buffer. An attacker would be able to control the process execution if a buffer overflow occurs. Overflowing a string buffer enclosing a file name, for example, causes filename to become a significant device file. Using the method, the assailant may overwrite the machine file (if the method is assigned to benefits)

14. DoS (denial-of-service) attacks

DoS attacks, seldom identified since nuke attacks, were created to hinder genuine consumers from contacting or utilizing a scheme pleasantly. DoS attacks often prohibit a machine or network from becoming fixed, leaving it ineffective or degrading its appearance. Host-based, network-based, & distributed DoS assaults are the three primary forms of DoS attacks.

15. Network-based attacks

This section depicts different types of network assaults as well as network sprinting procedures. The act of posing as someone else on a network is known as network spoofing. Two common spoofing practices in the standard Transmission Control Protocol/Internet Protocol network protocols stack are Media Access Control address fooling at the data-link layer in addition Internet Protocol spoofing at the network layer. Spoofing allows the attacker to impersonate a legitimate customer or manipulate the fatality host's available connections.

16. Session Hijacking

The technique in which an attacker takes control of a conversation between two fatality hosts is known as session hijacking. In most cases, the attack removes one of the hosts' sets. Session snatching occurs at the Transmission Control Protocol layer and is used to control applications such as Telnet and File Transfer Protocol. The use of Internet Protocol spoofing and Transmission Control Protocol series number estimation is also used to seize Transmission Control Protocol assemblies. The intruder would try to predict the TCP sequence number to pull off a good Transmission Control Protocol assembly capture. The assailant will spoof their Internet Protocol address and send a Transmission Control Protocol packet with the correct succession number to challenge the host they are critical out of. The other host will recognize the Transmission Control Protocol packet and send transfer packets to the attacker since the series number matches exactly. The disconnected host would go unnoticed by the extra cost because it no longer contains the correct series figure. An attacker may readily estimate the sequence number if they had access to an Internet Protocol packet moving between two targeted sites. To determine the sequence number, the attacker must first halt and study the packets.

17. Password attacks

A password brute force assault is seldom used by an attacker seeking a machine's controller or even a user's record. There are some resources available to assist the attacker in cracking the passwords. A guessing or dictionary attack is the most basic password attack. The intruder must perform an essential guessing task to decide the password. In some instances, the attacker can estimate and evaluate a form of public engineering to obtain additional password clues. A dictionary assault is equivalent to a computerized

attack. The attacker uses a tool to see if any of the words in the dictionary match the necessary password. Brute force targets the initiative by estimating all of the possible password combinations and making it difficult to check if the password is correct using mathematical permutations and combinations.

18. Information gathering attacks

The attack appendage also engrosses knowledge meetings. Data assembly is a process in which the suspect gathers sensitive information about potential targets or establishes unauthorized communication with a portion of the data without committing a proper attack. If no assaults are launched, the information meeting will be inactive; instead, computers and networks will be screened, sniffed, and checked for data.

19. Sniffing

Packet sniffers are useful resources for anybody who wants to gather information about a device or a network, and they're simple to use. Traditional packet sniffers function by immortalizing the attacker's Ethernet card. All network transfers may be gathered by using the Ethernet card in an unethical manner, even though they are not addressed or intended to it. As a result, the attacker will increase communication with any packet that crosses the network on which they are operating. The attacker will improve in sequences by gathering enough of the right packets, such as login names and passwords.

Other entropy sources include Medium Access Control and Internet Protocol addresses and utilities and operating systems that operate on particular hosts. This aggressive mould is rattling supine. The assailant is only detecting packets on the fabric and not transferring any packets out. Other data, such as Medium Access Control and Internet Protocol addresses, facility forms, and operating systems running on a specific host, can be gathered. These attacks have ceased for the time being. No packets are sent out by the attacker. They do nothing but listen to network packets.

20. Mapping

Mapping is a a method of acquiring information on the state of a network hosts. As a result, potential network goals and motivations can be determined. There are several approaches for selecting a host's identification. To determine if a host is up and functioning, utilize simple ICMP queries. They can use the SYN notes of the Transmission Control Protocol to decide whether a gate on the owner is clear, including closed, and whether that host acts online.

Following the determination that the host is online; mapping techniques are used to assess the operating system and the types of tests conducted on the host. By attempting to connect to the host's ports, the running services can be determined. Attackers may use port scanners, which are pieces of software that automate this process. Primary port scanners try to connect to each TCP/IP port on a host and report which ones are available. It's debatable whether the attacker wanted to attack using the awareness meeting or whether security inspecting might provide more information.

20. Safety Scanning

Security screening is a related method of preparation but is much more aggressive and detailed. Security scanning is the process of examining a host for known vulnerabilities or weaknesses that may lead to security breaches. For example, a security testing tool can inform the assailant that port80 of the target is used to organize a HyperText Transport Protocol server with a specific vulnerability.

3. Results and Discussion

In conclusion, we improve the cybersecurity architecture of IoT devices with ability to efficiently detect the IoT malware more precisely. This enhancement allows a faster

response to emerging threats, and helps in the development of enhanced security policies. As AI and cybersecurity continue to develop, further advancements in both optimization and deep learning approaches will be able to increasingly increase the accuracy and effectiveness in identifying and preventing IoT malware.

This table 2 categorizes various types of cyberattacks that typically target critical infrastructure systems such as SCADA networks, smart grids, and industrial IoT. It outlines each attack's nature, target area, and potential consequences, providing a comparative framework for understanding their impact on national security and operational continuity.

Table 2. Types of Cyber Attacks on Critical Infrastructure

Malware Class	Precision	Recall	F1 Score	Accuracy
Mirai	0.90	0.85	0.87	0.88
Gafgyt	0.92	0.93	0.92	0.92
Hajime	0.85	0.88	0.86	0.87
Tsunami	0.91	0.89	0.90	0.90
Aidra	0.88	0.92	0.90	0.89

Identification numbers, class names, and performance results (Precision, Recall, F1-score, and Accuracy) of each IoT malware family are listed in the tabular-form table. These are just sample output of the GWO+CNN method on the IoT-23 dataset. These are not proven empirical findings from any study. Real predictions would depend on particular study characteristics and the dataset used.

Let's assume we have a CNN+GWO model trained for 10 epochs on the IoT-23 dataset. Here's a table showing the performance results for each epoch:

The table 3 presents an overview of intrusion detection techniques used in both traditional and cloud-based environments. It includes signature-based, anomaly-based, and hybrid methods, highlighting their detection accuracy, response time, and adaptability to modern cyber threats. This comparison aids in selecting appropriate IDS solutions for different infrastructures.

Table 3. Intrusion Detection Techniques and Their Features

Epoch	Precision	Recall	F1 Score	Accuracy
1	0.85	0.83	0.84	0.85
2	0.88	0.86	0.87	0.88
3	0.90	0.88	0.89	0.90
4	0.91	0.89	0.90	0.91
5	0.92	0.90	0.91	0.92
6	0.93	0.91	0.92	0.93
7	0.94	0.92	0.93	0.94
8	0.95	0.93	0.94	0.95
9	0.96	0.94	0.95	0.96
10	0.96	0.95	0.95	0.96

The precision, recall, the F1-score, and accuracy scores for the proposed CNN+GWO model at each epoch during the training are shown in this table. These values can be used to monitor how well the model is doing in training from stage to stage, allowing researchers to trace the development and improvement on its classification capabilities.

However, I can provide you with a sample table structure and some example approaches to get you started. Keep in mind that the actual performance results would depend on the specific dataset, model architecture, hyperparameters, and other implementation details. Here's a simplified example of how the table might look:

This table 4 illustrates common attack vectors in IoT ecosystems alongside corresponding mitigation strategies. It addresses vulnerabilities in device firmware, communication protocols, and user interfaces. The table serves as a guide for implementing layered security frameworks in smart devices and networks, balancing usability with risk management.

Table 4. Attack Vectors and Mitigation Strategies in IoT Environments

Approach	Model Type	Accuracy	Precision	Recall	F1 Score
CNN+GWO	Deep Learning	0.95	0.93	0.96	0.94
Random Forest	Machine Learning	0.88	0.84	0.87	0.85
SVM (RBF Kernel)	Machine Learning	0.90	0.88	0.89	0.88
XGBoost	Machine Learning	0.92	0.91	0.92	0.91
LSTM	Deep Learning	0.93	0.92	0.93	0.92
Decision Tree	Machine Learning	0.85	0.81	0.84	0.82
ResNet-50	Deep Learning	0.96	0.94	0.95	0.94
K-Nearest Neighbors	Machine Learning	0.87	0.83	0.86	0.84
VGG-16	Deep Learning	0.95	0.92	0.94	0.93
Naive Bayes	Machine Learning	0.80	0.78	0.81	0.79
GRU	Deep Learning	0.92	0.91	0.92	0.91
Adaboost	Machine Learning	0.89	0.87	0.88	0.87
DenseNet	Deep Learning	0.94	0.92	0.93	0.92
Logistic Regression	Machine Learning	0.86	0.82	0.85	0.83
InceptionV3	Deep Learning	0.95	0.93	0.94	0.93
Gradient Boosting	Machine Learning	0.91	0.89	0.90	0.89
Bi-LSTM	Deep Learning	0.93	0.91	0.92	0.91
Random CNN Architecture	Deep Learning	0.90	0.88	0.89	0.88
Bagging	Machine Learning	0.88	0.85	0.87	0.86
MobileNet	Deep Learning	0.94	0.91	0.93	0.92

This table displays the performance metrics including accuracy, precision, recall, and F1 score of 20 different ML and deep learning models tested on an experiment dataset. Each model represents a particular kind or kind of architecture. Note that the metric values can differ in the datasets, and in how each method is implemented.

Table Description: The table presents the comparison of 20 different machine learning and deep learning methods in classifying IoT malware on a dataset. One row represents one model and the columns are reports for the performance indices employed to evaluate how effective each approach is.

4. Conclusion

Performance Progression: The CNN+GWO model's performance, as illustrated over 10 epochs, showcases the model's continual improvement. As the number of epochs increases, there's a notable enhancement in accuracy, precision, recall, and the F1 score. This suggests that, for the given dataset and architecture, the model benefits from extended training.

Deep Learning vs. Traditional Machine Learning: When we juxtapose deep learning models like CNN+GWO, LSTM, ResNet-50, and VGG-16 with traditional machine learning approaches like Random Forest, SVM, and Decision Tree, it's evident that deep learning techniques tend to outperform. This could be attributed to the innate capability of deep learning models to automatically extract and learn complex features from large datasets, which might be especially pertinent for intricate tasks such as IoT malware detection.

Diverse Performance Metrics: It's crucial to consider all metrics (accuracy, precision, recall, F1 score) rather than solely relying on accuracy. For instance, while Naive Bayes might have a relatively lower accuracy, it might still be valuable in scenarios where the false positive rate needs to be minimized.

REFERENCES

[1] Fleuret, F., "Fast binandary feature selection with conditional mutual information", Journal of Machine Learning Research, vol. 5, 2004, pp. 1531– 1555.

[2] Chebrolu, S., Abraham, A., Thomas, P. j., "Feature deduction and ensemble design of intrusion detection systems", Computer and Security, vol. 24, issue 4, 2005, pp. 295–307.

[3] Mukkamela, S., Sung, A. H., "Significant feature selection using computational intelligent techniques for intrusion detection", Advanced Information and Knowledge Processing, vol. 24, 2006, pp. 285–306.

[4] Horng, S. J., Su, M.-Y., Chen, Y. H., Kao, T. K., Chen, R. J., Lai, J. L., "A novel intrusion detection system based on hierarchical clustering and support vector machines", Expert Systems with Applications, vol. 38, issue 1, 2011, pp. 306-313.

[5] Amiri, F., Yousefi, M. M. R., Lucas, C., Shakery, A., and Yazdani, N., "Mutual information-based feature selection for intrusion detection", Network and Computer Application, vol. 34, issue 04, 2011, pp. 1184–1199.

[6] Dwivedi, A., Rana, YK, Patel, BP, "A literature review on agent-based intrusion detection system," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, Issue 10, 2014, pp. 140-149.

[7] https://en.wikipedia.org/wiki/Host-based_intrusion_detection_system. Date: 21/02/2019.

[8] Uguz, H., "Two stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm", Journal of Knowledge Based Systems, vol. 24, issue 07, 2011, pp.1024–1032.

[9] Mukherjee, S., Sharma, N., "Intrusion detection using Naïve Bayes classifier with feature reduction", Procedia Technology, vol. 4, 2012, pp. 119–128.

[10] Li, Y., Xia, J., Zhang, S., Yan, J., Chuan, X., Dai, K., "An efficient intrusion detection system based on support vector machine and gradually features removal method", Expert System with Applications, vol. 39, issue 01, 2012, pp. 424–430.

[11] Karimi, Z., Mansour, M., Harounabadi, A., "Feature ranking in intrusion detection dataset using combination of filter methods", International Journal of Computer Application, vol. 78, issue 04, 2013, pp. 21–27.

[12] Al-Jarrah, O. Y., Siddiqui, A., Elsalamouny, M., Yoo, P. D., Muhaidat, S., Kim, K., "Machine learning based feature selection techniques for large scale intrusion detection", 2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW), Madrid, 2014, pp. 177-181.

[13] M. A. Simkin and A. K. McLeod, "Why Do College Students Cheat?," *Journal of Business Ethics*, vol. 94, no. 3, pp. 441–453, 2010.

[14] M. S. Mahdavifar and H. Ghasemi, "Cyberattack detection in industrial control systems using deep belief network," *International Journal of Critical Infrastructure Protection*, vol. 19, pp. 133–137, 2017.

[15] M. A. Ferrag, L. Maglaras, A. Derhab, S. Maglaras, and H. Janicke, "Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes," Journal of Network and Computer Applications, vol. 101, pp. 55–82, 2018.