CU Global Management Review
Vol.13 | Issue 2 | July - December 2024, pp. 155-166

Detecting Botnets in Internet of Things (IoT) Applications Using LSTM and BiLSTM Models

Qamar Alam[1], Pallavi Singhal[2], Usha Sharma[3]

[1]Department of Computer Science, Maulana Azad University, Jodhpur, Rajasthan
[2] Department of Computer Science, Maulana Azad University, Jodhpur, Rajasthan
[3] Department of Computer Science, SNS College Satna, Madhya Pradesh

Abstract

The rise of IoT has led to a notable increase in the number of devices that are connected to the internet, creating a fertile ground for botnets to infiltrate and use these devices for malevolent intent. Because modern botnets employ sophisticated methodologies and IoT devices have limited processing capacity, traditional methods of detecting botnets in IoT contexts face major complications in this study, a Bidirectional Long Short-Term Memory and Long Short-Term Memory (LSTM-BiLSTM) hybrid model proposed as a unique method for botnet detection. While the LSTM component processes temporal sequences to meet the difficulty of detecting developing botnet behaviours over time, the LSTM component pulls complex features from IoT traffic data, capturing geographic patterns and anomalies. By combining the advantages of two deep learning architectures, this model improves detection accuracy and resistance to new threats. Using a large dataset of Internet of Things network traffic, we assess the BiLSTM-LSTM model's performance and compare its efficacy to other machine learning techniques and conventional detection methods. The model's promise for real-time, scalable botnet detection in Internet of Things contexts is shown by the results, which show notable improvements in detection rates and a decrease in false positives. This study provides a viable approach to improve cybersecurity in the quickly expanding IoT landscape and emphasises the significance of cutting-edge machine learning techniques in protecting IoT infrastructures.

Keywords: Internet of Things (IoT), Botnet, Bidirectional long short-term Memory, Long Short-Term Memory (LSTM)

1. Introduction

The Internet of Things explosive expansion has produced an interconnected world where devices communicate intelligently to optimize systems across various sector, from smart homes to industrial digitization. Still, this comparability has also exposed vulnerabilities to substantial cybersecurity risks, particularly the appearance of botnets(karat,2024) Botnets, collections of

infected devices managed remotely by cybercriminals, pose a severe risk to IoT environments by allowing extensive attacks like Distributed denial of service (DDos), data theft, and system breaches. Within the dynamic cybersecurity environment, the constant war against malicious software, typically called malware. This strategy requires support to stay up to date with the quick introduction of new malware and its variants, even though it has demonstrated efficacy in detecting known dangers.

The difference between conventional and modern malware is illustrated in Table 1. Algorithms like as Long short-term memory (LSTM) and Convolutional neural Network (CNN) have illustrated their effectiveness in malware analysis by analyzing its behavioral trends. These methods are improving our ability to decrease security issue and altering the manner that malicious code is examined (Aslan,2020). The procedure that produced the data involved creating a binary vector of permissions used ({1=used, 0=no used}) for each application that was examined. The "Type" of the malware/benign samples was also used to categorize them; 1 denoted malware and 0 non-malware.

2. Background

A network of connected, communicative devices is referred to Internet of Things (IOT) and transmitting data autonomously through the internet. These devices range from simple Sensors to sophisticated systems including intelligent homes or industrial equipment. While A network of compromised devices, or "bots," under the control of a hacker is known as a botnet. The risk associated with botnets is exacerbated by the often-insufficient security protocols in various IoT devices. Botnets are a tool that hackers can employ to interrupt services, obtain sensitive data or initiate distributed denial-of-service (DDoS) attacks.(Aslan,2019)

2.1 Malware

Any software or script created with the intent to cause damage is known as malware, short for malicious software, exploit, or compromise the integrity, functionality, or security of a device, network, or information system. It encompasses a wide range of threats and has become one of the critical obstacles in cybersecurity. There are various categories of malicious software including worms, trojans, crypto ware, adware, Rootkits, Botnets, Keyloggers and Fileless Malware, advertising-supported software, root-level kits, bot networks, keystroke loggers, and non-file-based malware. In the past, malware was created with simpler objectives, making it relatively easy to detect. This type of malware is referred to as traditional malware. In contrast, contemporary malware operates at the kernel level, making it more dangerous and harder to identify compared to traditional malware. (Vinayakumar,2020)

0 This advanced malware, known as next-generation malware, has the capability to carry out more severe attacks., including advanced persistent threats and precision-targeted exploits which have never been before, and more than one type of malware is used during the attack.

2.1.1 How Malware Spreads

- **Cybernetic data supplements**: Malware is often delivered through phishing messages incorporating detrimental attachments or web links.
- **Maliciously altered Online sites :** Malware can be automatically installed on a user's device through drive-by downloads when they visit a compromised or malicious website.
- **Application Deployment:** Software downloads from unverified sources can facilitate malware infiltration, especially if bundled with legitimate-looking programs.
- **Removable Media:** USB drives, CDs, or external hard drives can carry malware and infiltrate any system with which they are linked.
- **Network Exposures:** Malware can exploit weaknesses in network security to spread across systems and devices within a network.

2.1.1.2 Impact of Malware

- **Unauthorized Data Extraction:** Malware can steal sensitive data. Including financial details, intellectual property, and personal information.
- **Monetary Degradation:** Ransomware and fraud schemes Can cause substantial monetary damage to individuals and businesses.
- **System Damage:** Malware can corrupt or delete files, making systems inoperable and causing data loss.
- **Botnet Supervision:** Compromised systems can be used in botnets to conduct out cyberattacks or illicit actions, compromising performance and security.

2.2 Virus

Malicious software that enters systems by splicing itself into executable files or applications is known as a virus. Once activated, it can replicate itself, spread to other devices, and engage in harmful activities such as data corruption, data breach, or resource takeover. Viruses rely on human activity to propagate and cause damage to systems or networks.(Al-Janabi,2020) The analysis of traditional versus next-generation malware is presented in Table 1

Table 1. Conventional versus Advanced Malware

Identification Specification	Conventional	Next generation
Execution phase	Clearly implemented	Statically programmed
Environmental condition	Immobilized	Vivacious
Expansion	Each iteration is the same	Every version is unique
Endurance within the system	Physical	Constant
By means of transmission	It uses the .exe extension	Use different extensions as well
Make use of hiding strategies	not one	Sure
Kind of Strike	Overall	aimed at
Protective hurdle	Effortless	challenging
Intended Systems	General computers	Numerous devices
Coordinate among processes	A handful of operations	Numerous Processe

According to study findings and business evaluations, the volume, complexity and economic impact of malware on the global economy have been consistently rising in recent year.Every day, one million malicious files are produced, and by 2021, it is estimated that cybercrime will have caused $6 trillion in yearly economic harm worldwide.

2.2.1 Role of Viruses in IoT Botnets

- Virus as an Infective Entity: In the realm of IoT devices, a software infection can be developed to manipulate weaknesses in connected networks (e.g. Weak security protocols, outdated software). Once the virus infects the device, it can either self-replicate across the network or deploy supplementary malware modules to integrate the device into a botnet.

- Botnet Synthesis via Computer Worms: Following infection, the virus facilitates remote exploitation by attackers to take control of the compromised IoT devices, creating a botnet. The infected devices, now bots, are controlled by a primary command-and-control (C2) server manages the bots, directing them to execute coordinated intrusions, namely Distributed denial of service (DDos) attacks, or to perform harmful actions like stealing data or sending out spam. (M. Shurman,2019)

2.3 Techniques

In this section, you will receive a comprehensive analysis of the deep learning methodologies employed for malware detection and classification.

2.3.1 Convolutional Neural Networks (CNNs)

Convolutional neural networks are a form of artificial neural network that are commonly employed to analyze visual data, including both images and video sequences. They are specifically designed to understand the spatial relationships and patterns within the data. The main idea behind CNNs is inspired by how the human visual system works. When we analyze an image, our brain identifies distinct structures and entities by discerning patterns and features through perceptual processing and pattern recognition. Similarly, CNNs learn to identify patterns and features within an image by using a technique called convolution. Convolution is like a sliding window that scans the image and detects distinct attributes. These features can be edges, corners, or more complex structures.

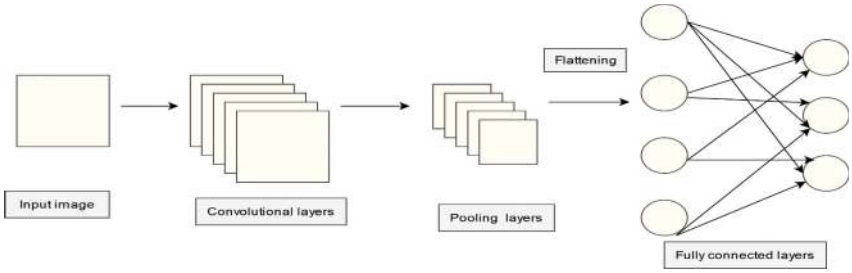


Figure 1: Foundational Convolutional Network Schema

2.3.2 Long Short-Term Memory (LSTM)

A Long Short-Term Memory (LSTM) network, which is a specified kind of Recurrent Neural Network (RNN), is engineered to process sequential data, including text, audio and time series It is adept at capturing and modelling long-term dependences with in such data. Unlike traditional RNNs, which struggle with the "vanishing gradient problem," LSTMs can remember information over long sequences, making them powerful for tasks where the order of inputs is crucial. At its fundamental level, an LSTM unit consists of three primary elements: input gate, output gate, and forget gate. By controlling information flow within the LSTM cell, these parts allow the cell to gradually increase or decrease data over time. (Wang,2021)

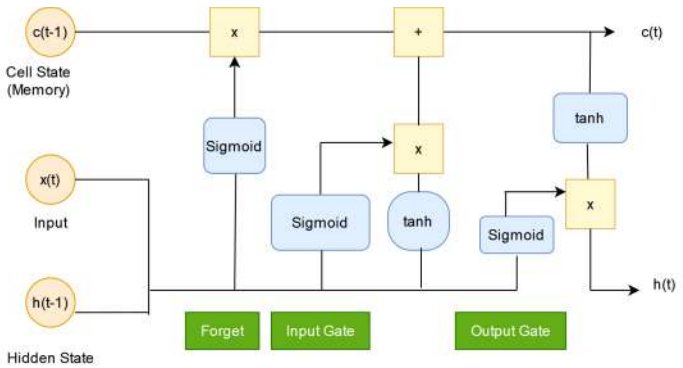


Figure 2: Standard LSTM Cell Configuration

2.3.3 BiLSTM (Bidirectional Long Short-Term Memory)

A Bidirectional Long Short-Term Memory (BiLSTM) network is a specialized recurrent neural network designed to overcome the limitations of traditional RNNs. (Almusawi,2022) It effectively addresses the vanishing-gradient problem and captures long-term dependencies in sequential data by processing information in both forward and backward directions simultaneously, considering context from both past and present timesteps.

Key Features:

1. LSTM Basics:

LSTM units are designed to capture long-term dependencies in sequential data by addressing the vanishing gradient problem found in traditional RNN

They consist of gates (input, forget, and output gates) to control the flow of information

2. Bidirectionality:

BiLSTM processes the input sequence in both directions, allowing it to understand context from both the past and the future.

The final output is a concatenation of the forward and backward LSTM outputs, enabling richer feature representation.(Aya Messai,2023)

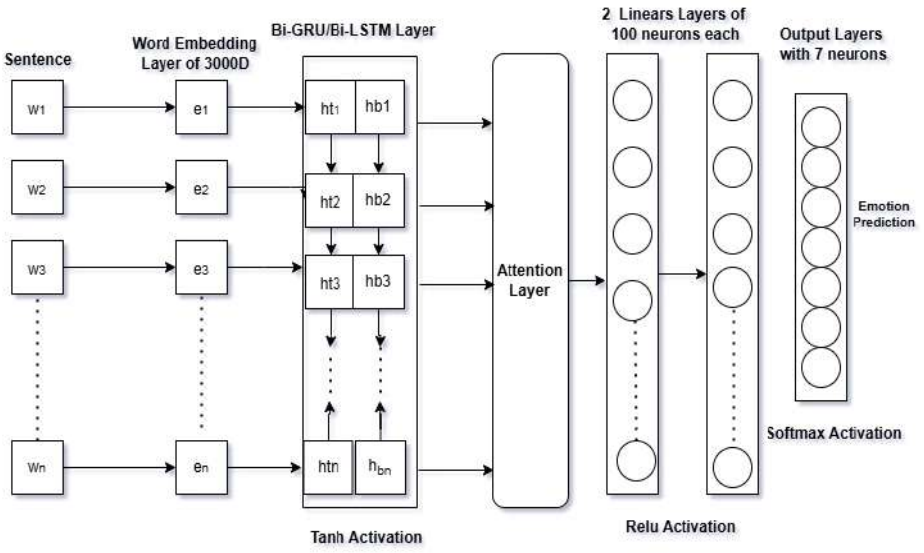


Figure 3: Single Task architecture

3. Data Repository

To evaluate the effectiveness of conventional machine learning model and deep learning models. It's crucial to create a thorough dataset that includes a diverse array of sample types. However, the availability of open-source datasets for network security and malware detection is limited due to privacy concerns of individuals and organizations. As malware has evolved and proliferated, consolidating a comprehensive source for all malware families has become increasingly challenging. Despite efforts by researchers to collaborate and share their discoveries, the task remains complex, a comprehensive dataset to obtain all the necessary samples is still lacking. In this research, the available dataset android The Android malware/benign permissions dataset used in this work was gathered by Urcuqui, & Navarro, (2016, April) and was obtained from primary sources. supervised learning techniques to detect and categorize Android malware.

This dataset is the outcome of my machine learning and Android security research project. The process that produced the binary vector of permissions used for each program that was looked at ({1=granted, 0=not granted}) was used to collect the data. Additionally, The malware being samples were classified by "Type":0 for non-malware samples and 1 for malware samples.

The dataset provided here is sourced from Kaggle and comprises approximately 331 attributes representing various Android permissions requested by applications (0 indicates not required, while 1 indicates required). The number of observations/malware instances for each permission is 398.

Table. 2 Android Cyber Threat Families

android	android.app.cts.permission.TEST_GRANTED	android.Intent.category.MASTER_CLEAR.permission.C2D_MESSAGE;	android.os.cts.permission.TEST_GRANTED;	android.permission.ACCESS_ALL_DOWNLOADS;	android.permission.ACCESS_ALL_EXTERNAL_STORAGE	android.permission.ACCESS_BLUETOOTH_SHARE	android.permission.ACCESS_CACHE_FILESYSTEM;	android.permission.ACCESS_CHECKIN_PROPERTIES	android.permission.ACCESS_COARSE_LOCATION
0 total values	398 total values	398 total values	398 total values	398 total values	398 total values	398 total values	398 total values	398 total values	398 total values
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0

3.1 Flow of Control

The extracts spatial features from the input data, which are then flattened and passed to the LSTM to capture temporal dependencies. The combined model generates the final output, with backpropagation used for optimization

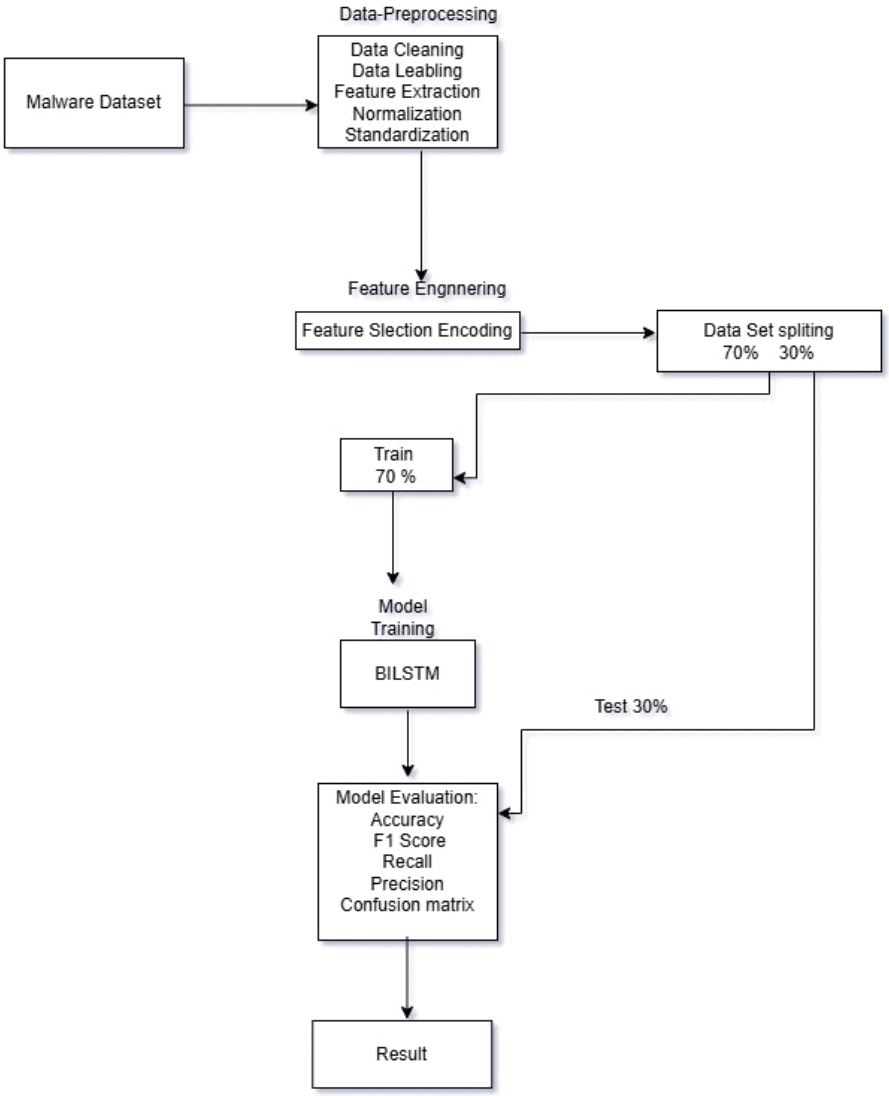


Figure 4: BiLSTM Model Architecture

4. Methodology

To optimize the dimensionality reduction of network traffic features without impacting classification effectiveness, a hybrid DL technique using Bidirectional LSTM (BiLSTM) architectures is developed.

A recurrent neural network architecture known as Bidirectional LSTM (BiLSTM) networks is specifically designed to manage long-term dependencies.(Hossain,2021) Bidirectional LSTM (BLSTM) can manage long "gaps" between the occurrence of a feature and when the model needs it. This is typically not possible with basic vanilla RNNs due to the vanishing gradient problem. The main distinction among a Bidirectional LSTM (BLSTM) and a regular basic RNN lies in the fact that Bidirectional LSTM (BLSTM) introduce an extra pathway for information flow. Put differently, aside from the covered state, there is a cell state capable of that can be employed to efficiently preserve data from previous stages. The purpose of the cell state is to act as a gradient pathway during backpropagation. This allows the gradient to travel much further back with a lower risk of vanishing or exploding during the process. On a related note, the Bidirectional LSTM (BLSTM) model is one of the most influential and widely applied learning strategies to date. Bidirectional LSTM (BLSTM) are employed in various application, including google allo, Google Translator, Apple's Siri, and Amazon Alexa.

BiLSTM (Bidirectional Long Short-Term Memory) The models are extensions of LSTM that analyse a data series using two LSTM networks, operating simultaneously in both forward and backword direction. The front layer examines the input similarly to a usual LSTM, conversely, the backward layer examines the data in the opposite order.(Aljanabi,2022)

Input Layer: In a neural network, the input layer acts as the key layer and point of entry.

Dropout Layer: During training, this layer introduces noise into the network by arbitrarily cutting the interactions within neurons in single phase and the upcoming. by performing this, minimizing overfitting result in enhanced model generalization . This often improves model fidelity throughout the evaluation process.

Bidirectional Layer: A interface layer that makes bidirectional model implementation possible for RNN layers. The bidirectional wrapper layer combines the forward and backward RNN layers into a single layer instead of building both segregated RNN layers also concatenating their outputs. (Khaitan,2016)

Dense Layer: Consists of a single vanilla neural network layer that is fully connected.

Embedding Layer: handling the conversion process of floating -point vectors from positive integer sources.

Conv1D Layer: Represents a one-dimensional convolutional layer in a neural network.

MaxPooling1D Layer: Carries out max pooling along single dimension.

The methodology for building this design is outlined in the following section. It is easy to transform this pseudo code into a real implementation using domain-specific deep learning libraries like Keras. However, fine-tuning the model parameters is more difficult and time intensive.(Wu Y,2016)Three gates are added to each network cell to improve the concept of memory in order to address the drawbacks of conventional RNNs.

Every period, when the cell reads inputs, a memory is maintained and updated.

Bidirectional LSTM (BLSTM) are composed of four components: the forget gate(f),the input gate(i),the memory cell (c),and the output gate (o).

Given the previous memory Ct−1, the updated cell memory C_t is calculated as:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Forget Gate: determines what data should be removed from the current memory

$$f_t = \sigma\left(W_f x_t + U_f \hbar_{t-1} + b_f\right)$$

Memory Gate: creates new potential memory states.

$$\tilde{C}_t = \tanh\left(W_c x_t + U_c \hbar_{t-1} + b_c\right)$$

Input Gate: The amount of information from the candidate memory that will be injected into the updated memory is determined by this gate.

$$i_t = \sigma\left(W_i x_t + U_i \hbar_{t-1} + b_i\right)$$

Output Gate: Evaluates the amount of cell memory that is kept.(Rongrong jiang,2024)

$$o_t = \sigma\left(W_o x_t + U_o \hbar_{t-1} + b_o\right)$$

Conclusion

This study Offred a summary of machine learning and deep learning methods of malware detection, along with the essential principles underlying these detection techniques. Several key studies were examined and organized based on their analysis methods: static, dynamic, or hybrid. To maintain classification accuracy while reducing the feature dimensionality of network traffic data, a BiLSTM model is implemented. bidirectional long short-term memory (BiLSTM) architectures are used in this hybrid deep learning technique. This dataset is the outcome of my machine learning and Android security research project. Additionally, the sample sets were categorized by "Type": with 0 representing non-malware and 1 representing malware.

However, it is crucial to recognize the constraints and contemplate future directions. Constraints include the necessity for the proposed methodology to utilize extensive datasets for model training. Model accuracy varies depending upon selection of layers hence required experience. Future study may involve applying the BiLSTM model to the experimental setting, with an emphasis on performance optimization and investigating the possibility of real-time threat detection.

References

Karat, G., Nair, N., et al. (2024). CNN-LSTM hybrid model for improved malware analysis and detection. ICIDCA. Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050924005982

Aslan, A., Samet, R., et.al. (2020) provide an extensive review of various malware detection methods in IEEE Access, volume 8, pages 6249-6271. https://doi.org/10.1109/ACCESS.2019.2963724

Vinayakumar, R., Alazab, M., Sonam, K. P., Venkatraman, S., et al. (2019). Effective malware detection through deep learning techniques' Access,7,46717–46738.https://doi.org/10.1109/ACCESS.2019.2906934

Al-Janabi, M., Altamimi, A.M., et al. conducted a comparative study on various machine learning techniques for malware classification and detection, presented at the 2020 21st International Arab Conference on Information Technology (ACIT).

Al-Rushdan H., Shurman M., Alnabelsi S.H., Althebyan Q., and others presented research on detecting and preventing zero-day attacks in software-defined networks at the 2019 International Arab Conference on Information Technology. The findings were published in the conference proceedings, spanning pages 278 to 282. doi: 10.1109/ACIT47987/2019.8991124.

Wang, R., Tian, C., & Yan, L. et.al. (2021). Malware detection using CNN via word embedding in cloud computing infrastructure. Security and Communication Networks, 2021. https://doi.org/10.1155/2021/8381550

Almusawi, H., & Alajeeli, A. et, al (2022). Hybrid malware detection and classification in real-time by deep learning techniques. ResearchGate. https://www.researchgate.net/publication/365196411

Hossain, H., Paul, A., Kayum, S., & others. (2021, December). Malware detection using neural networks. ResearchGate. https://www.researchgate.net/publication/3529271643.

Aljanabi, M., Altamimi, A. M., et al. (2022). A comparative analysis of machine learning techniques for classification and detection of malware. ResearchGate.https://www.researchgate.net/publication/360485697

Khaitan, P., et al. (2016). Chat smarter with Allo. Google AI Blog.https://ai.googleblog.com/2016/05/chat-smarter-with-allo.html

Wu, Y., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv. https://arxiv.org/abs/1609.08144

Steven Levy The iBrain is here—and it's already inside phone Wired.https://www.wired.com/2016/08/an-exclusive-look-at-how-ai-and-machine-learning-work-at-apple/,2016.

Gupta, A, et, al. (2018). How Alexa is learning to converse more naturally. Alexa Blogs. https://developer.amazon.com/blogs/alexa/post/15bf7d2a-5e5c-4d43-90ae-c2596c9cc3a6/how-alexa-is-learning-2018

Tavakoli, N., et al. (2019). Modeling genome data using bidirectional LSTM. In 2019 IEEE 43rd Annual Computer Software and Applications Conference (Vol. 2, pp. 183–188). IEEE.

Retrieved from: URL https://www.kaggle.com/code/srkarthik/android-malware-detection/input

Retrieved from: URL https://www.sciencedirect.com/topics/computer-science/bidirectional-long-short-term-memory-network Retrieved from: URL https://www.geeksforgeeks.org/bidirectional-lstm-in-nlp/

Aya Messai, Ahlem Drif., et al. (2023). Multi-input BiLSTM deep learning model for social bot detection. In 2023 IEEE international Conference on Advances in Electronics, Control and Communication

Jagdish, Sudhakar k et al (2024). Anomaly -Based Intrusion Detection System Using Bidirectional Long Short-Term Memory for Internet of Things in 2024 third international Conference on distributed Computing and Electrical Circuits and Electronics. https:// 10.1109/ICDCECE60827.2024.10549281

Rongrong jiang, Lilli Shi et al. (2024) Intelligent botnet detection in IoT networks using parallel CNN-LSTM fusion https:// https://doi.org/10.1002/cpe.8258

Feng Wei ,Uyen Trang Nguyen et al(2024) Twitter Bot Detection Using Bidirectional Long Short-Term Memory Neural Networks and word Embeddings in 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA).https:// 10.1109/TPS-ISA48467.2019.00021.

Zineb Ellaky, Saad Qaqa et al(2024) A Hybrid Deep Learning Architecture for Social Media Bots Detection Based on BiGRU-LSTM and GloVe Word Embedding IEEE Access(Volume:12) https:// 10.1109/ACCESS.2024.3430859

Imtiaz Ullah et al (2022) Design and Development of RNN Anomaly Detection Model for IoT Networks, https:// 10.1109/ACCESS.2022.3176317

Himanshu Nandanwar et al(2023) TL-BILSTM IoT: transfer learning model for prediction of intrusion detection system in IoT environment in 2023 International Journal of Information Security. https:// https://link.springer.com/article/10.1007/s10207-023-00787-8

Abdullah Alqahtani et al(2024) Optimized deep autoencoder and BiLSTM for intrusion detection in IoTs-Fog computing in 2024 Multimedia Tools and Applications. https://link.springer.com/article/10.1007/s11042-024-18919-0

Zineb Ellaky et al (2024) A Hybrid Deep Learning Architecture for Social Media Bots Detection Based on BiGRU-LSTM and GloVe Word Embedding in 2024 IEEE Access(Volume:12). https:// 10.1109/ACCESS.2024.3430859