SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549 / https://doi.org/10.14445/23488549/IJECE-V10I4P103

Volume 10 Issue 4, 21-30, April 2023
© 2023 Seventh Sense Research Group®

Original Article

Cloud-based Detection of Malware and Software Privacy Threats in Internet of Things using Deep Learning Approach

C. Narmadha[1], R. Muthuselvi[2], P. Somasundari[3], G. Sivagurunathan[4], Malini K V[5], Sathishkannan[6]

[1]*Department of ECE, Periyar Maniammai Institute of Science & Technology, Periyar Nagar, Thanjavur, India*
[2]*Department of CSE, Kamaraj College of Engineering and Technology, Virudhunagar, India*
[3]*Department of IT, Panimalar Engineering College, India*
[4]*Department of ECE, St. Joseph's College of Engineering, OMR, Chennai, Tamil Nadu.*
[5]*Department of EEE, Sri Sairam College of Engineering, Bangalore, India*
[6]*Department of CSE, Bannari Amman Institute of Technology, India*

[1]*Corresponding Author : narmadhaece@pmu.edu*

Received: 01 March 2023 Revised: 04 April 2023 Accepted: 16 April 2023 Published: 30 April 2023

Abstract *- The term "cyber-physical system" (CPS) refers to integrating computational and communication capabilities with physical processes. Because of patient data's regulatory and ethical implications, cybersecurity has emerged as a critical issue in the healthcare industry. Because of the sensitive nature of patient information, the layout of CPS models for any large databases requires extra precautions. Protecting user privacy and fending off attacks like spoofing, DoS, jamming, and eavesdropping are essential for cloud storage, which integrates multiple databases to deliver cutting-edge, intelligent services. This manuscript proposes a hybrid deep-learning method for scanning the entire IoT network for malware and pirated software. It is suggested that a Deep learning deep neural network be used to detect source code plagiarism in pirated software. Source code plagiarism is filtered through tokenisation and weighted feature methods, magnifying each token's significance. Next, we use a deep learning method to check for copied code. The data comes from Google's Code Jam (GCJ), and it was gathered with the intention of studying software theft. In addition, malicious infections in an IoT network can be detected by means of colour image visualisation using a deep convolutional neural network. Malware samples from the Maling dataset are used in the experiments. The experimental results show that the proposed methodology outperforms state-of-the-art methods in terms of classification performance when gauging the severity of cybersecurity threats in the Internet of Things (IoT).*

Keywords *- Cybersecurity, Malware Detection, Normalization, Software Privacy, Tokenization.*

I. Introduction

The Power Internet of Things, also known as PIoT, is the smart-grid-oriented Internet of Things that aims to achieve widespread intercommunication, adopting a comprehensive perception and effective data processing between all elements of the power system by capitalising on the advantages provided by advanced technology and data processing techniques [1]. The Industrial Internet of Things (IIoT), currently the biggest and most significant application of the Internet of Things, has brought about the greatest possibility and actively played a role in the ongoing advancements made by Industry 4.0 [2]. The Industrial Internet of Things (IIoT) incorporates Internet of Things technology, communication services, machine intelligence, cloud services, and big data analysis through every stage of the manufacturing process.

The Industrial Internet of Things (IIoT) is an open and extensible system for digital interaction. It enables the transfer of various data types between the industrial devices used in local and wider industrial activities [3, 4]. On the other hand, the enormous amount of data produced by the devices connected to the IIoT has resulted in the emergence of new requirements for the effectiveness and precision of automated, real-time data collecting, monitoring, and processing. In addition, the difficulties associated with maintaining data privacy and security will garner a significant amount of attention [5].

Patient safety is the primary factor to consider when it comes to maintaining the patient's privacy while still adhering to legal and ethical standards. Therefore, the highest possible level of attention needs to be given to the issue of data security during the process of designing CPS structures for medical domains [7,8]. Effectiveness and scalability are essential qualities for database management models. The clinical evidence is relevant to give a sufficient understanding of treatment methods, which is absolutely important to save the user's life, and the data is readily available to the legal, medical director whenever needed.

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

The data is relevant to offer specific insight into treatments that are absolutely important to save the user's life.

The cloud computing model is relevant to provide workable answers for the issues being investigated in this study. In recent research, an effort has been made to implement cutting-edge cloud services models for critical power systems (CPS), aiming to improve network dependability and initiate real-world data analysis. In addition, the term "cloud computing" refers to a type of computing infrastructure that can be utilised on demand by a user organisation. The phrase "as a service" refers to the provision of services such as processing, saving, networking, and software. The following is how reference [6] portrayed cloud computing: "The cloud" is defined as "a network of distributed and parallel computing (DC) that is constituted of interconnected as well as virtualisation, which is monitored dynamically and predicted as diversified computational power predicated on the service level arrangements (SLA) that are executed with a negotiating process among service givers and consumers."

Transferring data across various domains and providing information through the communication route are both essential [10]. In automotive networks, the exchange of data across various domains and the transmission of information across the transmission medium are both extremely important [10]. Google Inc. came up with the concept of FL as a solution to the problems of maintaining users' privacy and the burden of excessive communication. Problems arose as a result of merging data from a number of various nodes and keeping it in a centralised location [11,12]. The results of Computer Vision and Deep Convolutional algorithms are improved when a larger quantity of data is used to train the model. Nonetheless, analysing such a large quantity of data during training requires a longer amount of time. According to [13], learning with fewer data requires less time but results in a poorer accuracy score.

Privacy concerns in an IoT network can lead to the compromise of sensitive and private data through the use of both passive and active assaults (for example, a data poisoning attack). Sniffing significant portions of data that are publicly available is an example of a passive assault, whereas an active attack seeks to get access to private information, deduce its meaning, or modify it [16]. This might result in inefficient resource consumption by the devices participating in the system, as well as interruptions in the devices' typical flow of communication with one another.

Unfortunately, the majority of the IDS developed in the research use information directly from the network, resulting in poor performance with particular attack kinds in terms of high detection and the number of false alarms [15,37]. This is due to the fact that they use information directly from the network. In addition, scaling is one of the extra key challenges that come along with having IoT-enabled HS. The logic behind this is that as more devices connected to the Internet of Things are produced, there will be an exponential increase in the amount of data that must be stored [14].

This article is broken up into the following sections: The second section of this paper provides overviews of relevant work as well as the contribution of this article are mentioned. The proposed methodology is explained elaborately in Section 3. In Section 4, an experimental result and the discussions are provided. In conclusion, the paper is summed up in Section 5.

2. Related Work

FL was discussed by Khan et al. [20] in relation to IoT networks. Specifically, they highlighted some of the most recent and cutting-edge advances in FL that enable smart IoT applications. In addition to this, a taxonomy was constructed, making use of a number of different characteristics. In addition, many important issues with existing FL approaches dependent on a centralised aggregating server were found, and suggestions for fixing these issues were offered. In conclusion, a number of unresolved research challenges were highlighted, along with their root causes and potential solutions. The paper's primary flaw is the study's lack of clarity regarding the sort of review, in addition to the fact that the paper selection technique is not mentioned. However, the study addressed the unresolved problems and assessed the relevant previous work appropriately.

Numerous recent studies [18] in the area of IoT security have proposed various strategies for detecting and avoiding malware, including one that does not rely on machine learning. The authors of [19] propose a hybrid model for categorisation and an enhanced History-based IP Filtering (eHIPF) plan for creating a DDoS attack centre back for an SDN-based public cloud. In conclusion, this research was published efficiently to evaluate DDoS identification and mitigation in a Distributed architecture because it focused on solutions that did not rely on machine learning. However, the researchers' solution is not optimal for dealing with IoT Security issues in the face of the resurgence of cyber threats within the cloud computing environment.

Pham et al.[12] used fuzzy logic with the Internet of Things for the industry to comprehensively review data management, resource management, and privacy concerns (IIoT). The properties of IIoT, as well as the distributed and FL underpinnings, are first defined in this section of the study. This article will describe the rationale for integrating FL with IIoT to fulfil the goals of data privacy protection and on-device learning. The possibilities of using artificial intelligence, machine learning, and blockchain technology as Innovative methods for IIoT were highlighted. In addition to that, they investigated and summarised the many ways to work with enormous amounts of varied data. Following that, an exhaustive review of the data and risk allocation backed by IIoT application areas to FL in the industry sectors is presented next. In addition, they talked about some of the difficulties, potential solutions, and potential avenues for future research.

The problems of having a single node exposed when using centralised record keeping and having to perform expensive computations when using decentralised record keeping are considered in [22]. Then, a technique for cryptographic authentication is provided to strengthen the trustworthiness of the blockchain when it comes to the storage of medical records. In studies including such [21], the authors use data on power use as the input to their models. Wavelet transformations are utilised in the process of data collecting so that noise can be reduced. A neural net is utilised to differentiate between the typical power consumption of a chip and any difference in performance that may indicate the presence of a Trojan.

An anonymous authentication mechanism for vehicular fog services is proposed in reference [23]. This mechanism enables cross-datacenter verification, the confidentiality of automobiles, light weightness of verification communication systems, and resistance to attacks against data centers. Additionally, it achieves the lightweightness of authentication correspondence. In their paper [24], Wang et al. offer an authentication system for smart grids based on blockchain technology and use edge computing. This approach has the potential to give conditional anonymity in addition to adequate assurance of security. Self-organising maps, often known as SOMs, were utilised by Gao et al. [26] in order to identify hardware Trojans. They make use of Hotspot in order to capture the steady-state heat map produced by IC's running. The heat map is then subjected to a 2-dimensional analysis of principal components (PCA) in order to have characteristics extracted from it. The SOM is utilised to identify chips that have been infected with Trojans automatically. Both of these techniques can detect hardware effectively. Trojan.

In recent years, there has been an increase in the number of cyberattacks, which poses a threat to the protection and administration of safety before they take place. Successful cybersecurity approaches always consider the factors for protecting the privacy of persons, combine the organisation's approach with the interests of the individual, and provide several levels of security to encourage the use of safe procedures. There are many different components of the assessment and handling of cybersecurity concerns that need to be assessed [27]. These should be evaluated with attention given to the components and administration of cybersecurity practices. This layer contributes to the advancement of cloud-based security and administration in a variety of different ways [28]. The developers will be able to employ this method to their advantage and use robust security procedures and regulating tools. Developers are able to offer sufficient management capabilities for users by taking into consideration the construction and management of cloud-based models for the purpose of security management [29].

The CNN is utilised by other techniques in order to recognise fraudulent communications in the IoT automatically. In the paper [30], the authors convert the payload contained within the traffic packets into a binary form and then visualise it as a two-dimensional image. Then, in order to extract characteristics from traffic photos and classify malware, they use a compact CNN framework known as MobileNet. In [31], the authors investigate the topic of visual privacy in an Internet of Things environment. They offer a method for creating high-frame-rate films that are secured against falsification and maintain users' privacy using low-end Internet of Things cameras. A set of deep reinforcement learning algorithms and a federated learning framework have been integrated into an Internet of Things edge computing system by the researchers described in [35]. Their work is mostly geared toward enhancing the effectiveness of the mobile agent computing system as the primary focus. Deep learning methods are utilised in the research presented in reference 33 to first learn the behavioral object models from previous sensor data and then use this model to infer the FDI activity in real time. Wang et al. presented their ideas for similar research.

The major contribution of this paper is to detect malware and software privacy analysis. Data preprocessing, tokenisation and normalisation have been performed for malware detection, and the preprocessed data are given to the CNN+LSTM to predict the result.

3. Methods and Materials

As indicated in Figure 1, our proposed architecture model for addressing cybersecurity risks and implementing safeguards in commercial IoT environments can be found in this paper. For the purpose of processing malware programs and commercially pirated files, cloud storage features the implementation of four databases. Database 1 contains all of the unprocessed data on network traffic, and Database 2 has a history of malware data. Both databases can be accessed here. In addition, the latest signatures of any malware attacks that have been uncovered are saved in the third database. The software that has been illegally obtained is kept in Database 4, which is accessed by IoT devices. It serves as a repository for illegal copies, from which crackers attempt to distribute their unauthorised copies across the IoT network. Processing this enormous volume of data takes significant time and money. The preprocessing module received the raw data that was delivered from the first database. It does some preliminary processing on the original data and extracts valuable features. Following this step, the preprocessed data is sent to the detecting module. By studying the signatures included in databases 2 and 4, the detection module can identify malicious software and attacks from unlicensed software. The suggested system will alert the administrator to take appropriate action in the event that any malicious behaviour is noticed occurring within the network. Figure 1 shows the illustration of CNN+LSTM-based malware detection.

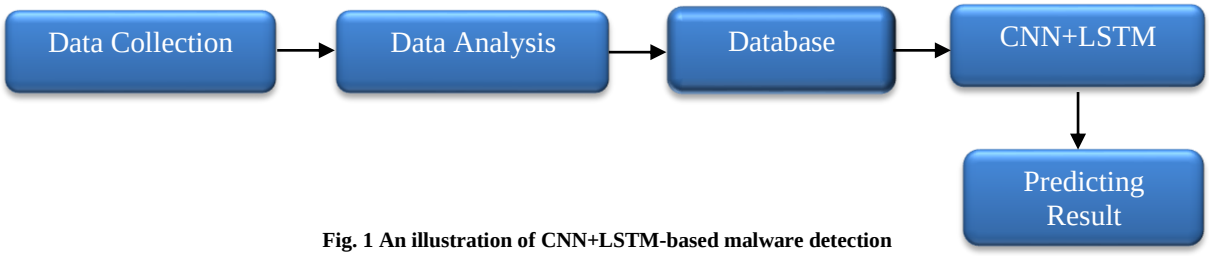

Fig. 1 An illustration of CNN+LSTM-based malware detection

3.1. Malware Threat Detection

3.1.1. Data Preprocessing

Raw binary files are used as the starting point for generating colour images, which are then used to convert the malware issue into an image classification challenge. It distinguishes the research proposal from methods currently considered to be cutting-edge, such as one in which malicious binary files are converted into a gray scale with 256 colours. This approach does not require the use of any reverse engineering tools like disassemblers or decompilers in order to function properly. In comparison to grayscale photographs, which only have 256 colours to choose from, colour images can extract superior features. In addition, the superior characteristics of malware photos might outperform others when it comes to the categorisation of malicious families. In the past, many virus detection techniques that utilised machine learning algorithms provided superior results when applied to grayscale photos. After the colour images were converted to a grayscale display, certain feature extraction algorithms were applied to categorise the various forms of malware. The classifying performance can be enhanced by applying feature reduction strategies to reduce the number of features used in the analysis. The findings demonstrated that machine learning techniques are not a superior option for identifying malware because they produce exponential values utilising colour images. When applied to large malware datasets, the performance of deep learning algorithms is superior since these kinds of systems can automatically apply filters to reduce background noise.

Thus, the utilisation of colour photos in conjunction with deep learning approaches produces superior outcomes. There are four stages involved in the process of converting a binary file containing malware to a colour image. The raw binary files are used first to construct the hexadecimal strings ranging from 0 to 15. Secondly, a hexadecimal flow is segmented into chunks of 8-bit vectors, each of which is evaluated as an unsigned integer. These 8-bit segments are then measured (0-255). Third, the 8-bit vector is then transformed into a two-dimensional matrix structure when the transformation has been completed. Fourth, each 8-bit number generated from a two-dimensional area is plotted with a combination of red, green, and blue hues with varying shades.

3.1.2. Data Normalisation

The process of scaling the data proportionally to ensure that all of the numbers fall within the acceptable range is known as data normalisation. There are primarily two benefits that come with normalising data. The first is to quicken the rate at which the model converges, and the second is to make the model more accurate.

The formula for calculating the normalisation of standard deviation is presented in formula (1), where the data set has been standardised. The data demonstrate a Gaussian distribution once the standard deviation has been normalised; this distribution has a mean = 0 and a variance = 1.

$$Normalisation = \frac{x-\mu}{\sigma} \quad (1)$$

When x is the first sample data set that was collected for the Internet of Things, M seems to be the value that represents the average, and S is the value that represents the standard deviation. If the data distribution is not generally near a Gaussian distribution, then the normalisation impact may not be very good when employing standard deviation normalisation.

3.1.3. Deep Convolutional Neural Network

Deep neural networks are exceptionally useful for a wide variety of machine learning tasks because they define parameterised functions from inputs to results as configurations of several layers of fundamental building blocks. These building blocks include affine transformations and straightforward nonlinear functions, amongst others. Examples of the latter type frequently used include sigmoids and linear activation units (ReLUs). Researchers can "train" such a parametric function with the purpose of matching any given subset of input/output instances by adjusting the parameters of these blocks and doing so in order to create a function that is parameterised.

To be more specific, we construct a gradient descent denoted by the letter L, which stands for the penalty that is associated with miss matched training data. The loss M(Θ) on the parameter, Θ is the average loss over all of the training examples (s1,..., s_N}, such that $M(\Theta) = \frac{1}{N}\sum_i M(\Theta, s_i)$. and it is expressed as a percentage. The goal of training is to identify a strategy that results in a tolerable loss and, ideally, as modest as possible.

The error function M is typically non-convex and challenging to optimise when dealing with complicated networks. The mini-batch gradient descent (SGD) algorithm is frequently utilised in practise to accomplish the task of minimisation. In this approach, one produces a batch B of random instances at each step and then calculates $h = 1/|B| \sum_{s\in B} \Delta_\theta M(\theta, s)$ as an estimate of the gradient

$\Delta_\theta M(\theta, s)$ after that, is updated in accordance with the transformed data h in the location of a minimum.

3.1.4. Convolution Layer

There are many applications for Convolutional Neural Networks (CNN) in cyber security, including vulnerability scanning, malware, and vulnerability identification. For the purpose of intrusion detection, a CNN is taught to recognise when network traffic deviates from its typical patterns. The convolutional layer of the CNN receives groups of adjacent packets as input. Then it uses filters to extract information like protocol, number of packets, and contents from those groups of packets. Following the convolution layers is a fully linked layer responsible for determining whether the input is "normal" or "invasive" based on the convolution results.

Normalisation in Batches

While the network model's parameters are iteratively updated and altered throughout training, the probability density function of the raw data at each layer likewise undergoes continuous modification, giving rise to the inner covariate shift phenomena. To prevent this from happening, the model training process should make use of a low recognition rate and better parameters. Applying saturated refers activation functions will make training the model more challenging and increase the training time. In this paper, we use the BN algorithm for the given model to boost the efficiency of the envisioned network.

3.1.5. Network Intrusion Detection using the BN Algorithm

The input is set to the minimum batch with s: $\theta = \{s_1, s_2, s_3, s_4, \ldots s_n\}$ and there are two parameters for learning such that r and θ. The predicted output will be $BNorm_{r,\theta}(s_i)$

$$Mini - Batch\ mean(\mu_\theta) = \frac{1}{n}\sum_{i=1}^{n} s_i \quad (2)$$

$$Mini - Batch\ Variance\ (\sigma_\theta^2) = \frac{1}{n}\sum_{i=1}^{n}(s_i - \mu_\theta)^2 \quad (3)$$

$$Normalize\ \widehat{s_i} = \left(s_i - \frac{\mu_\theta}{\sigma_\theta^2} + \gamma\right) \quad (4)$$

$$Scale\ and\ Shift\ (h) = r\widehat{s_i} + \theta = BNorm_{r,\theta}(s_i) \quad (5)$$

3.1.6. Weight Update

In order to adjust the mass and the bias, respectively, the BP technique and the steepest descent algorithm are both utilised during the weight update process. The utilisation of the gradient descent technique is made possible by the fact that the error function's derivatives are available for both the bias and the weight.

$$\frac{\partial}{\partial w_i^{(k)}}(Weight, bias) = \left[\frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial w_i^{(k)}} J(Weight, bias;\ s_i, h) + u w_i^k\right] \quad (6)$$

$$\frac{\partial}{\partial w_i^{(k)}}(Weight, bias) = \left[\frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial a_i^{(k)}} J(Weight, bias;\ s_i, h)\right] \quad (7)$$

where $Weight_i^{(k)}$ represents the layer's weight, $Bias_i^{(k)}$ the layer's bias for node I, and N, the normalised parameter. The feedback error is,

$$(f)_i^{(k)} = \left(\sum_{i=1}^{n}(weight)_i^{(k)}\, wi(w+1)\right).(f)_i^{(k)} \quad (8)$$

$$(f)_i^{(k)} = -(h - (b)_i^{(k)}).(f)_i^{(k)} \quad (9)$$

Where k is the final output layer, $(f)_i^{(k)}$ is the residual node, $(weight)_i^{(k)}$ is the activation value.

3.1.7. CNN-LSTM Algorithm

CNN is superior at feature extraction, but LSTM excels at processing time series, fixing the dependence problem between much time information, and improving recognition accuracy. In order to combine the most beneficial aspects of the multiple existing algorithms, the CNNGLSTM strategy is suggested in this study. Convolutional neural networks, a subset of MLP, are enjoying rapid growth in popularity. In comparison to more common feature selection algorithms, this one excels at feature learning. Because CNN's useful properties become more apparent the more the traffic data it collects, it is perfectly suited to big network systems. Feature sampling is handled by max pooling, while the convolutional layer handles feature extraction. After features have been extracted, the fully connected layer creates connections between them and the classifier, ultimately producing a classification result.

The long-term memory network (LSTM) is an improved form of recurrent neural network (RNNs) that aims to solve the explosive gradient problem. In order to eliminate transient errors and preserve stable visual features, Long short-term memory units use a different set of gate features to regulate input than traditional RNN ones.

It is possible that the CNN-LSTM method can convey both spatial and temporal information. Since there is perpetual incursion assault, the attack variable and key target are always in flux. The convolutional kernels operation, which has proven useful in image analysis, can be implemented in a convolutional neural network (CNN) to extract features at a high level. Since long short-term memory (LSTM) uses gate features and functionality to regulate this very same data storage and neglects earlier data, it is particularly suitable for processing data packet information and enhancing detection performance. This research confirms that the CNN-LSTM optimisation techniques model is useful for processing intrusion detection.

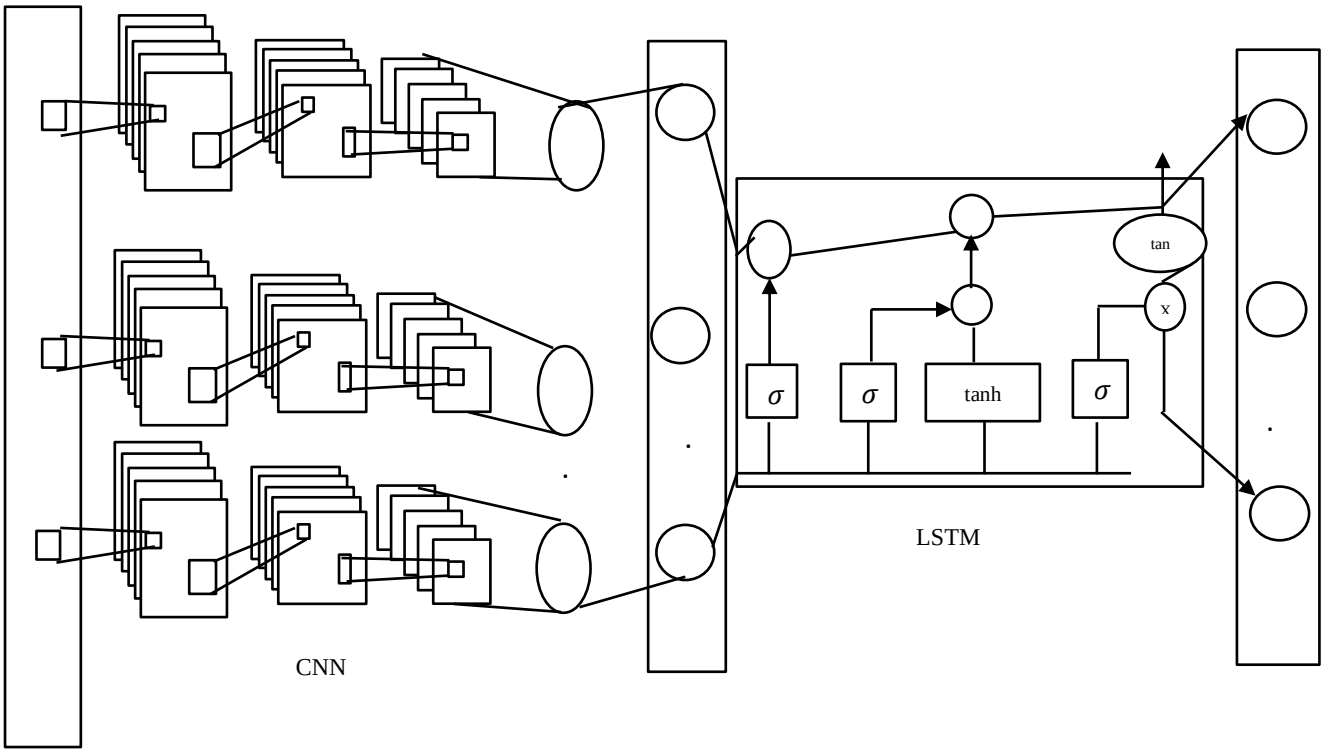

Fig. 2 Proposed architecture of CNN-LSTM

Fig.2 depicts the detailed procedures of the CNN-LSTM analytic paradigm. (1) Information from the IoV is obtained in real-time by the input layer via the stream data gathering module. This article analyses the dataset to look at factors like connection times, network protocols, and connection states. (2) The information is cleaned, in digital form, and standardised in preparation for further processing. Later, we will delve into the specific steps of the procedure. Finally (3), it sends the data file to a convolution in order to extract features, and the convolution returns the features. After each convolution layer, a pooling layer is added to reduce feature dimensions, increase convergence speed, and eliminate redundant features, all with the goal of preventing network overfitting. When the layer is fully connected, every one of the local characteristics is combined to form a global characteristic. The leaky ReLU activation function is used after every other artificial neuron in the fully connected layer has failed. Input the characteristics that CNN has indeed extracted into the LSTM. The outcome of the SoftMax function is the classification of networking data.

3.2. Software Piracy Threat Detection

The proposed method for detecting software piracy attacks is a deep learning-based technique. Various source codes capture plagiarism detection techniques. The stolen programme works the same way as the original. To reduce the dimensionality of the data, source files are segmented after traffic data are categorised as software piracy. In this stage, the TensorFlow framework is used to extract crucial features. To detect instances of plagiarism in code, the Keras deep learning API is used. Data on network activity is stored in D1, a database compiled from the Google Code Jam (GCJ) repository. The D1 files were assembled by 100 different programmers and included roughly 400 different documents detailing the code. The process of detecting software piracy threats begins with a preprocessing phase. The goal of this stage is to break the code down into manageable chunks. Next, the semi-code is deciphered into understandable language, eliminating any background noise. During tokenisation, meaningful tokens are created. Finally, a weighting mechanism predicated on the Natural log of the Term Frequency algorithm and the Term Frequency and Inverses Document Frequency is used to magnify the contribution of each token, as shown in (1).

$$W(x, D, DS) = TF(x, D)x\ IDF(x, DS) \qquad (10)$$

where x is the token definition, D is the document definition, DS is the set of all documents in the dataset, TF is the Term Frequency function definition, and IDF is the Inverse Document Frequency function definition. Tensorflow is used for deep learning and other high-level computations. Similar codes can be used to detect pirated software. In this setup, input and output data are processed by a fully interconnected network with many layers. The information is sent to the next layer, which comprises 100 neurons. There are 50 neurons in the second layer. Thirty neurons make up the third layer. The 4th dense layer is used in the output variable to determine the original source of the copied code. The dropout layer allows deep learning to circumvent the overfitting issue. Rectifier (ReLU) activation is used in the computation of the pattern:

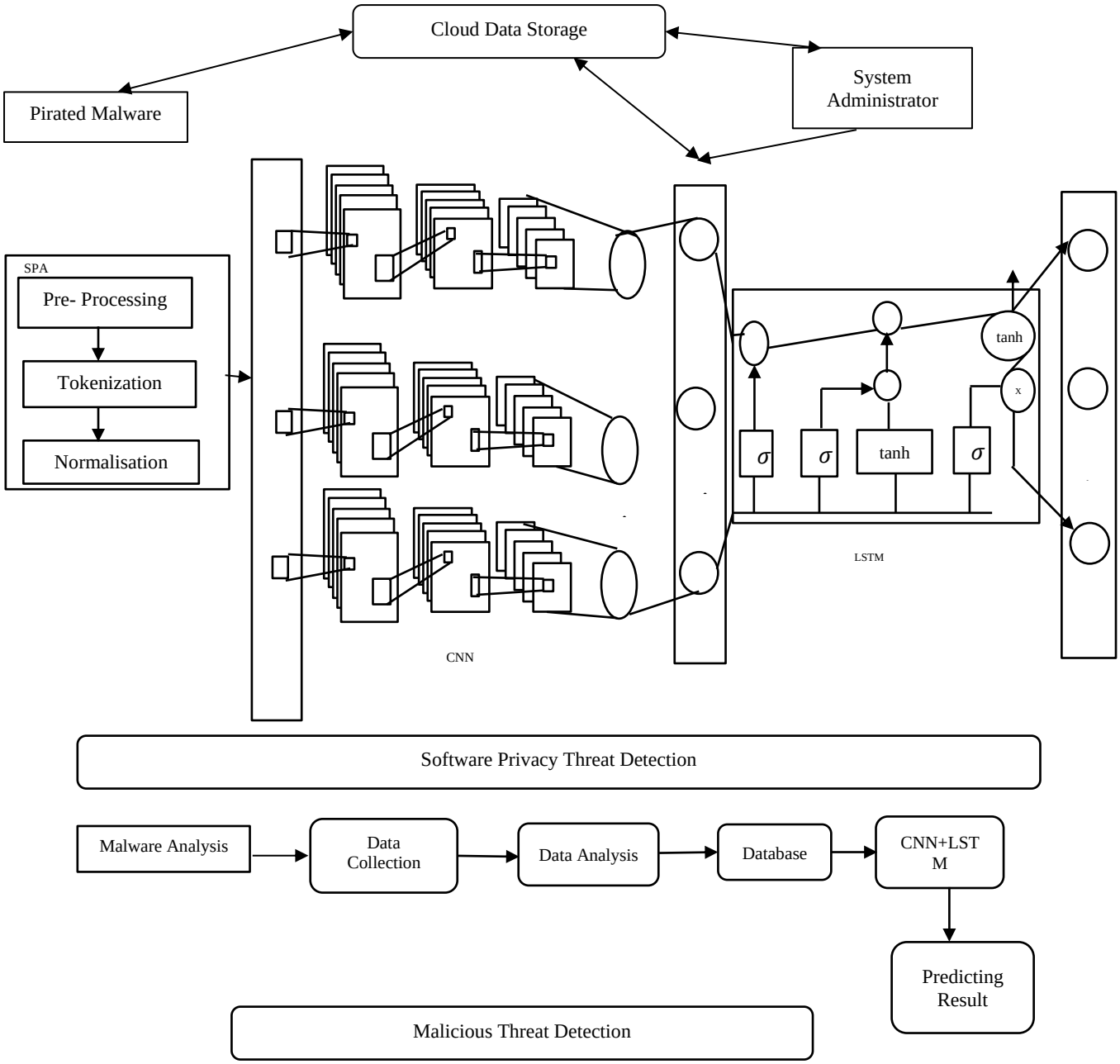

Fig. 3 Architecture of cybersecurity threat detection

$$f(s) = s^{+} = \max(0, s) \quad (11)$$

where x defines the input of the equivalent neurons. The multi-class problem is conducted using the sigmoid method defined by

$$sigmoid(s) = \frac{1}{1+e^{-x}} \quad (12)$$

The advantages of the deep learning method used to identify software piracy are as follows: (1) the algorithm is taught automatically; (2) the layout supports a wide range of computational types; (3) solutions are reliable during updates and extensions to the model; and (4) the suggested framework is scalable to large networks.Fig.3 shows the architecture for the detection of cyber security threats.

4. Experimental Results and Discussions

Two components, a preprocessor and a deep convolutional neural network, comprise a conceptual malware threat detection model. The problem is evolving into an image classification problem as colour images are generated in raw binary files. Grayscale is used, and the colour image is processed to extract features. A feature reduction technique is employed to improve classification accuracy. Its primary purpose is feature reduction. The following steps are taken to produce color information from a binary file: First, the hexadecimal strings are generated; then, they are divided into 8-bit vectors; finally, each vector is converted into a two-dimensional matrix; and finally, the matrix is plotted. After that, we use a Deep Convolutional Neural Network (DCNN) to spot the malicious code. Training images are fed into the DCNN. The Convolution

layer is used to boost signal features while decreasing noise. It helps lessen the issue of over-fitting. The computations in (4) are carried out by the convolutional layer:

Table 1. Configuration for the proposed work

Materials	Configuration
Operating System	Linux
CPU	Intel core i7 Processor
Memory	16GB
Language	Python 3.5.1
Framework	Keras 2.2.0

Pirated software's code similarity can be examined with the help of the software plagiarism measure. We used a dataset derived from Google Code Jam (GCJ) to evaluate the proposed software piracy methodology. To begin, the data set is preprocessed to extract the relevant tokens from each source alongside frequency information. Stemming, root word identification, the maximum and minimum token duration, the maximum and minimum token frequency, etc., are all part of the preprocessing phase. Token weights are retrieved using a combination of feature selection and extraction methods, such as Term Frequency and Inverse Document Frequency (TFIDF) and Logarithm Term Frequency (LogTF). Each token's impact within a document or across documents can be examined in greater detail with the help of weighting techniques. The initial dense layer has four input factors because each programmer tackled four unique programming problems. Fig.4 shows the loss and the validation loss. In 100 epochs, the validation and the loss occurred in the range between 0.0-0.01. Fig.5 shows the accuracy and the validation accuracy. In 100 epochs, 98% and 97% of validation accuracy and accuracy have been achieved.

Fig.6 shows the training and testing loss, achieved by 0.03 and 0.01, respectively. The accuracy of about 98% and 97% has been attained in training and testing, shown in Fig.7.

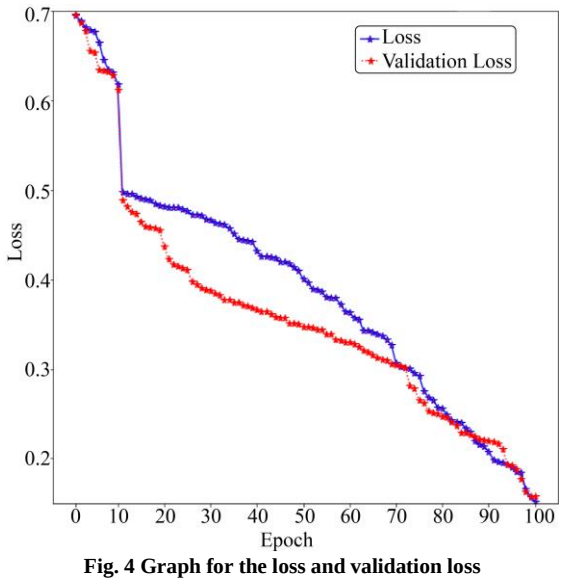

Fig. 4 Graph for the loss and validation loss

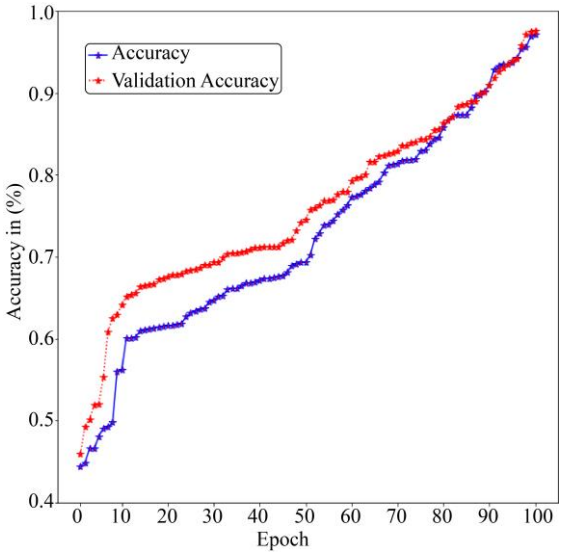

Fig. 5 Graph for the accuracy and validation accuracy

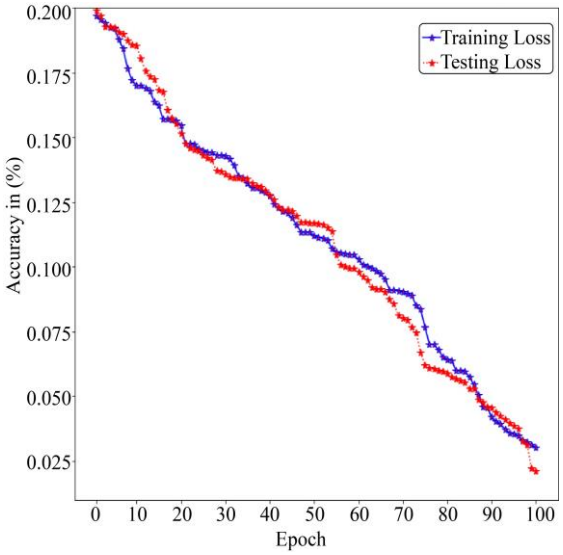

Fig. 6 Graph for the training and testing loss

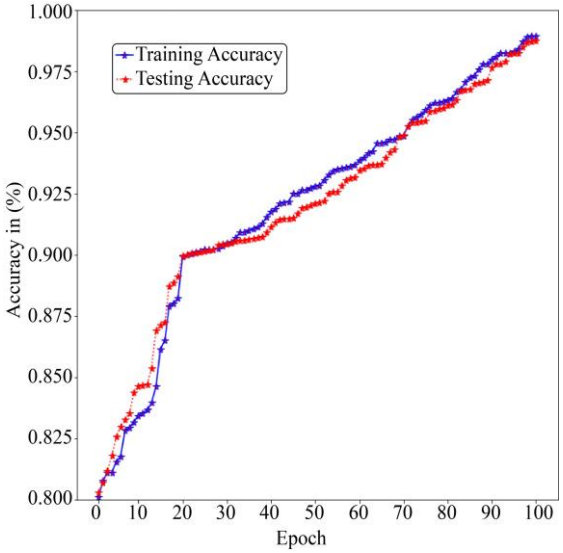

Fig. 7 Graph for the training and testing accuracy

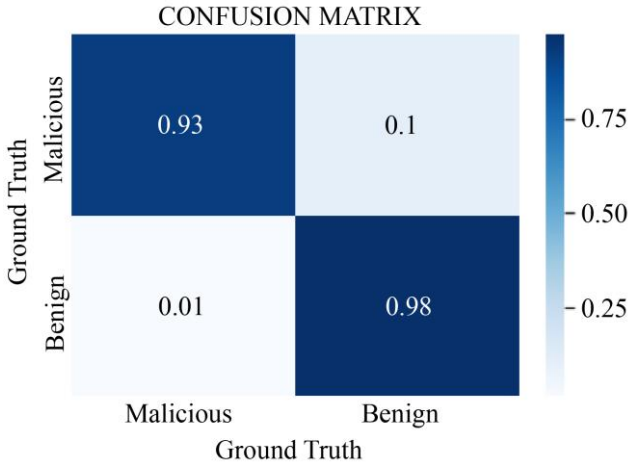

Fig. 8 shows the confusion matrix

Table 2. Performance measures for the proposed model

Metrics	Values
Precision	97.85
Recall	97.74
F-measure	98.86
Accuracy	98
Time	17s

5. Conclusion

Forecasts for the future indicate exponential growth in the IoT-based network used in the industry. The main difficulties in cyber security using IoT-based big data are the identification of software piracy and malware threats. We proposed a unified deep learning-based strategy to detect pirated and malicious files. To begin, it is suggested that software plagiarism be used in conjunction with a TensorFlow neural network to understand the advantages of the original software that has been pirated. For this study, we gathered 100 GCJ developers' source code files to understand the proposed method better. The original data is preprocessed to reduce background noise and better capture high-quality features like useful tokens. Then, we use TFIDF and LogTF weighting methods to magnify the effect of each feature on how similar the source code is. The values of the weights are then fed into the planned deep-learning technique. Second, we proposed a new approach to using IoT to detect malware, one that uses convolutional neural networks and color space visualisation. In order to better visualise the malware, we have transformed the files into color images. We then fed these graphical malware characteristics into a deep neural network using convolution. The experimental findings demonstrate that the combined approach retrieves the highest quality classification results compared to the state-of-the-art methods. Keywords are extracted via the authentication and authorisation process, but the internal structure of the source code is hidden. Using abstract syntax trees and control flow graphs can capture code syntactic and behavioural structure. We plan to investigate using these characteristics in the future to identify illegal copies. It is a major challenge to detect unknown malware. In addition, we will attempt to suggest a method that can identify malware from families we are not familiar.

References

[1] Guneet Bedi et al., "Review of Internet of Things (IoT) in Electric Power and Energy Systems," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 847–870, 2018. [CrossRef] [Google Scholar] [Publisher link]

[2] Hugh Boyes et al., "The Industrial Internet of Things (IIoT): An Analysis Framework," *Computers in Industrial*, vol. 101, pp. 1–12, 2018. [CrossRef] [Google Scholar] [Publisher link]

[3] Emiliano Sisinni et al., "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018. [CrossRef] [Google Scholar] [Publisher link]

[4] Xiaoding Wang et al., "A Secure Data Aggregation Strategy in Edge Computing and Blockchain Empowered Internet of Things," *IEEE Internet Things Journal*, vol. 9, no. 16, pp. 14237-14246, 2022. [CrossRef] [Google Scholar] [Publisher link]

[5] Jayasree Sengupta, Sushmita Ruj, and Sipra Das Bit, "A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT," *Journal of Network and Computer Applications*, vol. 149, 2020. [CrossRef] [Google Scholar] [Publisher link]

[6] Rajkumar Buyya et al., "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009. [CrossRef] [Google Scholar] [Publisher link]

[7] Daming Li et al., "A Novel CNN Based Security Guaranteed Image Watermarking Generation Scenario for Smart City Applications," *Information Sciences*, vol. 479, pp. 432–447, 2019. [CrossRef] [Google Scholar] [Publisher link]

[8] Mehedi Masud et al., "A Lightweight and Robust Secure Key Establishment Protocol for Internet of Medical Things in Covid-19 Patients Care," *IEEE Internet Things Journal*, vol. 8, no. 21, pp. 15694, 15703, 2020. [CrossRef] [Google Scholar] [Publisher link]

[9] S. Masood Ahamed, and V.N. Sharma, "Malware Detection using Optimized Random Forest Classifier within Mobile Devices," *SSRG International Journal of Computer Science and Engineering*, vol. 3, no. 5, pp. 46-52, 2016. [CrossRef] [Publisher link]

[10] Weizheng Wang et al., "Data Freshness Optimization under CAA in the UAV-Aided MECN: A Potential Game Perspective," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1-10, 2022. [CrossRef] [Google Scholar] [Publisher link]

[11] Supriya Yarradoddi, and Thippa Reddy Gadekallu, *Federated Learning Role in Big Data, Jot Services and Applications Security, Privacy and Trust in Jot a Survey*, Trust, Security and Privacy for Big Data, CRC Press, 2022. [Google Scholar] [Publisher link]

[12] M. Parimala et al., "Fusion of Federated Learning and Industrial Internet of Things: A Survey," *arXiv 2021, arXiv:2101.00798*, 2021. [CrossRef] [Google Scholar] [Publisher link]

[13] Shaashwat Agrawal et al., "Federated Learning for Intrusion Detection System: Concepts, Challenges and Future Directions," *Computer Communication*, 2021. [CrossRef] [Google Scholar] [Publisher link]

[14] Rajesh Gupta et al., "Smart Contract Privacy Protection using AI in Cyber-physical Systems: Tools, Techniques and Challenges," *IEEE Access*, vol. 8, pp. 24746–24772, 2020. [CrossRef] [Google Scholar] [Publisher link]
[15] Mahmudul Hasan et al., "Attack and Anomaly Detection in IoT Sensors in IoT Sites using Machine Learning Approaches," *Internet of Things*, vol. 7, p. 100059, 2019. [CrossRef] [Google Scholar] [Publisher link]
[16] Marwa Keshk et al., "An Integrated Framework for Privacy-preserving based Anomaly Detection for Cyber-physical Systems," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 1, pp. 66-79, 2019. [CrossRef] [Google Scholar] [Publisher link]
[17] Raveendranadh Bokka, and Tamilselvan Sadasivam, "Securing IoT Networks: RPL Attack Detection with Deep Learning GRU Networks," *International Journal of Recent Engineering Science*, vol. 10, no. 2, pp. 13-21, 2023. [CrossRef] [Publisher link]
[18] Pooja Yadav, Ankur Mittal, and Hemant Yadav, "IoT: Challenges and Issues in Indian Perspective," *3rd International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, 2018. [CrossRef] [Google Scholar] [Publisher link]
[19] Trung V. Phan, and Minho Park, "Efficient Distributed Denial-of Service Attack Defense in SDN-Based Cloud," *IEEE Access*, vol. 7, pp. 18701-18714, 2019. [CrossRef] [Google Scholar] [Publisher link]
[20] Latif U. Khan et al., "Federated Learning for Internet of Things: Recent Advances, Taxonomy, and Open Challenges," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 3, pp. 1759-1799, 2021. [CrossRef] [Google Scholar] [Publisher link]
[21] Xiaopeng Li et al., "Detection Method of Hardware Trojan based on Wavelet Noise Reduction and Neural Network," *Cloud Computing and Security*, pp. 256–265, 2018. [CrossRef] [Google Scholar] [Publisher link]
[22] Prateek Pandey, and Ratnesh Litoriya, "Securing and Authenticating Healthcare Records through Blockchain Technology," *Cryptologia*, vol. 44, no. 4, pp. 341–356, 2020. [CrossRef] [Google Scholar] [Publisher link]
[23] Yingying Yao et al., "BLA: Blockchain-Assisted Lightweight Anonymous Authentication for Distributed Vehicular Fog Services," *IEEE Internet Things Journal*, vol. 6, no. 2, pp. 3775–3784, 2019. [CrossRef] [Google Scholar] [Publisher link]
[24] Jing Wang et al., "Blockchain-based Anonymous Authentication with Key Management for Smart Grid Edge Computing Infrastructure," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1984–1992, 2020. [CrossRef] [Google Scholar] [Publisher link]
[25] R.Surendiran, and K.Alagarsamy, "A Crtitical Approach for Intruder Detection in Mobile Devices," *SSRG International Journal of Computer Science and Engineering*, vol. 1, no. 4, pp. 6-14, 2014. [CrossRef] [Publisher link]
[26] Honghao Gao et al., "Mining Consuming Behaviors with Temporal Evolution for Personalized Recommendation in Mobile Marketing Apps," *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1233–1248, 2020. [CrossRef] [Google Scholar] [Publisher link]
[27] Yanmiao Li et al., "Robust Detection for Network Intrusion of Industrial IoT Based on Multi-CNN Fusion," *Measurement*, vol. 154, p. 107450, 2020. [CrossRef] [Google Scholar] [Publisher link]
[28] Luke R. Parker et al., "Demise: Interpretable Deep Extraction and Mutual Information Selection Techniques for IoT Intrusion Detection," *14th International Conference on Availability, Reliability and Security*, pp. 1–10, 2019. [CrossRef] [Google Scholar] [Publisher link]
[29] Zhihong Tian et al., "A Distributed Deep Learning System for Web Attack Detection on Edge Devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963–1971, 2020. [CrossRef] [Google Scholar] [Publisher link]
[30] Lei Bai et al., "Automatic Device Classification from Network Traffic Streams of Internet of Things," *IEEE 43rd Conference on Local Computer Networks (LCN)*, 2018. [CrossRef] [Google Scholar] [Publisher link]
[31] Joshua Bassey et al., "Intrusion Detection for IoT Devices based on RF Fingerprinting using Deep Learning," *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, 2019. [CrossRef] [Google Scholar] [Publisher link]
[32] S. Veerapandi, R. Surendiran, and K. Alagarsamy, "Enhanced Fault Tolerant Cloud Architecture to Cloud based Computing using Both Proactive and Reactive Mechanisms," DS Journal of Digital Science and Technology, vol. 1, no. 1, pp. 32-40, 2022. [Google Scholar]
[33] Youbiao He, Gihan J. Mendis, and Jin Wei, "Real-time Detection of False Data Injection Attacks in Smart Grid: A Deep Learning-based Intelligent Mechanism," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2505–2516, 2017. [CrossRef] [Google Scholar] [Publisher link]
[34] Xiaofei Wang et al., "In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning," *IEEE Network*, vol. 33, no. 5, pp. 156–165, 2019. [CrossRef] [Google Scholar] [Publisher link]
[35] Ahmed Sedik et al., "Efficient Deep Learning Approach for Augmented Detection of Coronavirus Disease," *Neural Computing and Applications*, vol. 34, pp. 11423-1140, 2022. [CrossRef] [Google Scholar] [Publisher link]
[36] S. Veerapandi, R. Surendiran, and K. Alagarsamy, "Live Virtual Machine Pre-copy Migration Algorithm for Fault Isolation in Cloud Based Computing Systems," *DS Journal of Digital Science and Technology*, vol. 1, no. 1, pp. 23-31, 2022. [Google Scholar]
[37] Chao Liang et al., "Intrusion Detection System for Internet of Things Based on a Machine Learning Approach," *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, 2019. [CrossRef] [Google Scholar] [Publisher link]