Malware Cyber Threat Intelligence System for Internet of Things (IoT) Using Machine Learning

Peng Xiao

Information Center of Yunnan Power Grid Co., Ltd., Kunming, 650000, Yunnan, China
E-mail: xiaopeng202112@163.com

Received 05 December 2022; Accepted 27 February 2023;
Publication 05 December 2023

Abstract

Cyber Intelligence (CI) is a sophisticated security solution that uses machine learning models to protect networks against cyber-attack. Security concerns to IoT devices are exacerbated because of their inherent weaknesses in memory systems, physical and online interfaces, and network services. IoT devices are vulnerable to attacks because of the communication channels. That raises the risk of spoofing and Denial-of-Service (DoS) attacks on the entire system, which is a severe problem. Since the IoT ecosystem does not have encryption and access restrictions, cloud-based communications and data storage have become increasingly popular. An IoT-based Cyber Threat Intelligence System (IoT-CTIS) is designed in this article to detect malware and security threads using a machine learning algorithm. Because hackers are continuously attempting to get their hands on sensitive information, it is important that IoT devices have strong authentication measures in place. Multifactor authentication, digital certificates, and biometrics are just some of the methods that may be used to verify the identity of an Internet of Things device. All devices use Machine Learning (ML) assisted Logistic Regression

Journal of Cyber Security and Mobility, Vol. 13_1, 53–90.
doi: 10.13052/jcsm2245-1439.1313

(LR) techniques to address memory and Internet interface vulnerabilities. System integrity concerns, such as spoofing and Denial of Service (DoS) attacks, must be minimized using the Random Forest (RF) Algorithm. Default passwords are often provided with IoT devices, and many users don't bother to change them, making it simple for cybercriminals to get access. In other instances, people design insecure passwords that are easy to crack. The results of the experiments show that the method outperforms other similar strategies in terms of identification and wrong alarms. Checking your alarm system's functionality both locally and in terms of its connection to the monitoring centre is why you do it. Make sure your alarm system is working properly by checking it on a regular basis. It is recommended that you do system tests at least once every three months. The experimental analysis of IoT-CTIS outperforms the method in terms of accuracy (90%), precision (90%), F-measure (88%), Re-call (90%), RMSE (15%), MSE (5%), TPR (89%), TNR (8%), FRP (89%), FNR (8%), Security (93%), MCC (92%).

Keywords: Cyber threat, internet of things, machine learning, decision tree classification.

1 Introduction to Security Threats and Malware Detection

To enable Internet of Things (IoT) services for end-users and enterprises, the proliferation of IoT devices must consider the changing connections among Space, Air, Ground, and Sea (SAGS) networks. IoT systems significantly impact lives by providing automatic solutions to businesses and end customers. People can now incorporate and link physical objects, such as drones, cars, and other related software, to the Web and operate them virtually via cloud-based platforms. By enabling connectivity in more locations on the planet, IoT has lately provided a complete representation of the natural world [1]. Many cyber threat intelligence systems rely heavily on signature-based detection of malware, which is easily evaded by attackers who modify their malware to create new variants.

- Overconfidence in the system's ability to detect threats or missed threats due to a high false negative rate is two potential outcomes of using a malware cyber threat intelligence system.
- Attackers may be able to exploit blind spots in malware cyber threat intelligence systems due to their limited visibility into a subset of networks and/or systems.

- Some malware cyber threat intelligence systems may have an incomplete picture of attacker behaviour and tactics, leading to missed threats.
- It's possible that some malware cyber threat intelligence systems don't have complete data because they don't have access to all relevant data sources, such as dark web forums or underground marketplaces.
- Some malware cyber threat intelligence systems may have bias in the data they collect and analyse, giving an inaccurate picture of the threat landscape.

Some malware cyber threat intelligence systems may not be well integrated with other security systems like firewalls or intrusion detection systems, reducing their ability to detect and respond to threats. IoT has helped businesses and consumers in a wide range of ways, including increasing application accessibility, strengthening performance levels, lowering costs, and facilitating improved decision-making [2]. It is anticipated that 50 million network users by 2025, with the full potential of the IoT reaching $15 trillion by 2025 [3], because many firms have converted their businesses to integrate IoT-SAGS technology.

Modern computer technology and the Internet have made life simpler and more accessible for people. Nowadays, anything can be done online, including social contact, financial transactions, tracking different aspects of human physiology, etc. All of these advancements tempt criminals to conduct crimes online instead of in the real world.

Recent and commercial research claims that cyberattacks affect the worldwide economic vast amounts of money [4]. Malware is a standard tool used by online criminals to start attacks. Some software known as malware engages in unauthorized and suspicious actions on its victims' computers. The different varieties of malware include viruses, worms, Trojan horses, rootkits, malware, etc. Malware variations can steal security data, Distributed Denial of Service (DDoS) assaults, and cause havoc to computer networks. The latest malware types camouflage themselves in the targeted computer by encrypting data and stuffing it [5]. These novel varieties propagate by using people's trust as a vehicle for infection. For example, well-known virus distribution techniques include opening email messages, downloading false software, and accessing and file transfer from bogus websites. The malware affects the performance of the IoT systems, and the possibility of malware intrusion is higher in the IoT environment. So a better malware detection and classification model is needed to enhance the effectiveness of the entire IoT systems. One type of machine learning model used to identify

malware is the malware detection model. These models can be educated on data comprised of both safe and harmful program in order to pick up red flag characteristics. Examples of popular malware detection models include:

- Signature-based models, on the other hand, rely on predefined signatures or hash values of known malware to make that determination.
- Based on anomalies, these models look for unusual activity in software that could be malicious.
- These models, which are grounded in heuristics, rely on a predetermined set of rules to detect suspicious code.
- These models detect malware by analysing the data for tell-tale patterns and features using a variety of machine learning algorithms, such as decision trees, random forests, and neural networks.

It's worth noting that while these models can be very useful for detecting malware, they can also give false positives and might not be able to identify completely new forms of malware.

The research must find spyware as soon as it penetrates the computer networks to defend them. The process of evaluating a suspicious file to determine its maliciousness or benign nature is known as detecting attacks [6]. The classification of malware goes one step further. Describe the class or branch of malware used to classify the file once it was found malicious.

Cyber Threat Intelligence (CTI) is situational information about a risk that encompasses both low- and high-level indications such as Internet Protocols (IP), hashes, networking artifacts, and tactics, methods, and processes [7]. It has recently developed into an essential component of routine security operations, assisting organizations in prioritizing threats and quickly identifying, mitigating, or containing attacks. Nearly 60% of firms are already utilizing CTI, and 25% have intentions to soon include it in existing security management, as per a current survey [8]. It continues by stating that nearly 45% of these businesses have teams specifically tasked with implementing and maintaining CTI. Most prior research has concentrated on statistical and traditional Machine Learning (ML) for creating efficient Threat Intelligence (TI) models [9, 11]. Their approaches were problematic as a threat due to their excessive complexity, poor accuracy rate, and lack of generalization abilities.

The main contributions of this article are listed below:

- An IoT-based Cyber Threat Intelligence System (IoT-CTIS) is designed to detect malware and security threats using machine learning in IoT devices.

- A three-layer architecture is designed to enhance the. Security and reduce the computation complexity.
- CTI model is designed to extract and identify security threats and malware. The software results ensure the highest performance of the proposed model.

The remainder of the research is arranged as follows: Section 2 indicates the security threats and malware background. The proposed IoT-based Cyber Threat Intelligence System (IoT-CTIS) is designed, and algorithms are derived in Section 3. Section 4 analyses the performance of the proposed system, and the findings are contrasted with each other. Section 5 indicates the conclusion and future study of the research.

2 Background to the Malware Detection Models

The research takes time to identify and classify malware. In these stages, a variety of methods and techniques are employed. The malware must first be evaluated with appropriate technologies to be found. A tool data was recorded, and either manually or mechanically, features were retrieved. Data gathering techniques are employed to obtain important characteristics at this stage [12]. The retrieved features are then chosen based on a set of standards. Finally, to distinguish between malicious and benign variables, principal components are trained using Learning models or rule-based transfer learning [13]. A categorization is created by identifying the types and classifications of ransomware [14].

The examples are analyzed by Sikorski et al. to understand the characteristics and behaviors of the virus [15]. Spyware assessment is a critical step in the process. That is because malware is detected during the analytical process, and several questions, like how it is structured, how it spreads, and how much harm it has already done to the victims' computers, can be addressed. There are two categories of malware classification: static and dynamic model. Basic unit testing of malware is the first step, while complex dynamic analysis comes last [16]. Both manual and automated analysis is possible. While automatic classification demands significant data science coding abilities, the manual study includes subject expertise.

The use of computer understanding and big analytics to address the identified problems was the focus of many articles in the scientific community devoted to CTI. Kantarcioglu et al. [17] emphasize the importance of big data as a challenge for cyber security. According to Thuraisingham et al. [18],

computer security's troubles are broadly categorized as invasion mitigation and prevention, data truthfulness, policy-based communication, and risk-oriented safety metrics. They claim that the prospect of computer security lies in adopting a data-driven methodology. The report covers every category's current, highly advanced, and potential directions.

According to Harel et al. [19], the business has been preparing itself for the fight versus a proliferation of attackers by automating processes, collaborating with others, and implementing expert machines. The authors explicitly address guided and unstructured learning and provide examples of its usage in behavior analysis and outlier detection to identify a user's potential risk. The researcher describes a data-driven method to CTI in which they attempt to build a model for anticipating future attacks using the vulnerabilities exploitation information discovered on Twitter [20]. Another similar piece of research in this area is shown in [21], which advances the prediction models, notably by being diligent with the training sample.

Darabian et al. demonstrated an enhanced threat-hunting strategy for IoT computer viruses using a combination of extreme ML algorithms [22]. It performed reasonably well compared to other deep neural network models, like the stacking Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN). No testing utilizing performance measures proved the effectiveness of the authors' claim that it sped up the learning and detecting processes.

Similarly, the work in [23] suggested using ensemble techniques with a threat-hunting model to identify Windows, Android, and IoT ransomware. Its fundamental concept involved using a different classification classifier depending on standard machine learning for every extracted features, that changed into the original space. Each learned system was viewed as a membership value determining how closely a pattern fit into a given class. Additionally, a weighted assignment was utilized to assess the significance of every classification system on a given feature. The suggested model showed strong performance and effectiveness, F-measure, and higher accuracy. As these ensemble systems usually have poor interoperability, it can be challenging to draw meaningful conclusions.

Many works have been done on outlier detection using ML algorithms, supervised and unsupervised methods, and various fields. By identifying their examples throughout the learning phase, machine learning can obtain high efficiency in traffic classification using Bayesian Neural Networks with or without input from the application. Balagani et al. [24] used the k-means method for unlabeled data to increase classification precision for

outlier detection. Additionally, a classifier that learns incrementally is the best choice for acquiring the necessary information for continuous or flow data. Classifying a huge dataset presents challenges for the authors [25]. Because they create high accuracy, sensitivities, and specificity, they constructed three classification algorithms for binary categorization of internet information (normal or virus). They feel that this is applicable to be utilized in anomalous detection systems. The different classifiers majorly used for malware classifications are Support Vector Machine (SVM) [27], CNN [28], Decision Tree (DT) [29], Linear Discrimination Analysis (LDA) [30], and Naïve Bayes (NB) [31].

Cyber threat intelligence (CTI) technology is an evidence-based defensive architecture that allows for the proactive response to sophisticated cyber threats via the monitoring and sharing of security-related information across sectors. Misguided security rules might have a major impact on how well CTI systems work [32]. Numerous recent and ongoing difficulties associated with cyber threats are discussed, and the most recent case study is analysed. Last but not least, we've presented a novel system on Collaborative Cyber Threat Information Exchange (CCTI), highlighting the ability of a larger community to help identify problems through a computer essence centred on artificial intelligence and block chain.

Industry 4.0, the fourth stage of the industrial and manufacturing revolution, is distinguished from previous revolutions by its provision of Internet-connected smart systems, such as automated factories, organisations, development on demand, and 'just-in-time' development. Cyber-physical systems (CPSs), the Internet of Things (IoT), and the cloud and fog computing paradigms are all part of the integration that defines Industry 4.0, which aims to create intelligent machines, buildings, and communities [33]. Data from and to sensors and actuators, as well as other network traffic, fall under this category. The suggested threat intelligence method utilises beta mixture-hidden Markov models (MHMMs) to unearth unusual actions taken against both physical and networked systems. The CPS dataset of sensors and actuators, as well as the network traffic statistics, are used to assess the system. In comparison to five similar mechanisms, the suggested method performs better, indicating its potential as a practical approach to deploying Industry 4.0 in the real world.

- There are some cyber threats that cannot be detected by CPS because they fall outside the data collection parameters.
- As a result, it is possible that CPS will miss sophisticated attacks that employ cutting-edge methods like zero-day exploits.

- The possibility exists that encrypted network traffic will evade CPS's threat detection capabilities.
- Internal threats may go undetected by CPS.
- On a standalone system, CPS might not be able to spot suspicious activity.
- If a threat is constantly adapting, CPS might not be able to detect it.
- Unfortunately, CPS may miss some forms of malicious activity when monitoring in real time.

Interacting physical and computational elements form the basis of Cyber-Physical Systems (CPS). Control systems, medical devices, and industrial automation are a few of the many places you can find CPS. In spite of this, these systems encounter a number of obstacles, including:

- Cyber-Physical Systems are difficult to design and manage because of their complexity, as they consist of many interconnected parts.
- CPS is susceptible to cyber-attacks that can compromise their operation and integrity.
- There are strict real-time performance requirements that must be met by CPS in order for them to function properly.
- Compatibility with other systems or Interoperability is essential for CPS because it mandates the sharing of data and information with other systems.
- Maintaining proper operation under any and all conditions necessitates that CPS be both reliable and fault-tolerant.
- The confidentiality of individual information is a top priority for CPS.

These obstacles should be discussed in the introduction and literature section of CPS research manuscripts to set the stage for the work being presented.

Even though evasion attempts can fool the supervised neural detection method, which is extremely good at detecting viruses, misclassification can result. In an adversarial assault, attackers only change a small number of specified bytes in malware programs to get past the detection technique. The relevant contributions are made in the suggested malware categorization model to lessen the impact of evasion assaults.

3 Proposed IoT-Based Cyber Threat Intelligence System

The research explains the solution and execution for malware identification and classification in this section. This section is further segmented into two

subcategories; the first describes the procedure and design considerations in learning and creating the prototype, and the second describes the manufacturing that utilizes this prototype to extract pertinent information and data before stashing and distributing it. Since the critical emphasis is on the Natural Language Processing (NLP) based teaching of a prototype for retrieving CTI from written data, this segment is further divided into two subcategories.

3.1 CTI Extraction

By analysing human language data, computers can have more natural conversations with people and increase the efficiency of other language-based processes. Natural language processing (NLP) enables computers to do things like read text, understand voice, evaluate tone, and prioritise information. Extraction of high-level CTI signals from unorganized data was a vital goal of the study. To that purpose, it employed the Known Entity Identification Natural Language Processor approach. An essential part of identification is the process of removing names from text. The goal of the research was to identify critical Cyber Threat Intelligence (CTI) signals in a sea of unstructured data. The purpose was to detect and investigate potential security issues from vast amounts of unstructured data like text and logs.

The study used a Natural Language Processing (NLP) technique called Known Entity Identification to get the job done. Natural language processing, or NLP, is a subset of AI that focuses on how computers and people can communicate with one another using only language. NLP was used in this investigation to sort through the chaos of the data and pull out the useful CTI signals.

Known Entity Identification NLP sought to locate and extract relevant details from unstructured data, such as names, addresses, and dates. From there, this data was fed into a CTI signal generator for producing high-level signals for use in identifying and responding to security threats.

Extraction of high-level CTI signals from unstructured data relied heavily on the application of the Known Entity Identification NLP approach. To generate CTI signals for threat detection and response, natural language processing was used to sift through unstructured data and find relevant information.

The default object classifications in most NLP packages have been trained to identify classes as diverse as names, places, and organizations. NLP is so domain-specific that a method created for one area will almost certainly not work in another. No proof-of-concepts employed in this area have used and

produced a sequence pre-trained model exclusively for CTI. As a result of contextualising the training with practise exercises and real-world situations, the organisation is able to shift its attention from learning the processes involved to learning how those processes might benefit the business. It aids in the execution of the business process in whole or in part. Phishing is a kind of online fraud in which a computer pretends to be a person in order to steal sensitive information. Natural language processing (NLP) may be used to decode the bot or spam language used in the email. The underlying structure of the email itself may be analysed for clues about spammers and the contents they send. The training model provides a framework for this procedure and is input into the primary structure for additional processing. When applied to data in a particular field, the training model serves as a template for analysis e.g. natural language processing, image classification, etc. Machine learning algorithms are typically used as the main framework, as they can learn the underlying patterns and relationships in the data and then use that knowledge to make predictions or decisions when processing new data.

The statement "Natural Language Processor is so domain-specific that a method created for one area will almost certainly not work in another" implies that the methods and algorithms that are successful in processing text data in one domain such as social media may not be successful in processing text data in another domain e.g. scientific papers. This is due to the fact that there can be significant differences in language and setting between disciplines.

There are many potential applications for NLP in the security industry, including text classification and sentiment analysis to spot vulnerabilities, report summarization, and alert generation using natural language.

The three-tier architecture of the suggested IoT-CTIS system is designed and plotted in Figure 1. The IoT-CTIS system has three layers: the IoT layer, the Edge layer, and the Cloud layer. The input is classified into training and testing samples, and the machine learning model is used to classify normal and malicious samples. The necessary pieces are accessed from the cloud layer. One of the IoT layers, the Perception Layer, controls all of the intelligent nodes in the network. The Connectivity/Transport Layer facilitates information exchange between the cloud and connected devices, as well as between the cloud and the various gateways and networks. The hardware of your IoT devices, the embedded OS that controls the device's activities, and the device firmware that contains the software and instructions are all part of the edge layer of your IoT workload. Cloud layer that is either uniformly dispersed or intermittently scattered. The goal is a trained (fit) model that reliably extrapolates to novel, unanticipated inputs. To measure

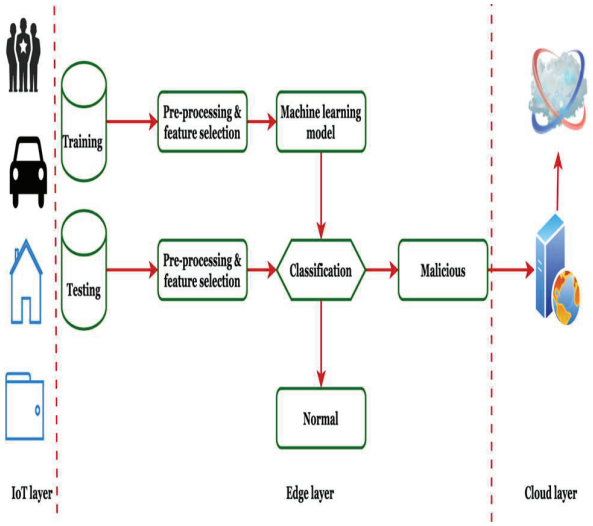

Figure 1 The three-tier architecture of the suggested IoT-CTIS system.

the fitted model's accuracy in categorizing new data, use "new" instances from the withheld datasets (validation and test datasets) to evaluate the model. An operation performed on data ahead of its main processing or subsequent analysis. Feature extraction is the procedure of converting raw data into a set of numerical features that can be processed without losing any of the information contained in the original data set. This term can be applied to any first or preparatory processing stage when multiple steps are required to prepare data for the user. When compared to using machine learning on unprocessed data, the outcomes are far more favourable.

Layered data collection and exchange is enabled by the Internet of Things (IoT), a network of physical devices such as sensors, cameras, and smart appliances. The edge layer sits between IoT gadgets and the cloud, where data is processed and analysed. The cloud infrastructure is a system of distributed servers used for scalable data storage and processing.

The input data e.g. network traffic, system logs collected from the IoT devices can be split into training and testing samples in the context of machine learning for security. The machine learning model is trained using the training samples, and its efficacy is then assessed using the testing samples.

There is a correlation between the training and test samples and the classification of the samples safe vs. harmful. To determine whether or not new data the testing samples is benign or malicious, the machine learning

model applies the patterns and relationships it has learned from the training samples.

Both pre-processing and feature extraction play crucial roles in any NLP system. In order to facilitate further processing, the raw data must be cleaned and formatted in the pre-processing stage. Activities like tokenization, lemmatization, and stop-word removal fall under this category. In the feature extraction phase, we pull actionable information from the cleaned and prepared data. That's why we need to do things like part-of-speech tagging, named entity recognition, and emotional analysis.

The proposed framework uses a three-tier setup, with the natural language processor located in the middle layer. The information gleaned from the first two layers' raw data is the domain of the third layer. The natural language processing system at this level would be in charge of the aforementioned pre-processing and feature extraction.

3.1.1 Collecting CTI data

It gathered several CTI documents from reputable security bloggers and danger findings from 4 sources, including FireEye, Kaspersky Safety Lab, and a compiled list of results from a Git repository that contained relevant data about cyber-attacks as applicant records for the test dataset. The selection of these source materials was based on several factors, including (i) how frequently they post case files, (ii) how thoroughly they are recognized to investigate dangers, and (iii) their standing in the data security sector as a result of their well-known investigative and defensive techniques.

3.1.2 Annotating CTI data

It gives an NLP model's supervised file containing continuous text labeled with the categories that need to be recognized to begin to spot the necessary phrases. It developed a web-based User Interface (UI) to construct such a database, making the manual annotating process simpler. It created a list of words that it wanted the models to be capable of understanding. To achieve this, it considered the extensive vocabulary and condensed the critical building pieces of victim targeted, resources, behavior, and desired effect to provide a robust dataset that contains:

- Actor: The hacking group responsible for a specific attack, such as the Carbanak cyber gang, etc.
- Targeted industry – The sector was the focus of the attack, such as the army, police, or financial organizations.

- Targeted position – The attack's precise geographical target, such as South Asia, Turkey, the US, etc., is analyzed.
- Intended impact – The attacker desired outcomes, such as information theft, monetary reward, or cyber espionage.
- Technique: The advanced methods the assailant uses, such as spear phishing messages, social control, watering holes, etc.
- The tool employed: The mechanisms that the assailant used, such as a backdoor, a reversed shell, a Mimi Katz, etc.
- Targeted implementation – The attackers hope to compromise programs, such as Microsoft Word, PowerShell, etc.

Other low-level indications, such as Internet Protocol (IP) addresses, passwords, domains, registry entries, files, and flaws, were added to the collection and were processed using regexes because they often follow a similar pattern. The annotation is created from the UI by choosing a set of words and assigning the appropriate label to it. The backend server ensures that the annotation document is prepared to be included in the test dataset when saved. It initially checks the entering text for Unicode letters and filters them out. Sentences are tokenized as the process' following phase, which yields a list of all the phrases in the text. Each term is tokenized by looping over the phrases so that each word is an individual token.

IP numbers and indications were tokenized correctly; the developers had to modify the tokenizer. The tokenized phrases are then Parts of Speech (POS) tagged; this means that the tags are applied according to the Part of Speech that each symbol in the sentence represents, such as NN for nouns, VB for verbs, etc. A label associated with each tokenized word indicates to the system if or not it wants to be recognized. In natural language processing, tokenization is often used as a preliminary step (NLP). Tokenization involves separating a text into its component words and phrases. You can then use these tokens in feature extraction or any number of other natural language processing operations.

The text can be obtained from various sources, such as documents, websites, or databases. To implement tokenization, it is sufficient to iteratively split the phrases into tokens.

The three-tier design proposed in this framework may or may not be implemented, depending on the chosen framework. To process and analyze the text, natural language processing is frequently used in the middle tier. To be sure how the Natural Language Processor is implemented in the proposed framework, more details are needed.

3.1.3 Training the CTI model

Stanford NLP is among the most well-known NLP resources, but it features a potent and tried-and-true LR Classifier. It employed a well-known approach for pattern classification called Conditional Random Fields (CRF). CRF is frequently used in NLP since they make accuracy while considering the context. To put it another way, straight chain CRF forecasts the labelling for a test while considering the Labelling for nearby samples. That is crucial in NLP since the Labelling for one example might influence the label for another. It gives the classifier the annotation trained model and uses the LR Classification method to learn it. The output system's NLP budgets help the generated model and use it to retrieve the CTI keywords.

3.2 Production System

The architecture divides the manufacturing systems into three major elements: the NLP, the quality rating, and the manufacturing component.

3.2.1 The natural language processor element

Based on the framework, this element understands the text information and extracts CTI from the document. Additionally, it includes a section for regex-parsing the CTI, which removes using unique regex rules for IPs, passwords, etc. After normalizing the data into the format, it is stored for later processing.

3.2.2 Quality ranking element

This element deduplicates the content to eliminate duplications and combines the material for one particular attack and/or operation once it has isolated and normalized the information. It is in charge of ranking the resources, which is crucial from the client's standpoint. The objective of the quality ranking system is to provide a consolidated and unified perspective on the attack or operation by identifying and combining attacks or operations that are similar. If a user is researching a specific attack, like a Distributed Denial of Service (DDoS) attack, the quality ranking system will look for, merge, and deduplicate all relevant data. Important details from the perspective of the client include the name of the attack, the start and end dates of the attack, the attack vector(s), target(s), and the amount of damage caused.

The goal of incident response strategy is to limit damage, speed up repairs, and lessen the likelihood of repeated cyberattacks. Planning for security breaches and the subsequent recovery is the primary emphasis of incident response protocols. Predicated on the signal-to-noise proportion that

it quantified by calculating the total phrases and how many of those phrases are marked as phrases, it allows users to see which references are producing the most recent data and how excellent the value of the original dataset is. It labels the information and submits it to Elasticsearch to be searched for quick retrieval.

3.2.3 Production component

The transmission of CTI data is the responsibility of the production element. It includes a publisher-subscriber system that could be manufactured into any defending or investigator process and benefit from the timely and valuable intelligence in addition to exposing an Application Programming Interface (API) that allows organizations to insert the data and use it as needed.

3.3 Machine Learning-Based CTI

The suggested system is built on machine learning approaches. It is intended to extract features of the patterns of significant risks, identify malicious activities of IoT communication, and determine the kinds from SAGS systems. Network management records all traffic going to IoT nodes, analyses it, and turns it into observations. Every observation offers valuable information about the characteristics and statistics of the connectivity, which would aid in assault detection. But since humans typically put these pieces together to make patterns, many subtle ways are overlooked.

In representing the first layer of architecture, It adds IoT modules as the initial element. This module explores knowledge about the show's activities and possible threats. It is challenging to analyze and check up on huge traffic information to separate threads or regular forms since attacker traffic is blended with usual flow of data. It mixes network information automatically to produce a new depiction that features more insightful and practical network structures.

The ML-based threat identification is expressed in Figure 2. The CTI model is used to detect deep pattern extraction and attack identification. The cloud server, edge controller, and application IoT devices are interconnected. When it comes to cyber protection, the CTI Cyber Threat Intelligence model is used to spot subtle trends and spot attacks. Which specific CTI model is used determines the specific method used for pattern extraction. A network is formed in the architecture described by connecting the cloud server, edge controller, and application IoT devices. A cloud server is a remote computer that stores and manages information and program that can be accessed over

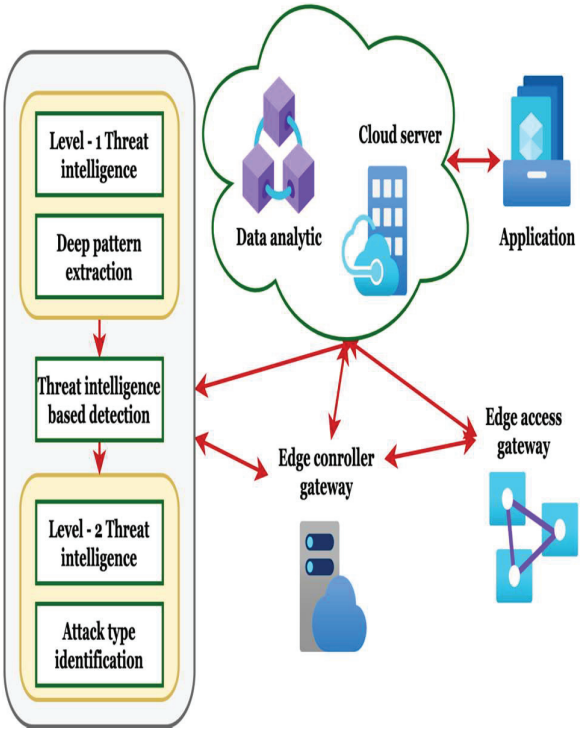

Figure 2 ML-based threat identification model.

the internet. The edge controller is the system or device at the network's periphery that coordinates and processes information sent and received between the cloud server and IoT gadgets. Application Internet of Things (IoT) devices are connected electronic gadgets that can be remotely accessed and managed. This is a very high-level description of the components, and the actual implementation and role of each component may differ depending on the network architecture in use. Fuzzy Based Algorithm Model is implemented for pattern extraction. Remaining are added The modules are created using generated machine learning that has the benefit of learning obscure and undiscovered connections without the requirement of class knowledge (i.e., attack or standard). Extracting broad patterns can seem like a wide range of data sources, which is very helpful for evaluating heterogeneous network IoT-SGAS traffic information and emerging assaults. Additionally, this paradigm resolves the privacy concerns associated with accessing and distributing this information because it relies on a place to define forms and code these in

fresh forms (if others need them). The module's information is fed into the IoT-CTIS system to ascertain whether specific patterns correspond to assaults.

As a result, this method lessens the need for passive detection and prevention methods that rely on conventional intrusion prevention methods (like signatures or regulations). It is constructed using supervised Machine Learning (ML) algorithms to find anomalous patterns that diverge from a base of typical flow previously unknown depending on experience. As a result, fewer erroneous negative patterns are mistakenly classified as standard ones.

For instance, it can specify a particular sequence of trends related to a DDoS botnet, malware, or another attack. This data is used in the 2nd layer of TI to give additional relevant information to comprehend these patterns that indicate to which assault they correspond. This engine, which is based on ML approaches, is quite capable of adapting these patterns to other known attacks.

3.3.1 Deep pattern extractor

From the original network information, a Deep Pattern Extractor (DPE) component helps to extract additional knowledge and patterns. It takes the data from the gathered observations and determines how the features are related, resulting in concise and practical pattern descriptions. That was an uncontrolled feed-forwarding neural technique with multiple subsystems (encoder and decoding) split by a code/bottleneck level. The input level and one or more concealed layers comprise the encoding sub-network that uses just the code/bottleneck level to obtain the output. In contrast, the decoding sub-network utilizes the coding level as input to recreate the input level.

Like other deep auto-encoders, the IoT-CTIS system operates by limiting learning and preventing the copying of input information while providing sparseness to every node's outputs in concealed layers. As a result, relatively few units are engaged for each view of network information. It can effectively extract relevant and generalized patterns because it streamlines the learning experience and generalizes to previously unknown material.

4 Mathematical Equation

$E(r_{\emptyset}(h_{\emptyset}(f_x), f_x))$ is the losses value/reconstruction mistake generated by DPE, where f_x denotes the (x) observations and $h_{\emptyset}(f_x)$ is the output of the encoding sub-network. This method relies on Equation (1). While Equation (2) is used to compute the decoding sub-network result, represented

by the $r_{\emptyset}(h_{\emptyset}(f_x), f_x)$.

$$h_{\emptyset}(f_x) = \propto \{B_{f_x} + w\} \tag{1}$$

$$r_{\emptyset}(h_{\emptyset}(f_x)) = \propto \left\{ \frac{B_{f_x}}{\propto} + w \right\} \tag{2}$$

f_x^T are the outputs from the encoded level using the coders, $\{B_{f_x}, w\}$ reflects the matrix of weighting and biased levels of the encoding layer, $\{\frac{B_{fx}}{\propto}, w\}$, $\emptyset$ provides the matrix of weighted and biased elements of the decoding layer and $\propto$ is the intended perception. The extraction algorithm is denoted in Algorithm 1.

Pattern extraction

#Training

For every observation in f do

Input = obtain features

Code = trained decoding network

End for

Testing

Coding_list = []

For every measurement in g compute

Input = obtain features

Pattern = coding data

Coding_list ← Pattern

End for

The extraction algorithm in a Malware Cyber Threat Intelligence System refers to the process of collecting, analysing and transforming raw data into structured information that can be used for threat analysis and mitigation. This can involve a variety of techniques such as data normalization, clustering, and categorization to extract relevant features, behaviours and patterns from malware samples, network traffic logs, and other sources of threat intelligence. The goal of the extraction algorithm is to provide actionable intelligence to security teams to detect, prevent, and respond to cyber-attacks. A mathematical method of comparing different values, $r_{\emptyset}(h_{\emptyset}(f_x), f_x)$, and

the error term is denoted E. It employs the Mean Square Error (MSE) from Equation (3). The MSE is the optimal option for a lost function since DEP attempts to forecast and rebuild the input data instead of classifying it, as we discovered from our trial-and-error tests. The quantity n here indicates all observations made throughout the learning experience.

$$E(r_{\emptyset}(h_{\emptyset}(f_x), f_x)) = \frac{1}{N}\sum_{x=0}^{N-1}(r_{\emptyset}(h_{\emptyset}(f_x)))^2 - (f_x)^2 \tag{3}$$

Since reducing reconstructive error/loss quantities $Min\{E(r_{\emptyset}(h_{\emptyset}(f_x), f_x))\}$ is the primary goal of learning. The DEP applies sparsity restrictions (f_x) on this training process to increase the reconstruction, allowing for the comprehension and extraction of the collected data and the significant connections. Equations (4) and (5) could describe how to do this by applying the activities regularizer algorithm R to every concealed layer's result. As a result, R adjusts the total of the actual values of the activation functions in the buried layers (here referred to as f_x^H to serve as a function for both encoding and decoding hidden levels) by the sparseness variable and penalizes it for the observed number β.

$$Min(E) = Min\{E(r_{\emptyset}(h_{\emptyset}(f_x), f_x)) + R\} \tag{4}$$

$$R = \beta\sum_{x=0}^{N-1} f_x^H \tag{5}$$

$Min\{E(r_{\emptyset}(h_{\emptyset}(f_x), f_x))\}$ is the primary goal of learning, and the activation function is denoted f_x^H. The number of observations is denoted β. The DDoS botnet strike pervades hundreds of connected gadgets. Its malevolent features typically have a higher transmission rate and reduced packets. So presuming that a gathered analysis is linked to that invasion, the DPE subsystem instantly uncovers the material of the internet traffic gathered, understands its behavior methods, and then standards them in a more comprehensive portrayal in an unsupervised fashion. If a network traffic assessment contains numeric data values, the DPE modules can discover its underlying features and compress them into a different form (i.e., practices).

4.1 TI driven detection

An ML technique-based component called TI uses the DPE component's intelligence to identify malicious actions in IoT networks. ML has the

potential to offer a more effective generalization capacity than traditional ML methods, which can be effective in the situation of unobserved data. The detecting engines are built on an LR. A recurrent neural network (NN) utilizes the concealed condition from the preceding keyframe (t) in the training experience, i.e., $h_{\emptyset}(f_x + g_{x-1})$, in contrary to a regular neural network. An LR cell is made up of two gateways, updates and resetting.

The first one determines what data should be kept and what evaluated data must be added. In contrast, the second determines how much-calculated data from the usually undiscovered level should be disregarded or destroyed. The result from the DPE, or the pattern(s)/code(s) that are most effective for detecting anomalous behaviors, can be retained by the LR. $T = (T_1, T_2, \ldots, T_n)$ is measured throughout the period $n = 1, 2, \ldots, N$, m denotes the mass of produced, as the DPE component outputs a succession of designs. The prior hidden state, g_{x-1}, serves as the input image for the LR, which accepts patterns T_n for each timestep t.

$$G_u = \propto \{B_{T_n}^V T_n + B_H^V T_{n-1} + W\} \tag{6}$$

The updated gate is then determined using Equation (6), wherein W is the biased and $B_{T_n}^V$, and B_H^V are the values of the updated gate layers for T_n and g_{x-1} correspondingly. The G_u to use the sigmoid transfer function to determine whether the newly computed data is pertinent and should be stored in memory. That is accomplished by converting these values obtained to a range of 0 to 1. While one is significant, 0 is not. The G_r Specified in Equation (7) is being used to regulate how much-calculated data from the preceding concealed state g_{x-1} is deleted.

$$G_r = \propto \{B_{T_n}^R T_n + B_H^R T_{n-1} + W\} \tag{7}$$

$$\hat{h}_n = Tanh\{B_{T_n} h_n + G_r \times B_H h_{n-1} + W\} \tag{8}$$

Equation (8) is used to compute the main memory contents, h depending on this value. The biasing function of the recurrent network with timer is denoted $B_{T_n}^R$, and the biasing function of the hidden layer is denoted B_H^R. The previous time function is denoted T_{n-1}, the hidden layer biasing function is denoted B_{T_n}. The hidden layer's final biased value is denoted B_H. The number of the hidden layer is denoted h_n, the previous layer hidden function is denoted h_{n-1}. It applies the Tanh functional to the total of $B_{T_n}^V\ T_n$. W is the element-wise relationship among G_r and $B_{T_n}^V\ T_{n-1}$. All values obtained are controlled and kept inside the border [−1,1] to prevent some calculated

values from bursting and making others irrelevant. Equation (9) determines the memory's ultimate contents in the present timestep (n), which is denoted H_n. To decide what information should be gathered from the current memory contents T_n and the prior step T_{n-1} and sent to the networks (i.e., period $n+1$), the G_u is employed.

$$H_n = Tanh\{G_u \times h_{n-1} + (1 - G_u)(\times)\hat{h}_n\} \tag{9}$$

Depending on the outcomes from the modules, the GRNN performs the exact arithmetic computations during the learning phase. The result from the LR levels is additionally supplied to the outcome nodes to carry out the binary classification task to decide what action ($\hat{O}$) should be taken about the series of features P. Equation (10) is used to minimize the loss of fitness values for data instances (i.e., batch size) among the total performance (O) and anticipated output ($\hat{O}$).

$$E(O, \hat{O}) = \frac{1}{N} \sum_{x=0}^{N-1} (\hat{O}_x)^2 - (O_x)^2 \tag{10}$$

The actual output is denoted O_x. Given network activity's evolving, massive, and various properties, this feature and the LR's teaching approach are beneficial for identifying the strange characteristics of network activity. For instance, the characteristics of flooding attacks collected from the DPE are supplied to the TI engine to see abnormal behaviors. The LR-based detection modules learn to keep or ignore spatial structure at every step as it deems appropriate for detecting anomalous behaviors. Surveillance, military bases, and even sports all benefit greatly from anomaly detection systems. The majority of currently available abnormal activity detectors depend on motion data collected over many frames to define abnormalities. Considering the need of executing the technique at the network edge, an auto encoder based approach is developed for abnormal activity detection. An auto encoder is trained using video frames showing typical activity, from which it extracts motion information for each spatiotemporal area. An abnormal occurrence is one that shows a significant statistical outlier compared to typical occurrences. The abnormal detection activity recognition algorithm is denoted in Algorithm 2.

#Abnormal activity recognition

For every f_x In C folds, do

Initialize $f_x = test\ set$

Training model = RF(C-1)

Prediction = Training model (f_x)

If prediction = usual, then

Display normal

Else

Display malicious

End if

End for

Ability to detect and identify abnormal or suspicious behaviour within a network or system that may indicate a potential cyber-attack is what is meant by "abnormal activity recognition" in a Malware Cyber Threat Intelligence System. Indicators like network traffic, system logs, user activity, and other relevant data sources may be examined to spot out-of-the-ordinary occurrences. The purpose of anomaly detection is to alert the security team ahead of time of any impending threats, and to aid in the prioritization of their response. Machine learning algorithms, statistical analysis, and behavioural analysis methods can all be used to monitor for and report on potentially malicious behaviour.

4.2 Attack type identification

The ML technique-based identification component mentioned in the method sums a background to the DPE. That resembles which invasion or danger a trend, in contrast to the detection component that denotes the malicious behaviors of networks, network activity is based on trends derived by the DPE but forgets their accurate kinds of unusual traffic. It is constructed using an RF with a softmax activation function on the hidden layers for differentiating different threat kinds. The feature extractor with a softmax value calculates the likelihood that a specific pattern corresponds to each threat class. The LR-based identification modules are trained to preserve the appropriate sequences.

Assume that a basic network framework comprises output nodes with a softmax functional and one RF level with n timesteps. The outputs network (activation feature) produces a encoding C-dimensional vectors O from the input signal of sequences $S = (S_1, S_2, \ldots, S_n)$ sustained over n timesteps.

Therefore, the likelihood that a single source S corresponds to a threat category (O) is determined using Equation (11).

$$Pr(\hat{O}_c = O_c|S) = \rho(S)_{O_c} = \frac{\exp(O_c)}{\prod_{x=0}^{C-1} \exp(O_x)} \tag{11}$$

The c-dimensional vector is denoted O_c. And the sequence of the input signal is denoted S. The predicted output threat is denoted $\hat{O}_c$. The categories of cross-entropy losses, or negative log, calculated over a group of many series of dimension n with Equation (12) have been utilized to evaluate the mistake of the output nodes (with a softmax activation function). The effectiveness of a machine learning classification model may be evaluated with the use of the cross entropy loss metric. For this reason, the loss (or error) is reported as a decimal between 0 and 1, with 0 representing a flawless model. Getting your model as close to 0 as feasible is the aim.

$$E(\hat{O}_c, O_c) = -\sum_{x=0}^{N-1} \sum_{c=0}^{C-1} \frac{O_c^{S_x}}{\log(Pr(\hat{O}_c = O_c|S_x))} \tag{12}$$

The C-dimensional vector is denoted O_c. The predicted threat is denoted $\hat{O}_c$. The input sequence is denoted S_x. Cross-entropy losses are a metric that assesses how off an algorithm is in predicting the probability distribution of its output nodes compared to the actual distribution. The cross-entropy loss measures how far off the mark a prediction is from the actual distribution by taking the negative log of the probability distribution. This means that the output nodes will make more mistakes as the cross-entropy loss increases. The number of input and dimensions are denoted N and C, respectively. The threat identification model is expressed in Algorithm 3.

#Threat identification model

For every f_x In C folds, do

Initialize $f_x = test\ set$

Training model = RF(C-1)

Prediction threat type = Training model (f_x)

Decision = Prediction threat type

End for

The recovered sequences are analyzed by CTI identification as flooding attacks. ML-depending identification engines can detect the threat and

Table 1

	SVM	CNN	DT	LDA	NB	IoT-CTIS
Accuracy	43	39	50	55	60	90
Precision	53	49	50	55	62	90
F-measure	45	49	63	52	60	88
Recall	42	57	69	65	62	90

differentiate between several versions, which can assist a security detail in performing an effective defensive and mitigating operation. Based on this information, security staff can immediately recognize that such Internet Protocol addresses the attack's origin and needs to be banned. The best malware cyber threat intelligence systems employ a multi-stage threat identification model to correctly identify and categorize threats. The following elements ought to be part of this model:

First, data is gathered from various sources, including system logs, network traffic, and malware samples, to establish a norm for system behaviour and spot outliers that may indicate an attack. Data analysis is the second step, and it involves looking through the information gathered to spot any oddities or suspicious patterns that might point to the presence of malware. Detection and categorization of threats through the use of cutting-edge analytics and machine learning algorithms is the third step. Fourth, reacting to incidents entails formulating a strategy to lessen the impact of the damage or danger that has already been done. Delivering complete reports on the threat and the response to stakeholders and management is the final step.

The proposed IoT-CTIS system is designed in this section with machine learning models like logical regression and random forest to detect and identify malware and cyber security threats. The software results of the IoT-CTIS system are analyzed and compared in the next section.

5 Simulation Analysis and Outcomes

The study and assessment of the suggested scheme, IoT-CTIS system, on various variables, which uses the UNSW-NB15 datasets, are presented in this section [26]. The IoT-CTIS system was created in Python on a Windows 8 computer with 8 GB of Random Access Memory (RAM) and an i5 Central Processing Unit. A random selection of 70% of every dataset's amount is used for training the DPE and 30% for evaluating every experiment. In order to answer research questions, put forward hypotheses, and assess results, one must engage in data collecting, which is the systematic gathering

and measurement of information on variables of interest. One way to get information is by observations of individuals in their natural environments at predetermined times. Researchers focus mostly on the actions of the people and communities they examine. Research methods might range from those that are regulated to those that are more organic or participant-based. To ensure that every observation is assessed at least once, 15-k cross-verifications are utilized for testing and certification. This research assessed ML algorithms to distinguish harmful activity from innocuous network data to implement the final concept on fog devices in the IoT-CTIS system in further work. As a result, the machine learning-based technique for anomaly identification is used in this assessment.

The simulation outcomes such as Precision (P), Recall (R), F measure (F), and Accuracy (A) are expressed in Equations (13) to (16).

$$P = \frac{Tr^P}{Tr^P + Fa^P} \tag{13}$$

$$R = \frac{Tr^P}{Tr^P + Fa^N} \tag{14}$$

$$F = 2P \times \frac{R}{P + R} \tag{15}$$

$$A = \frac{Tr^P + Tr^N}{Tr^P + Tr^N + Fa^P + Fa^N} \tag{16}$$

The true positive is denoted Tr^P, the true negative is denoted Tr^N, the false positive is denoted Fa^P, and false negative is denoted Fa^N. False Positive Rate (FPR) is denoted as the probability of the wrongly removing the null hypothesis. True Positive Rate (TPR) is denoted as the probability of correctly identifying the null hypothesis. The False Negative Ratio (FNR) is denoted as the probability of falsely identifying the classification. The True Negative Ratio (TNR) is defined as the probability of correctly identifying the wrong dataset as wrong.

The simulation findings of the IoT-CTIS system, such as accuracy and precision in detection and classifying the malware and cyber security threats, are analyzed and plotted in Figure 3. The precision and accuracy are computed using Equations (13) and (16). The simulation findings of the IoT-CTIS system in terms of malware detection accuracy and precision results. They are compared with existing classifiers namely SVM, CNN, DT, LDA, and

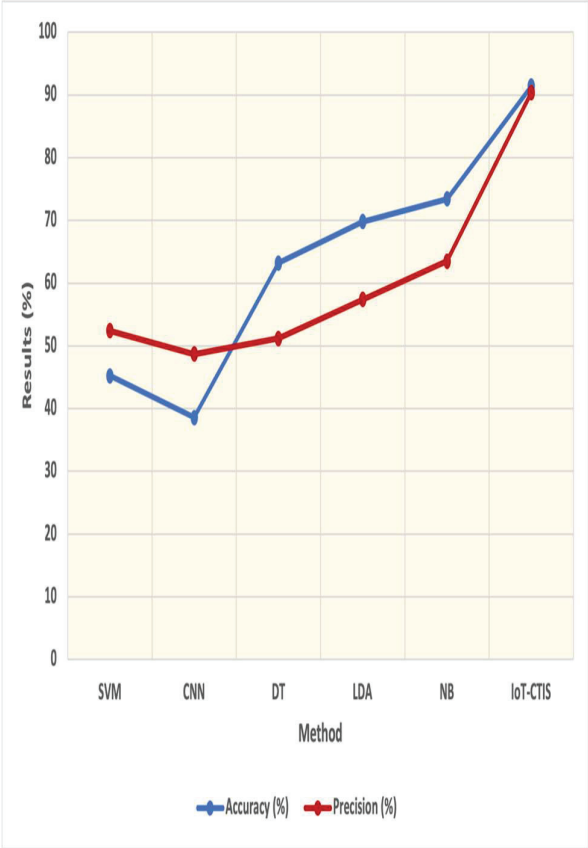

Figure 3 Simulation findings of the IoT-CTIS system.

NB. The IoT-CTIS system with a machine learning model and IoT enhances the overall system outcomes than the existing models.

The software performance analysis of the IoT-CTIS system in terms of F measure and recall are computed, and the outcomes are plotted in Figure 4. The F-measure and recall are computed using Equations (14) and (15). The proposed IoT-CTIS system outcomes are compared with existing models like SVM, CNN, DT, LDA, and NB. The SVM performs very poorly compared to other models. The proposed IoT-CTIS system outperforms all the models with the help of IoT, cyber threat intelligence, and machine learning models (LR and RF). The IoT-CTIS system efficiently analyses and detects the malware as normal and malicious nodes. The performance of the method

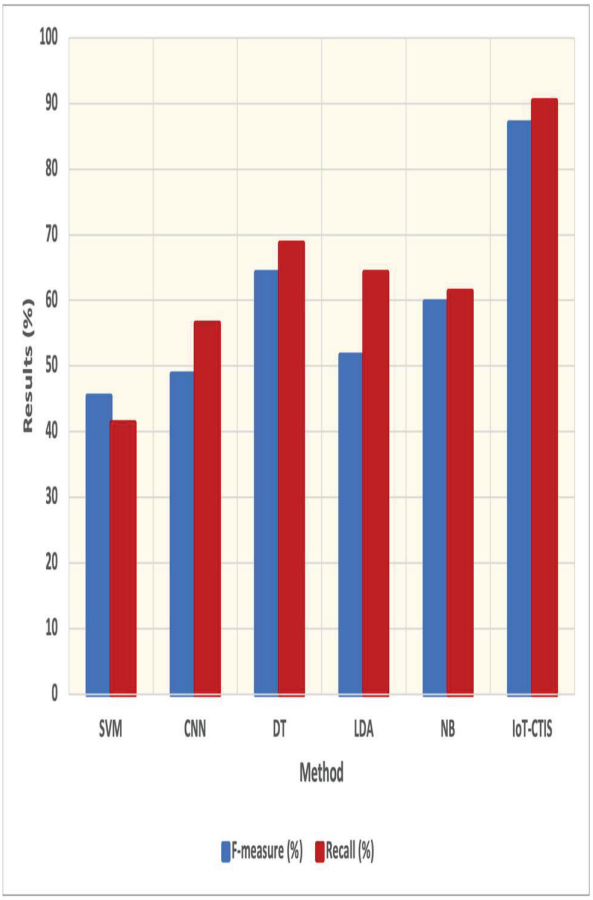

Figure 4 Software performance analysis of the IoT-CTIS system.

is based on the evaluation of the accuracy of the system in terms of identifying malicious activity and false positives. The accuracy of the system is measured by the true positive rate (TPR) and the false positive rate (FPR). The TPR measures the percentage of correctly identified malicious activity, while the FPR measures the percentage of wrongly identified non-malicious activity. The performance of the system is then evaluated by calculating the F-score, which is the harmonic mean of the TPR and the FPR, the higher the F-score, the better the performance of the system.

The error analysis of the IoT-CTIS system in terms of Root Mean Squared Error (RMSE) and Mean Squared Error (MSE) are analyzed and plotted in

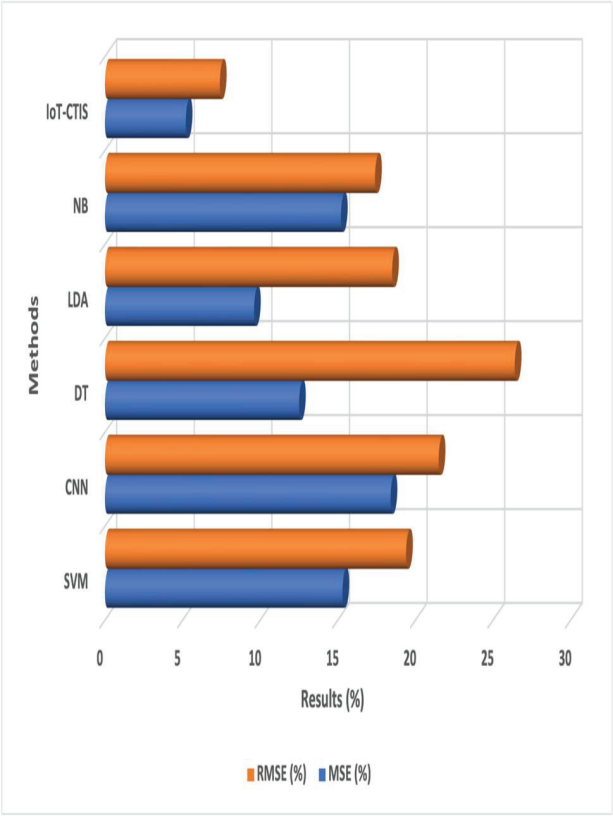

Figure 5 Error analysis of the IoT-CTIS system.

Figure 5. The MSE and RMSE are denoted in Equations (17) and (18).

$$MSE = \frac{1}{N}\sum_{x=0}^{N}\left(O_x - \hat{O}_x\right)^2 \tag{17}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{x=0}^{N}\left(O_x - \hat{O}_x\right)^2} \tag{18}$$

The actual output is denoted O_x, and the predicted output is denoted $\hat{O}_x$. The total number of samples is denoted N. The IoT-CTIS system error is computed as the function of the classification, detection of malware, and cyber security threats. The results are compared with the existing models,

and the outcomes depicted a higher performance of the suggested IoT-CTIS framework than the others. The IoT-CTIS system with machine learning and IoT enhances malware detection efficiency, reducing classification error. Analysing errors in a malware cyber threat intelligence system entails getting to the bottom of what went wrong and fixing whatever problems were uncovered. This may involve determining whether the error was caused by the user or the system, and whether it was the result of poor coding or an incorrect configuration. Understanding how the mistake affected the system or its users is also crucial. In order to prevent similar mistakes in the future, it's also crucial to pinpoint possible solutions and preventative measures.

The true positive and true negative rates are depicted in Figure 6(a), and the false positive and false negative rates are depicted in Figure 6(b). The simulation outcomes of the IoT-CTIS system are evaluated, and the results are compared with the existing models like SVM, CNN, DT, LDA,

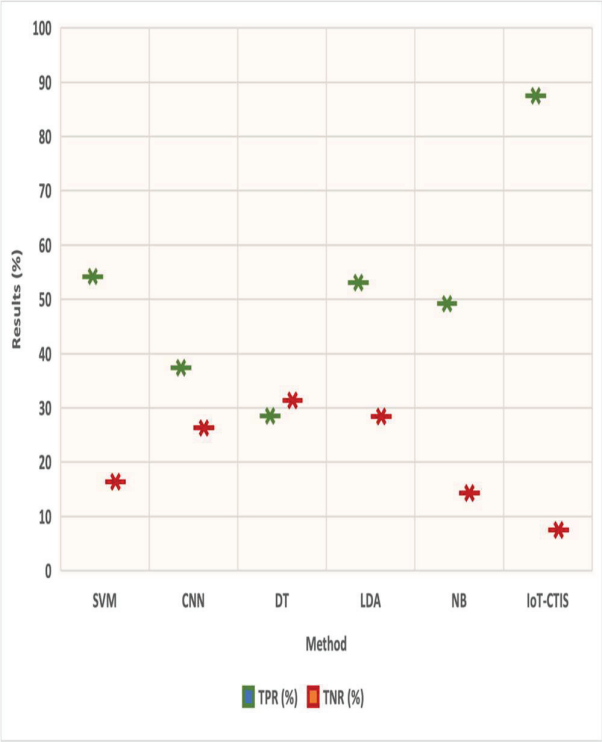

Figure 6(a) The true positive and negative rate analysis.

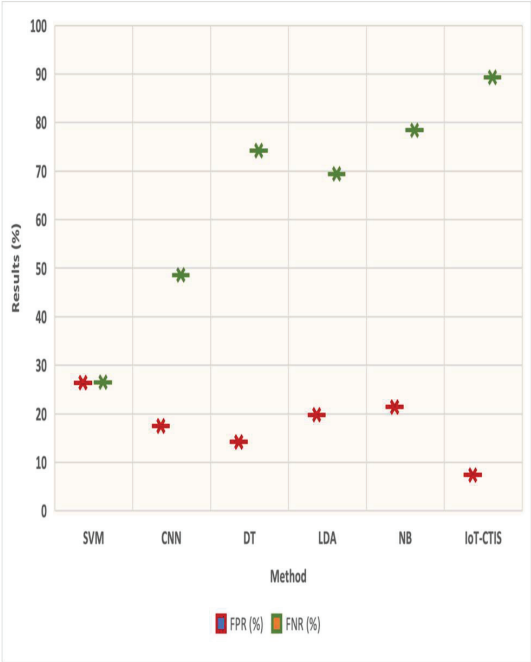

Figure 6(b) The false positive and negative rate analysis.

and NB. The TPR and TNR are expressed using Equations (19) and (20).

$$TPR = \frac{Tr^P}{Tr^P + Fa^N} \tag{19}$$

$$TNR = \frac{Fa^N}{Tr^P + Fa^N} \tag{20}$$

The true positive and false negative are denoted Tr^P and Fa^N. The FPR and FNR are expressed in Equations (21) and (22).

$$FPR = \frac{Tr^N}{Tr^N + Fa^P} \tag{21}$$

$$FNR = \frac{Fa^P}{Tr^N + Fa^P} \tag{22}$$

The false positive and true negative is denoted Fa^P and Tr^N. The IoT-CTIS system with machine learning models (LR and RF) and

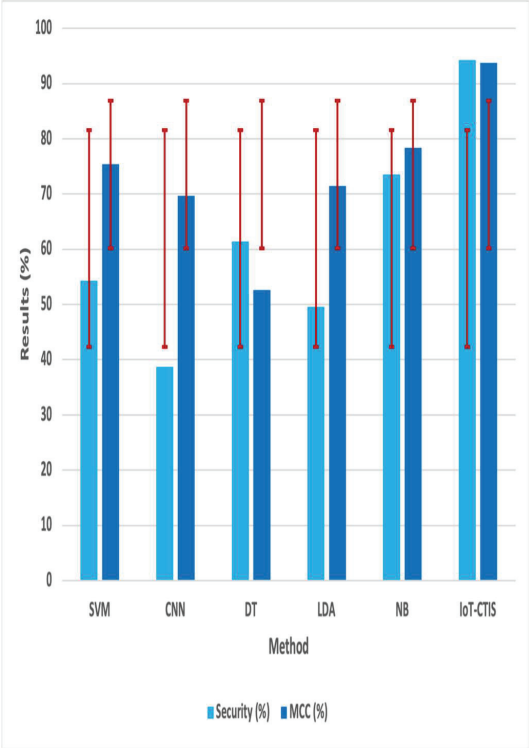

Figure 7 Security and MCC analysis of the IoT-CTIS system.

IoT enhances the overall outcomes than the existing models. The three-tier architecture increases connectivity and security.

The Security (S) and Mean Correlation Coefficient (MCC) analysis of the IoT-CTIS system are depicted in Figure 7. The security and MCC of the suggested IoT-CTIS system are higher than the existing models SVM, CNN, DT, LDA, and NB. The security and MCC are expressed in Equations (23) and (24).

$$S = \frac{Tr^P}{Tr^P + Tr^N} \tag{23}$$

$$MCC = \frac{Fa^P\, Tr^N - (Fa^P\, Fa^N)}{\sqrt{(Fa^P + Tr^P)(Tr^P + Fa^N)(Tr^N + Fa^P)(Tr^N + Fa^N)}} \tag{24}$$

The true positive and false negative are denoted Tr^P and Fa^N. The false positive and true negative is denoted Fa^P and Tr^N. The IoT-CTIS system, with the help of logical regression and random forest, enhances the malware detection and classification results and thus enhances the security. The IoT-CTIS modules with edge and cloud technologies further enhance the MCC than the existing models. An organization's ability to detect, analyse, and respond to cyber threats in a timely and effective manner is the primary objective of a malware cyber threat intelligence system. Malware cyber threat intelligence should include thorough security analysis. The external environment, internal systems, data, and processes associated with the system should all be evaluated as part of a thorough security analysis. Possible dangers to the system, as well as its vulnerabilities and risks, should be catalogued in the report, along with suggestions for eliminating or minimizing them. The analysis should also evaluate the system's architecture and design to guarantee its safety and resilience against cyber-attacks. Finally, the analysis should evaluate the system's security controls, policies, and procedures to guarantee they are sufficient and being carried out as intended.

The proposed IoT-CTIS system is analyzed, and the results are compared with existing models. The IoT-CTIS system with machine learning models and IoT enhances the overall system performance and security in the entire network. All parameters are elaborated, for this to be possible, it is necessary to have what are often referred to as the communication, control, and computing. You won't have a system where physical processes may influence calculations and vice versa without these three components.

6 Conclusion and Future Study

The extraction of beneficial cyber-threat characteristics from IoT data of Space, Air, Ground, and Sea (SAGS) systems that can assist in identifying assaults has been presented in this research. An IoT-based Cyber Threat Intelligence System (IoT-CTIS) is designed in this article to detect malware and security threads using a machine learning algorithm. The use of a machine learning algorithm allows for the development of an Internet of Things Cyber Threat Intelligence System (IoT-CTIS) that can identify malware and security threats. With its real-time monitoring and threat detection capabilities, IoT-CTIS aims to increase the safety of connected devices. Malware and other security threats in IoT-CTIS can be detected with the help of machine learning algorithms. Malware can be categorized based on its behaviour and attributes using supervised learning algorithms like decision trees and random forests.

The presence of a security threat can be inferred from anomalous behaviour using unsupervised learning algorithms like clustering and anomaly detection. Overall, IoT-CTIS can be built to counteract malware and security threats using machine learning techniques. It's worth noting, though, that the system's efficacy hinges on the quality and relevance of the data used to train the algorithm. The suggested algorithm automatically learns the obscure and unidentified patterns of IoT communication without needing the user to understand what is being searched. It depicts and codes these recently found trends in new ways that can be fed into the right engine to help recognize unusual IoT network activity based on established knowledge. The engine, founded on the LR-output layers, has a softmax activation function to give the recovered patterns meaning by recognizing their malicious types.

The suggested IoT-CTIS system can extract threat signals from heterogeneous networking IoT internet traffic utilizing the UNSW-NB15 databases. Its performance while feeding the TI detection engine with the distinguishing features as inputs demonstrates the high quality of such patterns and aids in the model's definition of abnormal activity. The TI's next level exhibits a respectable performance for detecting harmful pattern patterns. In subsequent work, the research intends to assess the effectiveness of the suggested method using an existing IoT system and look into getting TI from IoT systems, including their logs. Moreover, the research intends to improve and advance the IoT-CTIS scheme using microservices in the future. The present study is limited by the number of IoT devices and limited data resources. The outcomes are enhanced in the future using the big data analytics module. Production enhancement, configuration change, standardisation, and IT are only few of the areas where CPS in manufacturing faces obstacles. We identify five specific risk factors: prior engagement with child protective services, mental health or drug misuse issues, domestic violence, ineffective parenting, financial difficulties, and child safety/special needs. The experimental analysis of IoT-CTIS outperforms the method in terms of accuracy (90%), precision (90%), F-measure (88%), Re-call (90%), RMSE (15%), MSE (5%), TPR (89%), TNR (8%), FRP (89%), FNR (8%), Security (93%), MCC (92%).

References

[1] Kato, N., Fadlullah, Z. M., Tang, F., Mao, B., Tani, S., Okamura, A., and Liu, J. (2019). Optimizing space-air-ground integrated networks by artificial intelligence. IEEE Wireless Communications, 26(4), 140–147.

[2] Brous, P., Janssen, M., and Herder, P. (2020). The dual effects of the Internet of Things (IoT): A systematic review of the benefits and risks of IoT adoption by organizations. International Journal of Information Management, 51, 101952.
[3] Jalali, M. S., Kaiser, J. P., Siegel, M., and Madnick, S. (2019). The Internet of things promises new benefits and risks: a systematic analysis of adoption dynamics of IoT products. IEEE Security & Privacy, 17(2), 39–48.
[4] Lyer, R. (2019). The political economy of cyberspace crime and security. Academia. Edu.
[5] Aslan, Ö. A., and Samet, R. (2020). A comprehensive review of malware detection approaches. IEEE Access, 8, 6249–6271.
[6] Gupta, R., and Agarwal, S. P. (2017). A comparative study of cyber threats in emerging economies. Globus: An International Journal of Management & IT, 8(2), 24–28.
[7] Ghazi, Y., Anwar, Z., Mumtaz, R., Saleem, S., and Tahir, A. (2018, December). A supervised machine learning-based approach automatically extracts high-level threat intelligence from unstructured sources. In 2018 International Conference on Frontiers of Information Technology (FIT) (pp. 129–134). IEEE.
[8] Shackleford, D. (2017). Cyber threat intelligence, successes, and failures: The 2017 CTI survey. SANS Institute.
[9] Ghanaei, V., Iliopoulos, C. S., and Overill, R. E. (2016, July). Statistical approach towards malware classification and detection. In 2016 SAI Computing Conference (SAI) (pp. 1093–1099). IEEE.
[10] Khurana, N., Mittal, S., Piplai, A., and Joshi, A. (2019, October). Preventing poisoning attacks on AI-based threat intelligence systems. In 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP) (pp. 1–6). IEEE.
[11] Homayoun, S., Dehghantanha, A., Ahmadzadeh, M., Hashemi, S., and Khayami, R. (2017). Know abnormal, find evil: frequent pattern mining for ransomware threat hunting and intelligence. IEEE transactions on emerging topics in computing, 8(2), 341–351.
[12] Aslan, Ö., Samet, R., and Tanrıöver, Ö. Ö. (2020). Using a subtractive center behavioral model to detect malware. Security and Communication Networks, 2020.
[13] Komatwar, R., and Kokare, M. (2021). RETRACTED ARTICLE: A Survey on Malware Detection and Classification. Journal of Applied Security Research, 16(3), 390–420.

[14] Roseline, S. A., Geetha, S., Kadry, S., and Nam, Y. (2020). Intelligent vision-based malware detection and classification using a deep random forest paradigm. IEEE Access, 8, 206303–206324.

[15] Sikorski, M., and Honig, A. (2012). Practical malware analysis: the hands-on guide to dissecting malicious software. No starch press.

[16] Aslan, Ö. (2017, November). Performance comparison of static malware analysis tools versus antivirus scanners to detect malware. In International Multidisciplinary Studies Congress (IMSC).

[17] Kantarcioglu, M., and Xi, B. (2016, October). Adversarial data mining: Big data meets cyber security. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (pp. 1866–1867).

[18] Thuraisingham, B., Kantarcioglu, M., Hamlen, K., Khan, L., Finin, T., Joshi, A., ... and Bertino, E. (2016, July). A data-driven approach for the science of cyber security: Challenges and directions. In 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI) (pp. 1–10). IEEE.

[19] Harel, Y., Gal, I. B., and Elovici, Y. (2017). Cyber security and the role of intelligent systems in addressing its challenges. ACM Transactions on Intelligent Systems and Technology (TIST), 8(4), 1–12.

[20] Sabottke, C., Suciu, O., and Dumitraş, T. (2015). Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting {Real-World} Exploits. In 24th USENIX Security Symposium (USENIX Security 15) (pp. 1041–1056).

[21] Bullough, B. L., Yanchenko, A. K., Smith, C. L., and Zipkin, J. R. (2017, March). Predicting exploitation of disclosed software vulnerabilities using open-source data. In Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics (pp. 45–53).

[22] Darabian, H., Dehghantanha, A., Hashemi, S., Taheri, M., Azmoodeh, A., Homayoun, S., ... and Parizi, R. M. (2020). A multiview learning method for malware threat hunting: Windows, IoT, and android as case studies. World Wide Web, 23(2), 1241–1260.

[23] Al-Hawawreh, M., Sitnikova, E., and den Hartog, F. (2019, August). An efficient intrusion detection model for edge system in brownfield industrial Internet of Things. In Proceedings of the 3rd International Conference on Big Data and Internet of Things (pp. 83–87).

[24] Balagani, K. S., Phoha, V. V., and Kuchimanchi, G. K. (2007, April). A Divergence-measure Based Classification Method for Detecting Anomalies in Network Traffic. In 2007 IEEE International Conference on Networking, Sensing and Control (pp. 374–379). IEEE.

[25] Kruczkowski, M., and Niewiadomska-Szynkiewicz, E. (2014). Comparative study of supervised learning methods for malware analysis. Journal of Telecommunications and Information Technology.
[26] https://research.unsw.edu.au/projects/unsw-nb15-dataset
[27] Sheykhmousa, M., Mahdianpari, M., Ghanbari, H., Mohammadimanesh, F., Ghamisi, P., and Homayouni, S. (2020). Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13, 6308–6325.
[28] Lu, J., Tan, L., and Jiang, H. (2021). Review on convolutional neural network (CNN) applied to plant leaf disease classification. Agriculture, 11(8), 707.
[29] Charbuty, B., and Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2(01), 20–28.
[30] Yan, C., Chang, X., Luo, M., Zheng, Q., Zhang, X., Li, Z., and Nie, F. (2020). Self-weighted robust LDA for multiclass classification with edge classes. ACM Transactions on Intelligent Systems and Technology (TIST), 12(1), 1–19.
[31] Xu, F., Pan, Z., and Xia, R. (2020). E-commerce product review sentiment classification based on a naïve Bayes continuous learning framework. Information Processing & Management, 57(5), 102221.
[32] Saxena, R., and Gayathri, E. (2022). Cyber threat intelligence challenges: Leveraging blockchain intelligence with possible solution. *Materials Today: Proceedings*, *51*, 682–689.
[33] Moustafa, N., Adi, E., Turnbull, B., and Hu, J. (2018). A new threat intelligence scheme for safeguarding industry 4.0 systems. *IEEE Access*, *6*, 32910–32924.

Biography

Peng Xiao was born in Kunming, Yunnan, P.R. China, in 1988. He received the bachelor's degree from Yunnan University Dianchi College, P.R. China in 2012. Now, he works in Information Center of Yunnan Power Grid Co., Ltd, Kunming, Yunnan, China. His research interests is mainly information security evaluation technology, include network attack and defense technology, network security management, enterprise security system construction, etc.