



Anomaly Detection in Smart Agriculture Systems on Network Edge Using Deep Learning Technique

Bandar A Alanazi¹ , and Ibrahim Alrashdi^{2,*} 

¹Department of Computer Science, College of Computer and Information Sciences, Jouf University, Sakaka 2014, Saudi Arabia; 431100006@ju.edu.sa.

²Department of Computer Science, College of Computer and Information Sciences, Jouf University, Sakaka 2014, Saudi Arabia; irrashdi@ju.edu.sa.

* Correspondence: irrashdi@ju.edu.sa.

Abstract: With the widespread adoption of Internet of Things (IoT) technologies across various domains, including smart agriculture, urban environments, and homes, the threat of zero-day attacks has surged. This research delves into the application of deep learning techniques to detect anomalies in smart agricultural systems at the network edge, with a specific focus on safeguarding them against Distributed Denial of Service (DDoS) attacks. In this study, we propose an anomaly detection model based on CNN-LSTM to analyze sensor data collected from IoT devices. We rigorously train and test our model using two distinct datasets of sensor readings, simulating potential DDoS attack scenarios. The model's performance is assessed using key metrics such as detection accuracy, recall, and F1-score. Our results demonstrate the effectiveness of our approach, achieving an impressive anomaly detection accuracy of 99.7%. This research contributes significantly to the development of robust and efficient attack and anomaly detection techniques for smart agriculture systems at the network edge, ultimately enhancing the reliability and sustainability of agricultural practices.

Keywords: Anomaly detection, Smart agriculture, Network edge, Deep learning, Internet of Things (IoT), Zero-day attacks, Distributed Denial of Service (DDoS), Sensor data analysis, CNN-LSTM.

Phase	Date
Received	27-01-2023
Revised	22-06-2023
Accepted	28-06-2023
Published	30-06-2023

Introduction

Agriculture is one of the most important elements of life, as it is one of the main sources through which a person obtains the nutrients that he feeds on, and it also has an impact on the economy in some countries that seek to provide the necessary food and achieve self-sufficiency in crops. The traditional farming system suffers from many factors. The most important of which is climate change, which greatly affects agriculture, such as high or low temperature and humidity levels, determining the proportions of fertilizers, pesticides, and other influence [1]. This led to the need to convert traditional farms into smart farms, supported by the Internet of Things (IoT) that helps farmers overcome the obstacles of climate change and wasting water and improve the quality of crops through sensors that measure the temperature and soil in the fields in smart irrigation systems [2]. IoT plays an important role in crop quality through environmental monitoring and data analysis. Since smart agriculture integrates elements from the traditional Internet, IoT devices, cellular networks, and wireless networks, it may all incorporate security issues

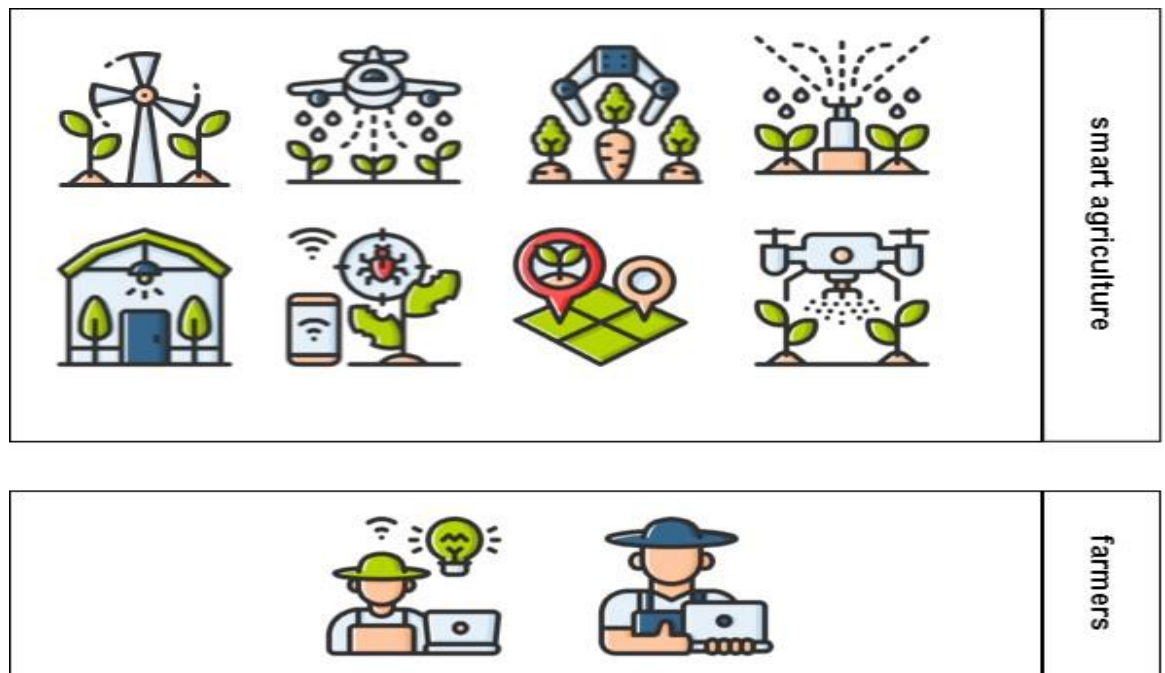


Figure 1. Smart Agriculture Overview.

present in these technologies [3]. However, IoT devices are the most vulnerable to 1
 cyberattacks because they are generally exposed to the Internet, and this can expose them 2
 to attacks that may lead to data manipulation, violation of their privacy, and prevention 3
 of their availability [4]. Cybercriminals can exploit vulnerabilities to expose one or more 4
 parts of agriculture to danger, such as damaging crops by increasing or decreasing the 5
 amounts of water or increasing the proportions of pesticides or fertilizers, and they can 6
 also violate privacy by leaking data during communication or access to the system [5]. 7

Threats to smart farms can be mitigated by using Deep learning algorithms to detect 9
 anomalies in data traffic in IoT devices and to identify abnormal or different events or 10
 observations in data traffic than normal system behavior [6,4], the use of Deep learning 11
 algorithms To detect anomalies in cloud computing is not sufficient due to the gap 12
 between IoT devices and the cloud, studies have shown that using it at the edge of the 13
 network is more efficient in solving local problems and provides less response time 14
 compared to cloud computing due to its proximity to the data source and peripheral 15
 devices, which increases its effectiveness In the speed of detection of threats and attacks 16
 on smart farms [7]. 17

1.1. Significant of Research 18

This project focuses on the most critical smart agriculture systems (See Figure 1), 19
 which play a crucial role in achieving these goals by providing farmers with real-time data 20
 about crop health, soil quality, and weather patterns. For instance, the vast amounts of 21
 data generated by IoT devices and sensors require efficient analysis to identify potential 22
 problems and anomalies. The proposed research on anomaly detection in smart 23
 agriculture systems using deep learning techniques aims to address this challenge. By 24

developing a system that can operate on the network edge, where data is collected, the proposed research can reduce latency and improve performance.

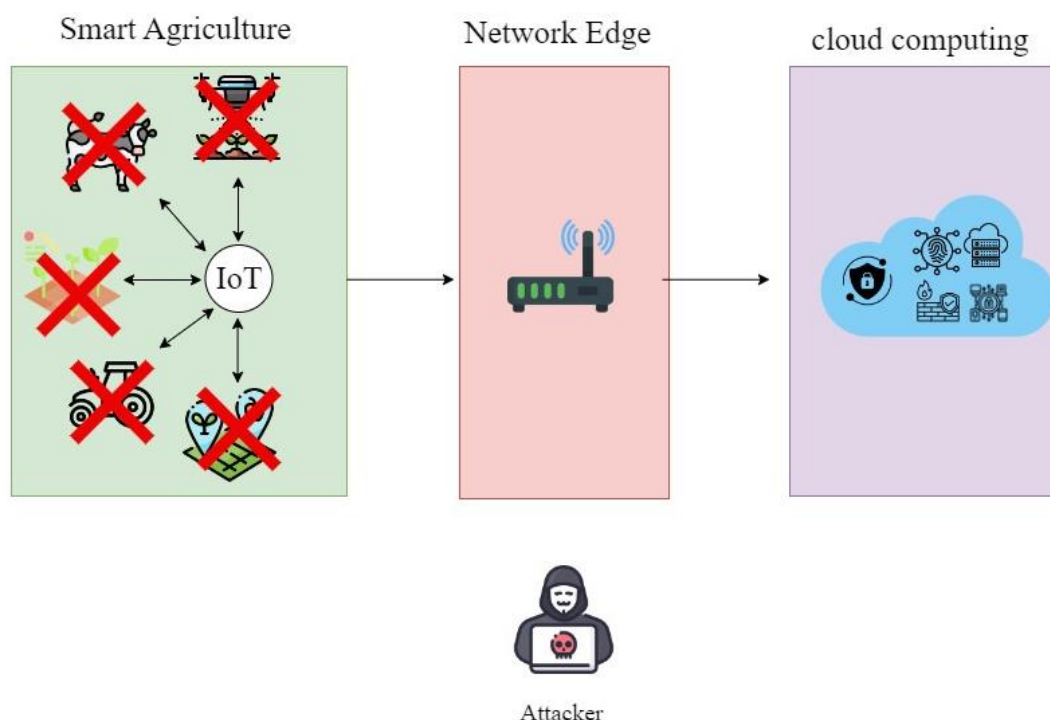


Figure 2. Smart Agriculture and DDoS attacks

This system can automatically detect anomalies in agriculture systems with high accuracy and speed, enabling farmers to take timely action to prevent or minimize crop damage. Furthermore, the research can contribute to the development of sustainable agriculture practices by improving the efficiency of resource usage and reducing environmental impact. By providing farmers with real-time data about crop health and soil quality, smart agriculture systems can help optimize the use of resources, such as water and fertilizers, and reduce waste. This can lead to significant cost savings for farmers while also reducing the environmental impact of farming.

The use of deep learning techniques, such as CNN and LSTM, can greatly improve the accuracy and speed of the proposed system. By mitigating the risks faced by smart agriculture such as DDoS attacks and others, the proposed research could have broader implications beyond the agricultural sector. IoT devices and sensor networks for data collection and analysis are becoming increasingly prevalent in other industries, such as manufacturing and healthcare. The results of this research can be applied to these industries to improve resource use efficiency, reduce waste, and improve performance.

1.2. Problem statement

Smart agriculture systems rely on the acquisition and analysis of data from IoT devices and sensors to optimize crop yields, reduce waste, and improve overall efficiency. However, if a DDoS attack is successful, it can crash the system and render it inoperable for an extended period. This can result in significant financial losses for farmers, as crop yields may be reduced, and valuable data may be lost as shown in Figure 2. Additionally,

DDoS attacks can compromise the integrity of data collected by smart agriculture systems, leading to inaccurate data analysis and decision-making. Traditional security measures, such as firewalls and intrusion detection systems, are not always successful in preventing these attacks. Moreover, we think that cloud-based anomaly detection systems are not well suited to smart farming systems, as they require high bandwidth and can lead to high latency compared to edge-based systems.

Therefore, there is a pressing need to develop an efficient and effective approach to detecting and mitigating DDoS attacks on smart agriculture systems by detecting anomalies at the network edge. Although deep learning techniques have shown promising results for anomaly detection, it is unclear whether they can be effectively employed at the network edge. This research aims to assess the feasibility of using deep learning techniques to detect anomalies in smart agriculture systems at the network edge to mitigate DDoS attacks. Additionally, the study will investigate the potential advantages of detecting anomalies at the network edge, such as reducing network latency and enhancing system performance. The proposed research will contribute to the advancement of knowledge in the field of smart agriculture systems security. Furthermore, this study's findings will provide valuable insight into the feasibility of employing deep learning techniques for anomaly detection at the network edge. Ultimately, this research will facilitate the development of effective approaches to securing smart agriculture systems against DDoS attacks, which are critical for ensuring the sustainable and efficient operation of these systems.

1.3. Research Goals and Objectives

The main goal of this research is to detect DDoS attacks at the edge network for IoT-based smart agriculture. The objectives of this research project are as follows:

- Investigate the feasibility of using deep learning techniques for detecting anomalies in smart agriculture systems at the network edge to mitigate DDoS attacks.
- Evaluate the effectiveness of various deep learning models for anomaly detection in smart agriculture systems at the network edge.
- Develop a practical solution for detecting anomalies in smart agriculture systems at the network edge to mitigate DDoS attacks.
- Study the proposed solution through experimentation and evaluation using data sets collected from IoT devices that simulate DDoS attacks.
- Contribute to the advancement of knowledge in the field of smart agriculture systems security and anomaly detection at the network edge.

1.4. Research contribution

The major of this research are summarized as follows:

- Explore the challenges and threats in smart farms and use Deep learning techniques to detect anomaly network traffic from cyberattacks on the network edge instead of central cloud computing.
- Propose a deep learning-based approach for detecting anomalies in smart agriculture systems on the network edge.

- Utilizes convolutional neural networks (CNN) and Long Short-Term Memory (LSTM) algorithms to classify sensor data collected from IoT devices into normal or anomalous classes.

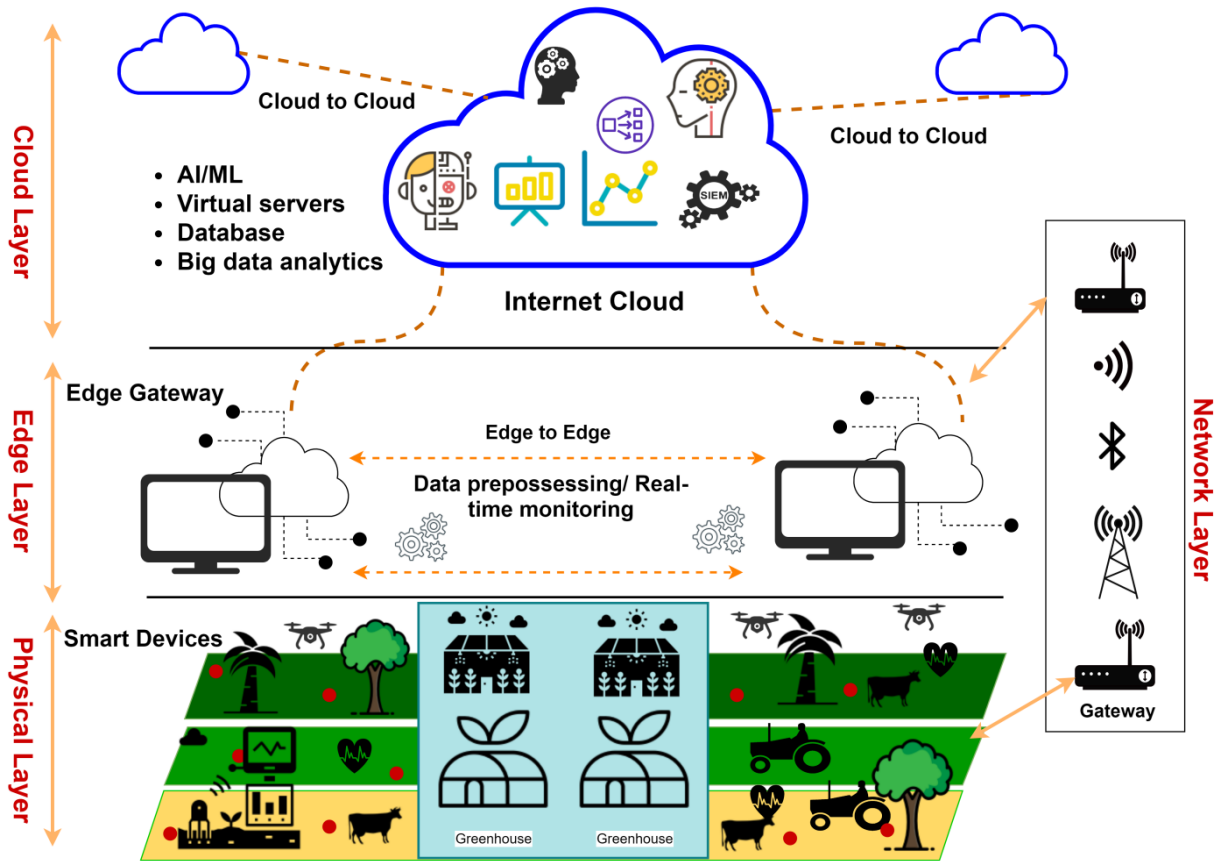


Figure 3. Smart Agriculture Architecture layers [1]

1.5. Research Organization

The remainder of this research project is structured as follows. Section 2 presents the background of smart agriculture, providing an in-depth overview of various aspects related to this field. It encompasses the Smart agriculture architecture, DDoS attacks in smart agriculture, machine learning at the network edge, and deep learning at the network edge for smart agriculture. Section 3 focuses on related work. It delves into the latest studies conducted on the detection of distributed DDoS attacks in smart agriculture. Section 4 introduces the methodology of our work, while Section 5 presents the results and related discussions. Finally, section 6 encapsulate the conclusion and future work of this study.

2. Background

This section introduces the background of smart agriculture and provides an overview of various aspects related to it. It includes the Smart Agriculture Architecture, DDoS attacks in smart agriculture, machine learning in the network edge, and deep learning in the network edge for smart agriculture.

2.1. Smart Agriculture Architecture

Smart agriculture has emerged as a major player in modern farming practices. It takes advantage of advanced technologies such as the IoT, cloud computing, and machine learning to improve crop production, improve resource management, and reduce environmental impact. However, the increasing reliance on digital technologies exposes smart agriculture systems to various cybersecurity threats [1,5]. In the Smart agriculture architecture, the layers typically include sensor devices, connectivity, data processing and analytics, and application layers. Sensor devices collect data on various parameters such as soil moisture, temperature, and crop health. The connectivity layer enables communication between the sensors and the data processing and analytics layer, which involves cloud-based or edge-based systems for data storage, analysis, and decision-making. The application layer encompasses the user interfaces and applications that provide farmers with insights, recommendations, and control over the agricultural processes [1,2, 17]. Figure 3 shows the layers of Smart agriculture architecture.

- **The Sensor Layer:** This layer is the foundation of the smart farming architecture, consisting of various sensors deployed throughout the farm. These sensors monitor and collect data on environmental conditions, soil moisture levels, temperature, humidity, and other relevant parameters. They provide real-time information to the higher layers of the architecture for analysis and decision-making.
- **The Edge Layer:** This layer acts as a vital intermediary between the Sensor Layer and the Cloud Layer, performing data processing and analysis locally. Securing this layer is essential for data protection, system performance, and defense against cyberattacks. It ensures data integrity, and real-time threat detection, and follows a layered security approach. This investment safeguards sensitive data, optimizes operations, and ensures the long-term success of smart farming.
- **The Cloud Layer:** This layer represents the centralized computing infrastructure where data from the sensors is stored, processed, and analyzed. It typically involves cloud-based platforms and services that provide advanced analytics, machine learning algorithms, and data storage capabilities. The Cloud Layer enables farmers to gain insights, make informed decisions, and optimize farming operations based on the collected data.

2.2. Importance of Edge Networks Security

Cloud computing in Smart agriculture refers to the utilization of remote servers and data centers for storing and processing agricultural data. It offers vast storage capacity and computational power, enabling complex data analysis and resource-intensive applications[3,14]. However, relying solely on the cloud for all computational tasks may introduce potential risks, including the potential for Distributed Denial of Service (DDoS) attacks that can disrupt cloud services. On the other hand, edge networks in Smart agriculture involve deploying computational resources closer to the data source, such as on farm gateways or edge devices. Edge computing brings processing capabilities closer to the agricultural field, reducing latency and dependence on remote cloud servers. It enables real-time data analysis, decision-making, and immediate responses to changing

conditions. By utilizing edge networks, Smart agriculture systems can benefit from faster response times, reduced reliance on cloud connectivity, and improved overall system resilience against DDoS attacks [7].

2.3. DDoS attack in Smart Agriculture

DDoS attacks can affect smart agriculture systems by overloading their network resources, making systems inaccessible and causing significant disruptions to farming operations, as shown in Figure 4. To protect these systems, it is necessary to develop robust and effective solutions that can detect and mitigate DDoS attacks [2,3]. Traditionally, smart agriculture systems rely on cloud computing to store, process, and analyze the huge amount of data collected from IoT devices. However, this centralization of data processing can lead to latency, and increased network congestion [3].

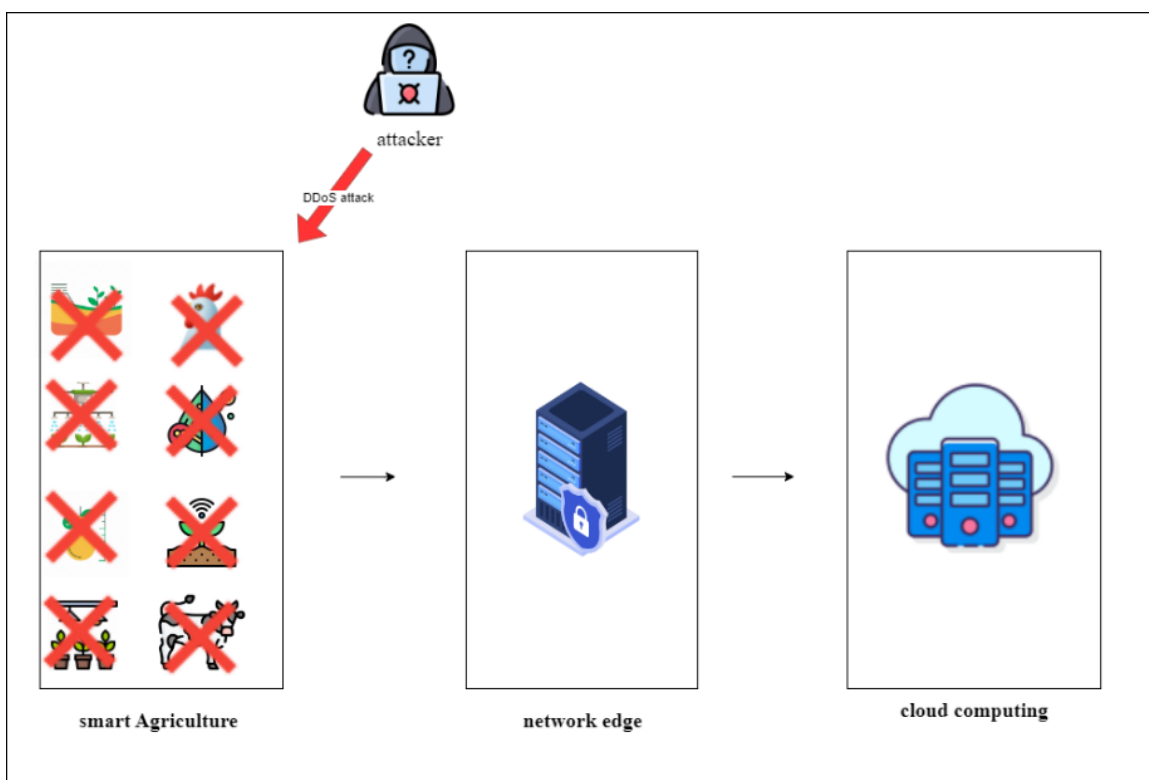


Figure 4. The impact of DDoS attacks on smart agriculture

Table 1 highlights the various forms of DDoS attacks and their impacts on smart agriculture.

Table 1 Summary of Prominent Types of DDoS Attacks and Their Impact on Smart Agriculture

Attack Type	Description	Impact on Smart Agriculture
UDP Flood	Overwhelming a target with a large volume of UDP packets	Disrupts communication between smart agricultural devices
TCP SYN Flood	Exhausting server resources by sending a flood of SYN requests	Causes service unavailability and disrupts data collection

ICMP Flood	Flooding a target with ICMP Echo Request (ping) packets	Overloads network devices and hampers remote monitoring
HTTP Flood	Overloading a web server with a massive amount of HTTP requests	Disrupts access to smart agriculture platforms and services
DNS Amplification	Exploiting misconfigured DNS servers to amplify attack traffic	Overwhelms network infrastructure and disrupts DNS resolution
NTP Amplification	Abusing Network Time Protocol servers for amplified attacks	Consumes bandwidth and disrupts time synchronization
IoT Botnets	Compromising IoT devices to form a botnet for DDoS attacks	Disrupts connectivity of IoT-based agricultural systems

These attacks have significant implications for smart agriculture. They can disrupt communication and data transfer between smart devices, leading to service unavailability and hampering remote monitoring capabilities. The overload on network devices can hinder real-time data collection and analysis [26]. Additionally, attacks targeting web servers or platforms can disrupt access to crucial agricultural services, affecting farm management and decision-making processes [30]. Network Edge offers a promising alternative by shifting data processing and analytics to devices closer to IoT devices, at the network edge. This decentralized approach reduces latency, conserves bandwidth, and enables real-time decision-making. Moreover, by processing data locally, edge computing can enhance the security and resilience of smart agriculture systems against DDoS attacks [7].

2.3. Machine Learning in Network Edge

Machine Learning (ML) has various applications, including detecting DDoS attacks on networks. ML can be used to analyze traffic and communication data to identify unusual patterns indicative of DDoS attacks. Models are trained on datasets containing normal traffic behavior and attack patterns and then used to identify and classify attacks [28]. Deep Learning (DL) is a subfield of ML that relies on deep artificial neural networks. These networks are trained to analyze data using complex methods to detect patterns and make classifications. Deep learning excels at extracting intricate information and achieving high accuracy in detection [23]. Deep learning is superior to traditional machine learning in several aspects when it comes to detecting DDoS attacks:

- Deep representation of data: Deep learning utilizes deep neural networks that represent data in a profound manner, enabling the analysis of multiple features and complex intricacies in monitored data.
- Representation learning capabilities: Deep neural networks can learn the most useful and meaningful representations for detecting DDoS attacks. They can uncover subtle and intricate patterns that are challenging to detect using traditional machine-learning approaches.

- Self-learning ability: Deep learning has the capacity for self-learning from data, allowing it to discover new patterns and information without the need for continuous training or supervision.

2.4. Deep Learning in Network Edge for Smart Agriculture

Deep learning, a subfield of machine learning, has shown remarkable success in various areas, including computer vision, natural language processing, and anomaly detection. Deep learning techniques, however, can automatically learn and extract complex patterns from input data, which makes them well-suited for detecting anomalies in network traffic [4,6]. In the context of DDoS attack mitigation, deep learning models can be trained to identify normal and malicious network traffic patterns. When deployed at the network edge, these models can provide real-time anomaly detection, allowing for rapid response and mitigation of DDoS attacks in smart agriculture systems[2,10].

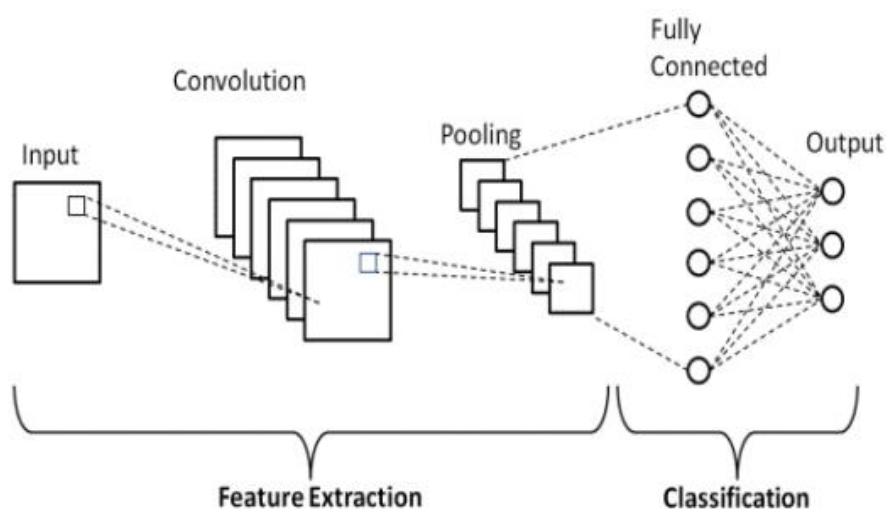


Figure 5. CNN architecture [31]

2.4.1. Convolutional Neural Network (CNN)

The CNN algorithm is a deep learning model specifically designed for processing and analyzing high-dimensional data, such as images and videos. CNNs have proven to be highly effective in extracting spatial features from data and performing complex pattern recognition tasks [31]. In the context of detecting DDoS attacks, CNNs can be utilized to analyze network traffic data and identify anomalous patterns associated with such attacks. By training the CNN on a dataset that includes both normal and attack traffic, the algorithm can learn to recognize specific patterns and behaviors that are indicative of DDoS attacks. The benefit of using CNNs for DDoS attack detection lies in their ability to automatically extract relevant features from the network traffic data. The convolutional layers of the CNN perform localized operations on the data, capturing spatial dependencies and detecting patterns at different levels of abstraction [11]. This allows the model to identify subtle variations and anomalies in the traffic flow that may indicate the presence of a DDoS attack. Additionally, the pooling layers of CNNs help in reducing the dimensionality of the data, making it more manageable for subsequent analysis. The fully

connected layers at the end of the CNN are responsible for making the final predictions based on the extracted features (See Figure 5). By leveraging the power of CNNs in detecting and analyzing patterns in network traffic data, organizations can enhance their ability to identify and mitigate DDoS attacks in a timely manner. This can help in safeguarding the availability and reliability of network services, ensuring uninterrupted operations, and protecting against potential financial and reputational damages associated with DDoS attacks.

2.4.2. Long Short-Term Memory (LSTM)

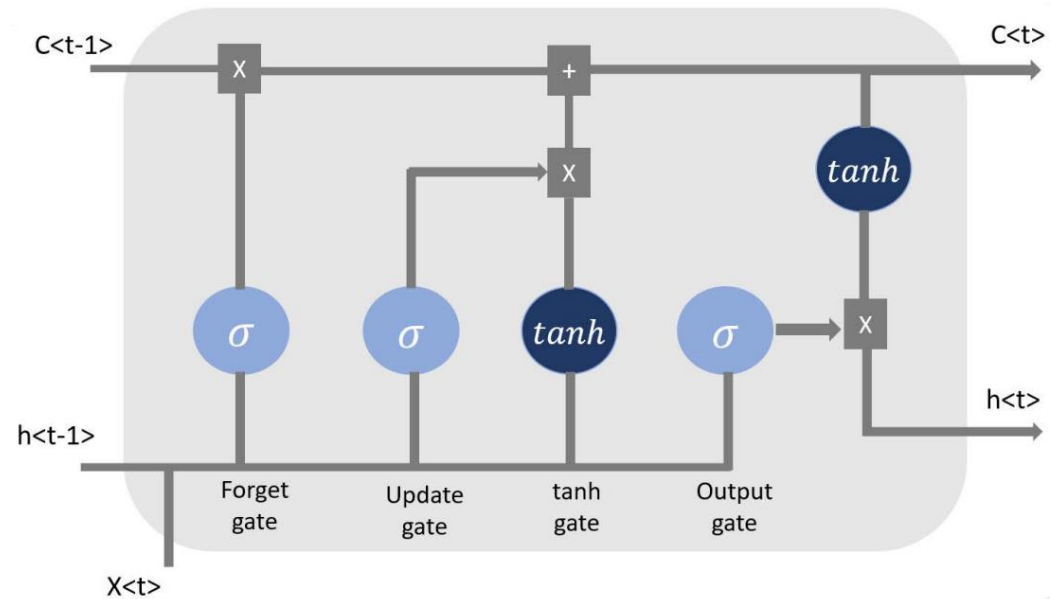


Figure 6. LSTM architecture [23]

The LSTM algorithm is a type of recurrent neural network (RNN) widely used for analyzing time series data and prediction tasks. It is effective in capturing long-term dependencies and contextual information in sequential data [10]. The LSTM algorithm consists of LSTM units, which are memory cells capable of storing and accessing information over extended time periods. These cells allow important information from previous parts of the sequence to be preserved and used for making predictions in subsequent time steps [4]. LSTM units have three main components: the input gate, the forget gate, and the output gate. These gates control the flow of information into, out of, and within the memory cells. The input gate determines which parts of the input are important for memory, the forget gate decides which information should be discarded from the memory cells, and the output gate regulates the flow of information from the memory cells to the LSTM unit's output [23]. During training, the LSTM algorithm learns to adjust the weights and biases of its gates and memory cells to minimize the difference between the expected output and the actual output. This is done through a process called backpropagation, where the error is propagated backward through time, and gradients are used to update the parameters of the LSTM units (See Figure 6).

When combining LSTM with the CNN algorithm, the system's performance can be improved by leveraging the CNN's ability to extract spatial features from input data. The

CNN is typically used as a preprocessing step to extract relevant features from the data, and then the LSTM is used to analyze the temporal dependencies and make predictions based on the extracted features. By combining the strengths of CNN and LSTM, the algorithm can capture both spatial and temporal patterns in the data, leading to improved performance in tasks such as anomaly detection and reducing the impact of DDoS attacks. The integration of CNN and LSTM algorithms in a CNN-LSTM model has been implemented, resulting in superior performance in detecting anomalies and DDoS attacks. By combining their strengths, the model captures spatial and temporal patterns simultaneously, leading to more accurate detection compared to using the algorithms separately. The CNN component extracts spatial features, while the LSTM component analyzes temporal patterns. This integrated approach enhances the model's ability to identify complex patterns, making it an effective tool for network security [23].

One reason for the decrease in detection rates when using the algorithms separately is the limited scope of each algorithm. The CNN algorithm primarily focuses on spatial features, while the LSTM algorithm primarily deals with temporal patterns. When applied individually, each algorithm may overlook important information from the other domain. By solely using the CNN algorithm, for example, the model may struggle to capture long-term temporal patterns that are crucial for detecting certain types of attacks or anomalies. Similarly, relying solely on the LSTM algorithm may result in overlooking spatial features and patterns that are indicative of attacks or anomalies. Therefore, when using the algorithms separately, the model lacks a comprehensive analysis of both spatial and temporal dimensions, leading to a lower detection rate. The integrated CNN-LSTM model addresses this limitation by leveraging the complementary strengths of both algorithms, resulting in improved performance in detecting anomalies and DDoS attacks.

3. Related work

This section explores the latest studies about the research being conducted on the detection of distributed DDoS attacks in smart agriculture. It will cover the available research regarding this issue. Abbas Yazdinejad et al., [1] discussed the importance of smart farms in enhancing global food security and the environment, and provided a classification of attacks based on their targeted components of smart agriculture such as Attacks on Hardware, Attacks on the Network and Related Equipment, Attacks on Data, Attacks on Code (applications), Attacks on Support Chain, Misuse Attacks, as well as a Cyber-Kill Chain based systematic classification of these threats. They also discussed the anatomical and behavioral characteristics of APTs and provided a survey on risk mitigation strategies and countermeasures on various layers to reduce attacks on smart agriculture. Bam Bahadur Sinha et al., [2] presented a survey on the recent advancements and challenges of IoT in smart agriculture. They also discussed the various advantages and limitations of the technology and its use in the production of smart agricultural services. They mentioned that the increasing number of IoT-linked devices and sensors in the agricultural sector is expected to have a significant impact on the environment. The authors also discuss various security concerns and attacks that threaten smart agriculture

such as DoS/DDoS, MiTM, Autonomous System Hijacking, Malware injection, Phishing, and others, They see cloud repositories as vulnerable to data manipulation and unauthorized access to resources that could disrupt the smart farming process.

Rettore de Araujo Zanella et al. [3] studied smart agriculture, Architecture, and Key Security Threats. They also believe that moving part of the security systems to the edge could lead to a reduction in the financial costs of smart agriculture. Data consumed or pre-processed at the network edge saves bandwidth and can reduce computing resources required from the cloud and protect privacy. Thus, the cloud can store and process data, make decisions, and interact with consumers. Mostafa Abdullah et al., [4] discuss the study of defect detection and failure classification of a maintenance problem in digital agriculture based on IoT and smart manufacturing. They analyzed data from sensors spread across a farm with data from seven different types of sensors and evaluated and compared the performance of semi-supervised ARIMA and LSTM models for anomaly detection. Their findings indicate that LSTM leads to a better prediction of anomaly detection than ARIMA but requires a longer training time.

The paper of Konstantinos Demestichas et al., [5] presents an overview of the evolution of ICT solutions and how they can be used and their impact on the agricultural sector, as well as a comprehensive literature review on the use of ICTs in agriculture, as well as emerging threats and associated vulnerabilities. The authors also highlight key innovations, technologies, benefits, threats, and mitigation measures of ICTs in Smart Farming. Weijun Cheng et al., [6] proposed an anomaly detection model using GAN and LMST that can process multidimensional time-series data generated by intelligent agricultural IoT. They concluded that joint training of the encoder and decoder reduced the time for anomaly detection by improving the anomaly detection performance through the use of decoder architecture. They concluded that the proposed model can detect anomalies in smart farms. Thong Voand et al., [7] studied the role of edge and fog computing in providing more efficiency than cloud computing because it provides less latency compared to its cloud computing theory. After all, the operations take place near the source of the data from which they are generated, thus providing faster data transfer speed and much less delay compared to cloud computing, which is relatively far away about the IoT devices. This process also reduces existing issues with storage and low system throughput. C. Catalano et al., [8] proposed an approach to detect anomalies in smart farming systems that, through the use of multiple sensor systems and decision support systems, can collect, analyze, and process huge amounts of data about farming. The proposed approach is based on two algorithms multivariate linear regression (MLR) and a long-term memory (LSTM) neural network algorithm, in which they applied an anomaly detection system on a real dataset from a smart agricultural system in Italy. They concluded that the proposed approach is capable of detecting anomalies. Juliet Chebet Moso et al., [9] propose an adaptation of an abnormality detector group called ELSCP. An unsupervised methodology based on the temporal data of smart agriculture, which is applied to harvest data, crop condition (damaged or undamaged), and anomaly detection. They found that 30% of the anomalies detected were related to crop damage. Therefore,

they consider the incorporation of anomaly detection into the decision-making process of farm operators necessary to improve harvesting efficiency. Yizhen Jia et al.[10] proposed an edge-centric IoT defense scheme called FlowGuard to detect, identify, classify, and mitigate IoT DDoS attacks. They also introduced a new algorithm for detecting DDoS attacks based on traffic differences to identify and classify DDoS. By generating a large dataset with the DDoS BoNeSi and SlowHTTPTest simulators, and combining it with the CICDDoS2019 dataset, to test the accuracy of identification and classification as well as the efficiency of the model. Their results indicate that the proposed LSTM can detect attacks with high accuracy.

Marcos V. O. de Assis et al.[11] proposed a security system for SDN that uses CNNs to prevent DDoS attacks. They tested the system using SDN simulation data and the CiCDDoS 2019 dataset. The results indicated that the system is promising in defending against next-generation DDoS attacks, According to their results the CNN-based approach effectively detected and mitigated these attacks in both simulated and real-world scenarios. de Araujo Zanella et al [14] discussed the challenges to food production due to climate change, water crisis, and population growth and proposed the use of a low-cost hybrid anomaly detector called CEIFA to improve reliability and safety, which can identify failures, malfunctions, errors, and attacks that may affect these systems by filtering the data sent by the agricultural system's sensors and operating On resource-limited hardware, to save computing costs. Their results show an efficiency in detecting defects. R. Chaganti, et al.[15] discuss the impact of the IoT on the smart agriculture industry and its improvement. However, they believe that the implementation of new technologies such as the IoT can also bring security risks. Therefore, they proposed a cloud-enabled smart farm security monitoring framework that can effectively monitor device state and sense anomalies while mitigating security attacks using behavioral patterns. The framework includes a blockchain-based smart contract implementation. They concluded that the framework can detect security anomalies in real-time and update other farm nodes.

Kumar, P. et al.[16] discuss the technologies used in smart agriculture, in addition to the challenges that these devices may face from data misuse and other risks to the privacy of this data. They also propose a deep learning (FL) framework based on privacy coding, called PEFL, which adopts Perturbation-based encryption and long-term memory automatic encryption technology to achieve privacy. A recurrent module neural network algorithm based on FL gates was designed using encoded data for intrusion detection, and the experimental results show that it can efficiently identify normal patterns and attack patterns. Chen, S. et al.[17] proposed an intelligent agricultural monitoring system based on an IoT cloud platform. The system includes sensors to collect data on environmental factors such as temperature, humidity, and soil moisture. The data is then transferred to a cloud platform for storage and analysis. The system also includes a decision-making module that uses data analytics to provide farmers with real-time feedback on crop growth and environmental conditions. The proposed system aims to improve crop yields, reduce resource use, and reduce the impact of environmental factors on crop growth. The study

concluded that the proposed system provides an effective solution to modernize agriculture and meet the challenges of food production in the face of climate change, water scarcity, and population growth. Adkisson, M et al. [18] proposed an autoencoder-based anomaly detection system for smart farming ecosystems. The system uses unsupervised learning to detect anomalies in the data collected by sensors in the farming ecosystem. The proposed system is implemented using an autoencoder neural network model, which learns the normal patterns of the data and can detect any anomalies that deviate from these patterns. The system is evaluated using data collected from a real-world smart farming system, and the results show that the proposed system can effectively detect anomalies in the data. The study concludes that the proposed system provides an effective solution for improving the reliability and safety of smart farming systems, and can help farmers identify potential issues in their ecosystems and take timely action to prevent crop loss and other negative impacts.

Rodríguez, J et al. [19] proposed a smart farming system called IoT-Agro for Colombian coffee farms. The proposed system aims to improve crop yields, reduce resource use, and minimize the impact of environmental factors on crop growth. The study evaluates the proposed system using data collected from a real-world coffee farm in Colombia, and the results show that the system can effectively collect and analyze data, and provide farmers with useful insights for improving coffee production. The study concludes that the proposed IoT-Agro system provides an effective and efficient way to modernize agriculture and meet the challenges of food production in the coffee industry.

Yoa, Seungdong, et al. [20] proposed a self-supervised learning method for anomaly detection in various applications, including smart agriculture. The proposed method uses dynamic local increment to generate augmented samples from the original data and train the model in a self-supervised manner without the need for labeled data. The proposed method is evaluated using two publicly available datasets, and their results show that the proposed self-supervised learning method with dynamic local reinforcement can effectively detect anomalies in various applications, including smart agriculture, without the need for classified data. Chukkapalli, S. S. L et al.[21] proposed a cyber-physical system security monitoring system for intelligent agriculture using digital twins based on a knowledge graph. The system aims to detect and prevent cyber attacks on the smart farming system by creating a digital twin of the system and monitoring it for any abnormal behavior or security threats. The proposed system uses a knowledge graph to represent the smart farming system and its components and uses machine learning algorithms to detect anomalies and potential security threats. The system is evaluated using data collected from the smart farming system in the real world, and their findings show that the proposed system can effectively detect and prevent cyberattacks on the system. Tukur, et al.[22] discuss the challenges and insider attacks in IoT environments through the use of blockchain technologies. The authors propose the use of edge-based technology and blockchain to enhance the detection of insider attacks. This is achieved by distributing sensitive data across multiple devices in the IoT, making it difficult for attackers to access

the data in its entirety. Machine learning and artificial intelligence techniques are employed to detect abnormal patterns in the data and identify potential insider attacks.

4. Methodology

This section presents or explores the proposed model, describes the framework of the model, explains the data collection process, discusses the preprocessing of data, and details the steps involved in model building and training.

4.1. Overview of the proposed model

In this section, we propose a deep learning-based approach for detecting anomalies in smart agriculture systems on the network edge. The data flow process in securing smart farms from DDoS attacks begins at the sensor device within the farm. The sensor device collects various data points, such as environmental conditions, crop health metrics, and livestock monitoring information. These data points are then transmitted to the network edge, which serves as the first line of defense against potential DDoS attacks. At the network edge, the collected data is subjected to deep learning algorithms CNN-LSTM. The CNN component is responsible for extracting spatial and temporal features from the data, while the Long Short-Term Memory (LSTM) component handles the sequential analysis, capturing dependencies over time.

The data is passed through the CNN-LSTM model, which has been trained on a large dataset of both normal and anomalous farm data. During the classification process, the model compares the input data to the learned patterns and determines whether it is normal or anomalous. If the data is classified as normal, it is allowed to continue its flow through the network. However, if the data is classified as anomalous, an alert is promptly generated to notify the farm operator or relevant personnel about the potential security breach. Simultaneously, as the alert is sent, the anomalous data is blocked from further progression within the network. This proactive measure prevents potentially malicious or harmful data from infiltrating the smart farm system and causing disruptions or damages. By employing deep learning algorithms, specifically CNN-LSTM, at the network edge, the smart farm system can effectively analyze and classify incoming data in real time. This proactive approach ensures the security and integrity of the farm's operations by swiftly identifying and mitigating potential DDoS attacks. Furthermore, the system's ability to generate alerts enables prompt responses from farm operators or security personnel, allowing them to take appropriate actions to safeguard the smart farm environment. The data flow process in securing smart farms from DDoS attacks involves the collection of data from the sensor device, passing it through the network edge, and applying deep learning techniques, such as CNN-LSTM, for real-time classification. This approach ensures that normal data is allowed to proceed while anomalous data triggers an alert and subsequent blocking. By implementing such a robust security system, smart farms can operate with enhanced protection against DDoS attacks, maintaining the integrity and uninterrupted functionality of their agricultural operations. In conclusion, using this proposed approach at the edge of the network offers several advantages. Firstly, it enables real-time analysis and classification of incoming data, allowing for immediate action in case of anomalies. Secondly, by applying the model at the network edge, the

computational burden is distributed, reducing the reliance on cloud-based solutions and potentially improving response times. Additionally, it adds an extra layer of security by detecting and blocking anomalous data at the network edge itself.

4.2. The framework of the proposed model

To identify DDoS attacks in smart agriculture, we present deep learning-based IDS models, including CNN and LSTM. The framework of the developing security systems for protecting smart agriculture is presented in Figure 7. Two public datasets, IoT-23 [13] and CICDDoS19 [29] are used to train and test the proposed models.

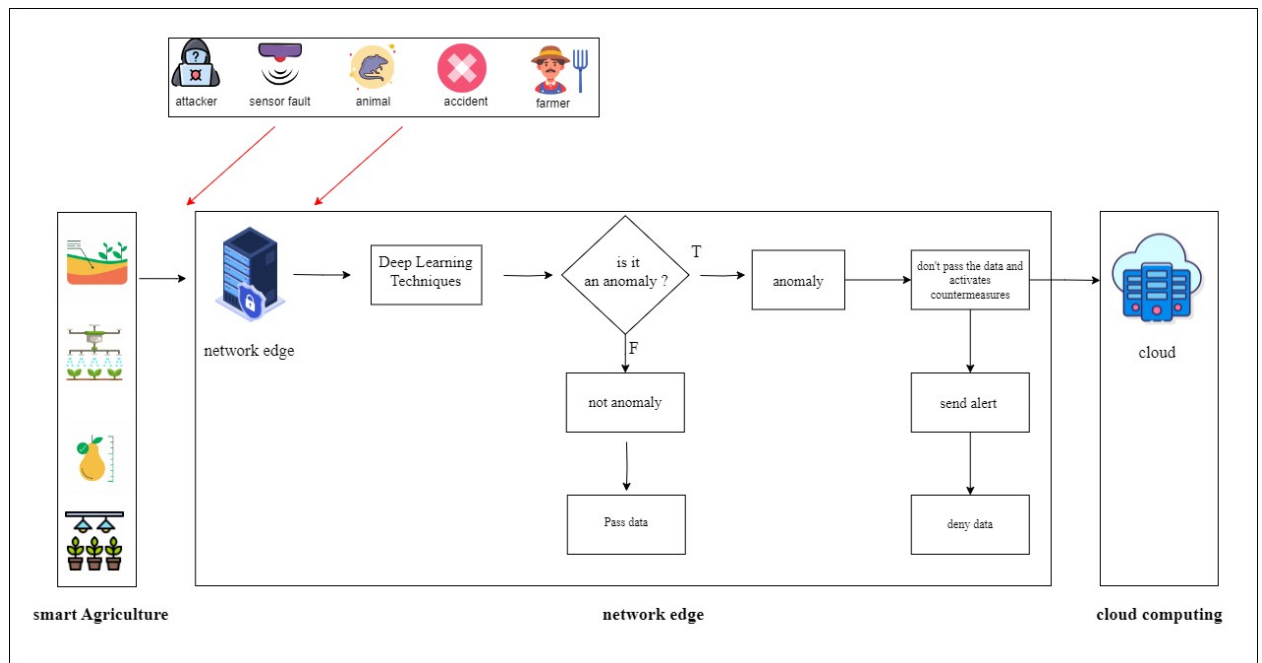


Figure 7. Diagram flow of the Data proposed into the Smart Agriculture System.

Data Stream: This step involves defining the stream of data from the source (IoT-23 and CICDDoS19 datasets) to the final output (anomaly detection). It involves identifying the data sources, data collection methods, and data processing steps required to achieve the desired output.

Data Collection: In this step, the IoT-23 and CICDDoS19 datasets are collected and prepared for analysis. This involves understanding the structure and format of the data, identifying any missing or corrupt data, and cleaning the data to ensure it is ready for processing.

Data Preprocessing: This step involves preparing the data for analysis by applying various techniques such as Data balancing, Handling Null values, and Drop Data duplicates. This is done to ensure the data is in a format that can be used by the anomaly detection algorithm. For instance, removing duplicate data, handling missing values, and normalizing the data.

Concept Drift Detection: This step involves identifying changes in the underlying data distribution over time, which could indicate a concept drift. The CNN-LMST algorithm, along with the use of a dummy encoding function, is employed to identify patterns in the

data that deviate from the expected behavior and signal any changes in the data distribution.

A Classifier: In this step, The LabelEncoder function was used to convert categorical labels into numerical representations, such as changing "DDoS attack" and "normal" to the numerical values 1 and 0. This conversion enhances the algorithms' ability to learn from the data and improves the accuracy of classification by presenting the data in a numeric form.

Clustering: In this step, the anomalous data is clustered based on the patterns of anomalous behavior. This is done to identify the source of the anomalies and understand the nature of the attack.

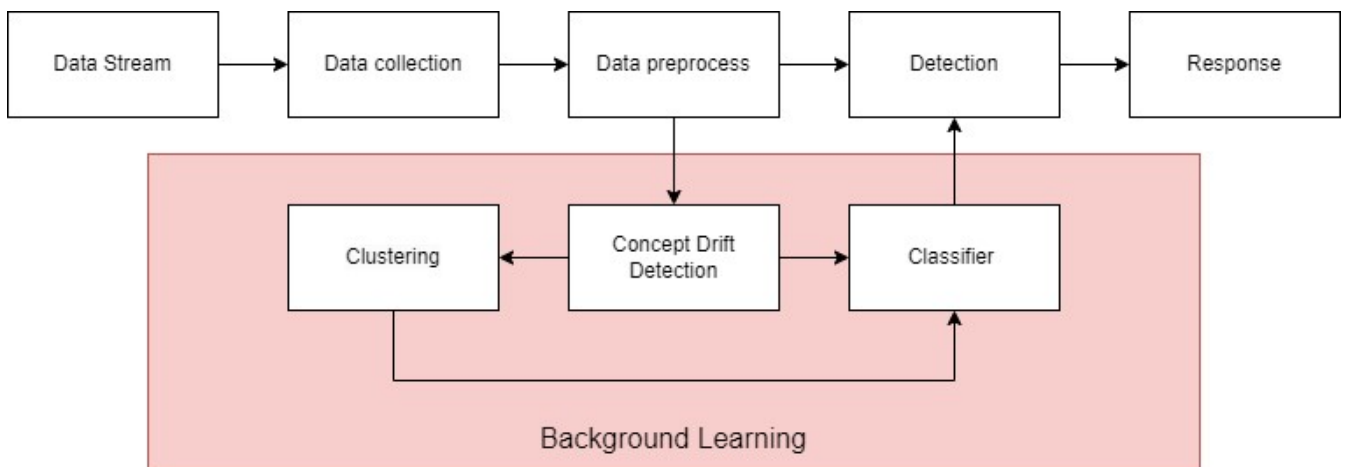


Figure 8. Framework of the proposed model

Detecting: In this step, the clustered anomalous behavior is analyzed to detect anomalous behavior in real time. If anomalous behavior is detected, an alert is generated to notify the appropriate personnel.

Response: This step involves developing an appropriate response to the detected anomaly. The response could involve shutting down the system, implementing additional security measures, or taking other appropriate action to mitigate the impact of the anomaly (See Figure 8).

4.3. Data Collection

4.3.1. IoT-23 datasets

The IoT-23 data is a comprehensive collection of network traffic data gathered from IoT devices. It contains 20 malware captures from infected IoT devices and 3 captures of benign IoT traffic. The data was captured between 2018 and 2019 at the Stratosphere Laboratory in the Czech Republic, with the current version captured in 2023. The dataset aims to provide researchers with a rich source of labeled IoT malware infections and benign IoT traffic for the development of machine learning algorithms.[13]. Each data unit in the dataset consists of several attributes of the captured packet, as well as its label. The label indicates whether the packet is normal (labeled "Benign") or represents some kind of attack (labeled according to the attack type). The attack types include C&C, DDoS, FileDownload, HeartBeat, Mirai, Okiru, PartOfAHorizontalPortScan, and Torii. For this project, we used one scenario from the dataset: "CTU-IoT-Malware-Capture-1-1." This

scenario contains over a million records (1,008,748 to be exact), which was more than sufficient for our purposes. The packets in this scenario are either the Benign (normal) or Malicious type. Each packet contains information such as source and destination IP addresses, length, size, and other details. In total, there are 23 features listed in Appendix A.

4.3.2. CIC-DDoS2019 Dataset

The CICDDoS2019 dataset is a valuable resource for DDoS research due to several significant characteristics. It offers a diverse range of DDoS attacks, allowing researchers to investigate distinct attack vectors' attributes and trends. The dataset was obtained from a simulated network environment that closely emulated real-world scenarios, making it practical and facilitating the creation of resilient defense strategies. It contains network traffic from various sources and destinations, showcasing a broad range of traffic patterns and behaviors [29]. The dataset includes various attributes for each network flow, such as source and destination IP addresses, port numbers, protocol types, and payload sizes, providing ample data for feature engineering and analysis. It also includes annotated labels for every network flow denoting the presence of DDoS attacks, enabling the implementation of supervised learning methodologies and the assessment of detection algorithms. Lastly, the dataset consists of over 15 million network flow records, indicating scalability and providing a vast amount of data for comprehensive analysis and model training.

4.4. preprocessing Data

In this study, two datasets, namely IoT-23 and CICDDoS19, were utilized, and their details are presented in Tables 2 and 3. To begin the analysis, the Scenario packets were loaded into a data frame using the Pandas library in Python. The data was preprocessed by removing any null values to ensure high-quality results.

Table 2. IoT-23 dataset attack classification

NO	Category	Volume
1	Normal	43177
2	Mirai	756
3	File download	8035
4	HeartBeat	12895
5	C&C	2381
6	Torii	33858
7	Port Scan	6544
8	DDoS	20769
9	Okiru	13718

Table 3. CICDDoS2019 dataset attack classification

NO	Category	Volume
1	Normal	21366
2	DrDoS_LDAP	199957
3	DrDoS_SNMP	199942
4	DrDoS_SSDP	199884

5	DrDoS_NetBIOS	199409
6	DrDoS_MSSQL	199347
7	TFTP	199171
8	DrDoS_UDP	199069
9	DrDoS_DNS	198249
10	UDP-lag	195886
11	DrDoS_NTP	187325
12	WebDDoS	421

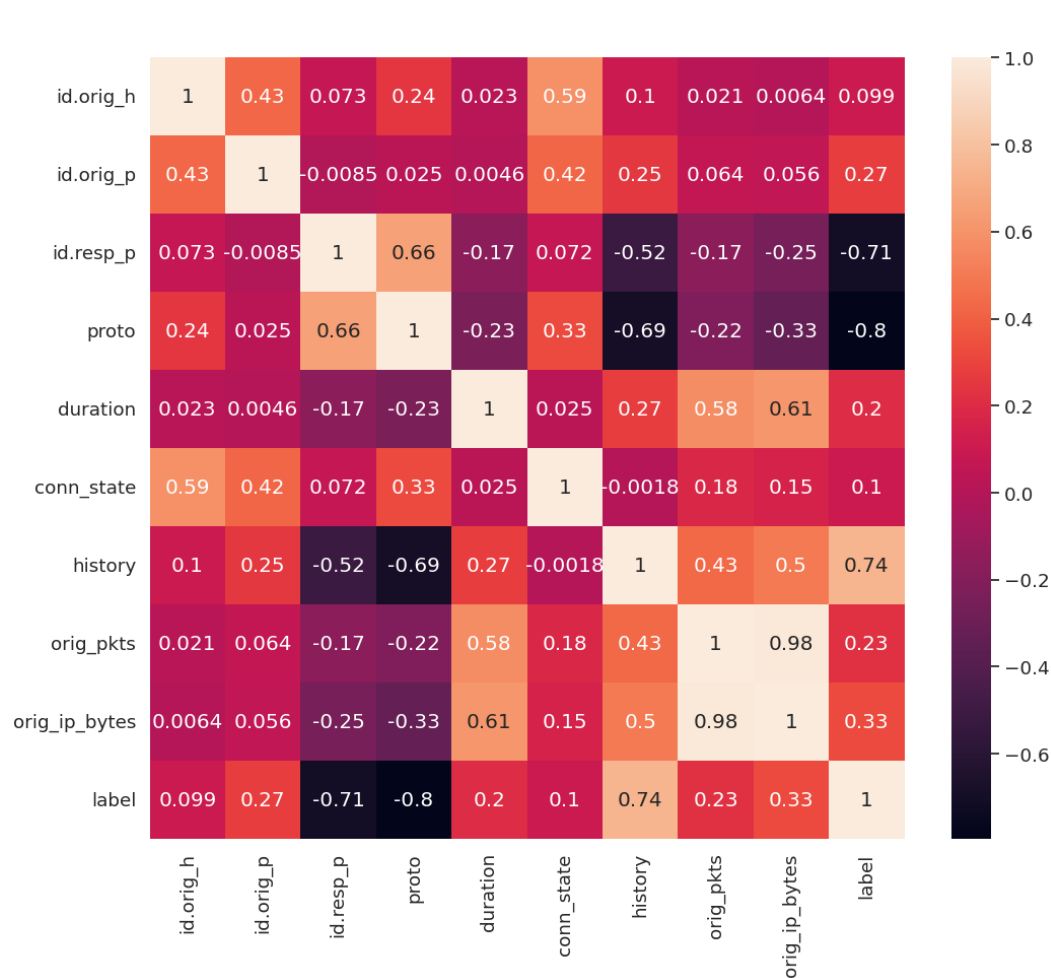


Figure 9 Correlation matrix after and before representing all the features in a numerical form of IoT-23 Dataset

The one-hot-encoding algorithm was applied to the labels, converting the Benign class into 0 and the Malicious class into 1. A correlation matrix was then calculated to analyze the dependencies between the data. Some features, such as "missed bytes," were found to be constant across all values. These features were dropped, along with any unnecessary features such as timestamps, which showed a very weak correlation with the label of the dataset. Some values were not numerical, such as the IP source and destination of the packet, and could not be included in the correlation matrix. These features were converted into their numerical representation. Other values, such as the protocol of each packet (e.g., HTTP, TCP, or UDP), were represented using the one-hot-encoding method. The correlation matrix was recalculated after representing all features in numerical form.

4.4.1 Preprocessing IoT-23 dataset

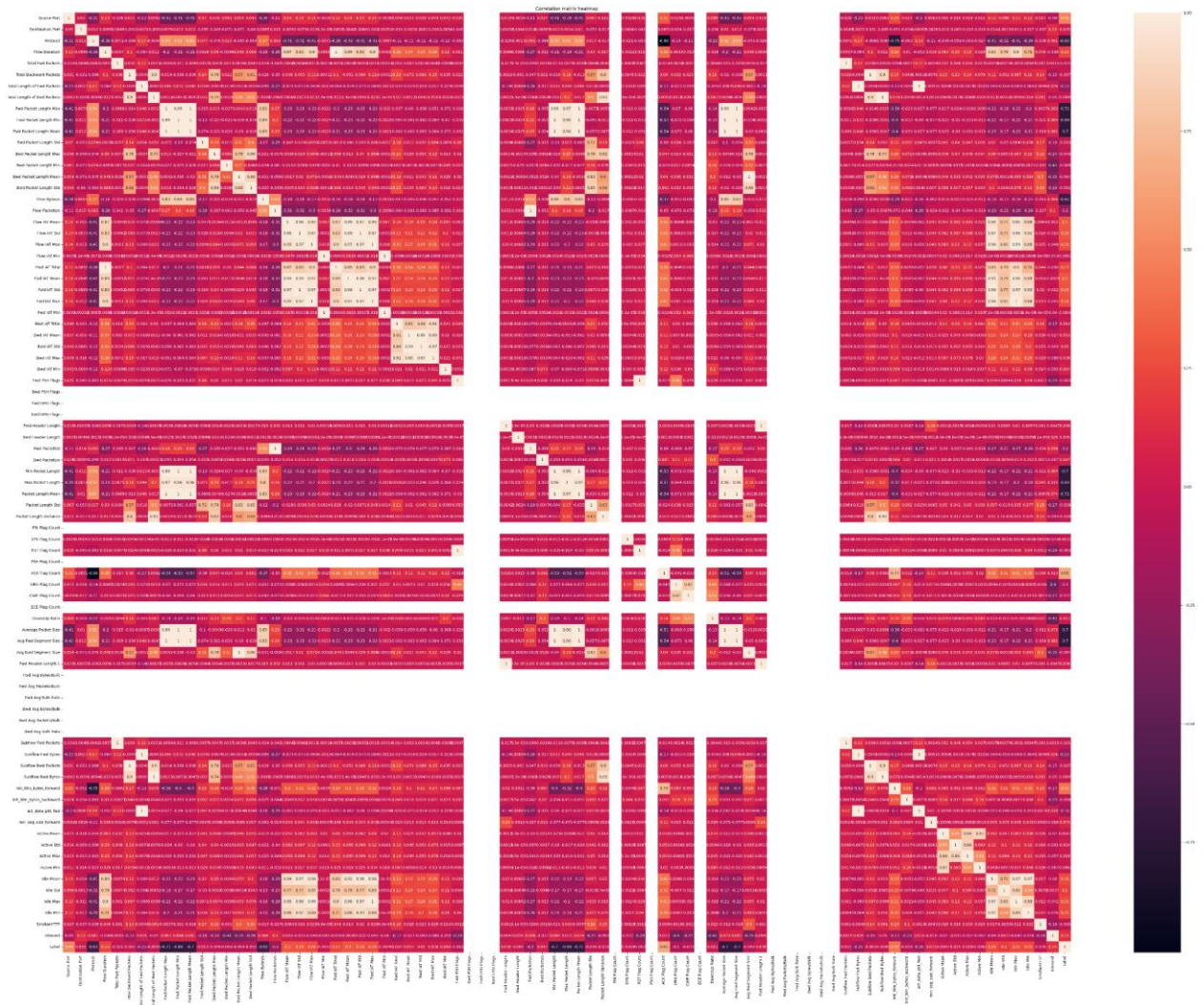


Figure 10. Correlation matrix after and before representing all the features in a numerical form of CICDDoS2019

In the IoT-23 dataset, the testing was conducted on Google Colab Pro with a Tesla V100-SXM2 equipped with 27.4GB of RAM. The dataset was initially split into training and testing sets, with 80% allocated for training and 20% for testing purposes. In the IoT-23 dataset highest 9 features in the correlation matrix with respect to the label were taken and the rest of the 21 features were dropped to speed up the computation and reduce the noise that might affect the performance of the model as shown in Figure 9. These features are listed as follow: ['proto', 'history', 'id.resp_p', 'orig_ip_bytes', 'id.orig_p', 'orig_pkts', 'duration', 'conn_state', 'id.orig_h'] These are the features that will represent the input of the model and the output will represent the type of the packet whether it is a safe packer or malicious. The data then is normalized using the max-min normalization algorithm to eliminate the possibility of one feature being dominant and negate the weight of the other features.

4.4.2 Preprocessing CICDDoS2019 dataset

In the CICDDoS2019 dataset, there was a significant difference in terms of the size of the data and the number of features compared to the first dataset. The number of features

was much larger, requiring more powerful hardware. As a result, we had to change the testing environment and conducted the test on an Amazon Web service `mi.m5.16xlarge` with 64 VCPUs and equipped with 256 gigabytes of RAM. Initially, the dataset was split into training and testing sets, with 80% devoted to training and 20% to testing. The CICDDoS2019 dataset contains 83 features as shown in Figure 10, the top 69 features in the correlation matrix were taken with respect to the label, and the rest of the features were dropped to speed up the computation and reduce noise that might affect model performance. These features are listed in Appendix B. These are the features that will represent the form input and the output will represent the type of package whether it is a safe or malicious package. The data is then normalized using a minimum normalization algorithm to eliminate the possibility that one trait is dominant and unweight the other features. In the CICDDoS2019 dataset After conducting a correlation matrix analysis, the decision was made to eliminate the four features that displayed the least amount of correlation. To decrease the dimensionality of the dataset and eliminate characteristics that exhibit minimal correlation with the target variable or other pertinent features, a correlation matrix was computed. The dataset underwent feature selection based on correlation analysis, resulting in the identification and subsequent removal of the four features with the lowest correlation. This procedure facilitates the optimization of the dataset by eliminating features that may be redundant or lack informative value.

It is often necessary to encode textual columns, such as the "Label" column, into numerical values to effectively utilize them in machine learning algorithms. During this stage, a suitable encoding methodology was implemented to transform the textual labels into numerical representations. Typical techniques for encoding data involve label encoding, one-hot encoding, or ordinal encoding, contingent upon the particular needs of the analysis. Omitting null values is a crucial step in ensuring the precision and dependability of the analysis. The absence or nullity of values can have a substantial impact on the results. Consequently, in this stage, any rows or columns that contained null values were eliminated from the dataset. In cases where data is missing, imputation methods may be utilized to complete the dataset, taking into account the type and scope of the absent information. In cases where the quantity of missing values is significant or if imputation is deemed unsuitable, the exclusion of null values is a legitimate method to safeguard the dataset's integrity. The CICDDoS2019 dataset underwent preprocessing procedures to facilitate further analysis, which involved ensuring data consistency, reducing dimensionality, and addressing missing values. The aforementioned preprocessing procedures establish the fundamental framework for proficient modeling and analysis of the given dataset.

4.5. Model Building and Training

In the IoT-23 dataset, this Keras Sequential model comprises four layers: a convolutional layer, a max pooling layer, a long short-term memory (LSTM) layer, and a dense layer. The model's input shape is defined as a tuple (9, 1), indicating that the input data consists of sequences of length 9, with each element being a single value.

- **The first layer** is a Conv1D layer that performs one-dimensional convolution operations using 512 filters of size 3 and a ReLU activation function. This layer

extracts features from the input data by applying a set of filters. The filters slide over the input data and perform element-wise multiplication followed by summation to produce a single output value for each filter. The ReLU activation function is then applied to introduce non-linearity into the model.

- **The second layer** is a MaxPooling1D layer that performs one-dimensional max pooling operations with a pool size of 5. This operation takes the maximum value over a sliding window of size 5 along the temporal dimension of the feature maps produced by the convolutional layer. This reduces the dimensionality of the feature maps while retaining the most important information.

Layer (type)	Output Shape	Param #
conv1d_12 (Conv1D)	(None, 7, 32)	128
max_pooling1d_8 (MaxPooling1D)	(None, 3, 32)	0
lstm_4 (LSTM)	(None, 32)	8320
dropout_10 (Dropout)	(None, 32)	0
dense_24 (Dense)	(None, 10)	330
dense_25 (Dense)	(None, 1)	11
dropout_11 (Dropout)	(None, 1)	0
dense_26 (Dense)	(None, 20)	40
dense_27 (Dense)	(None, 2)	42

Figure 11. The model structure IoT-23 dataset

- **The third layer** is an LSTM layer with 64 units. LSTM is a type of recurrent neural network (RNN) capable of processing sequences of inputs while maintaining an internal state. This layer processes the pooled feature maps using LSTM cells to capture temporal dependencies in the data. LSTM cells have an internal memory and use gating mechanisms to control the flow of information into and out of the cell.
- **The fourth and final layer** is a dense layer that produces the final output of the model. This layer takes the output from the LSTM layer and applies a linear transformation followed by an activation function to produce the final output. The model is illustrated in Figure 11.

In the CICDDoS2019 dataset, this Keras Sequential model comprises four layers: a convolutional layer, a max pooling layer, a long short-term memory (LSTM) layer, and a dense layer. The model's input shape is defined as a tuple (69, 1), indicating that the input data consists of sequences of length 69, with each element being a single value.

- **The first layer** is a Conv1D layer that performs one-dimensional convolution operations using 128 filters of size 3 and a ReLU activation function. This layer

extracts features from the input data by applying a set of filters. The filters slide over the input data and perform element-wise multiplication followed by summation to produce a single output value for each filter. The ReLU activation function is then applied to introduce non-linearity into the model.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 1748, 32)	128
max_pooling1d (MaxPooling1D)	(None, 874, 32)	0
lstm (LSTM)	(None, 32)	8320
dropout_2 (Dropout)	(None, 32)	0
dense_8 (Dense)	(None, 10)	330
dense_9 (Dense)	(None, 1)	11
dropout_3 (Dropout)	(None, 1)	0
dense_10 (Dense)	(None, 20)	40
dense_11 (Dense)	(None, 2)	42

Figure 12. The model structure IoT-23 dataset

- **The second layer** is a MaxPooling1D layer that performs one-dimensional max pooling operations with a pool size of 2. This operation takes the maximum value over a sliding window of size 2 along the temporal dimension of the feature maps produced by the convolutional layer. This reduces the dimensionality of the feature maps while retaining the most important information.
- **The third layer** is an LSTM layer with 100 units. LSTM is a type of recurrent neural network (RNN) capable of processing sequences of inputs while maintaining an internal state. This layer processes the pooled feature maps using LSTM cells to capture temporal dependencies in the data. LSTM cells have an internal memory and use gating mechanisms to control the flow of information into and out of the cell.
- **The fourth and final layer** is a dense layer that produces the final output of the model. This layer takes the output from the LSTM layer and applies a linear transformation followed by an activation function to produce the final output. The model is illustrated in Figure 12.

5. Results and Discussion

This section presents the results and discussion of the evaluation of different algorithms for the study. It presents the results obtained from comparing and analyzing the performance of the algorithms employed in the study.

5.1. Performance metrics

In this research, the various performance metrics are used to analyze the appropriateness of the proposed design for detecting DDoS attacks. One of these is the confusion matrix, a standard metric for assessing an IDS's effectiveness. Other evaluation metrics,

such as F1-Score, precision, recall, and accuracy, are also considered to compare our model with the existing techniques.

The CNN-LSTM models are validated using the accuracy, precision, recall, and F1 score. Accuracy is expressed as the proportion of accurately identified samples to the total number of samples. Precision is measured by the ratio of appropriately classified items to the total TP (True Positive) and FP (False Positive). The recall value is determined by calculating the overall amount of TP measurements by the total number of TP and FN (False Negative). Finally, the F1 score is computed as the weighted average of precision and recall. Additionally, we also calculate TPR, TNR, FPR, and FNR. Where TPR (True Positive Rate) refers to the number of abnormal items that test positive, the TNR (True Negative Rate) is the number of normal samples that are found to be negative, the number of normal samples that test positive is known as the FPR (False Positive Rate), and FNR (False Negative Rate) is the number of abnormal samples that test negative.

Accuracy is to calculate the level of agreement between the predicted or calculated values and the actual or expected values in a given context. It helps assess the correctness and precision of the calculations, models, or measurements being performed. Accuracy is crucial in various fields such as scientific research, data analysis, engineering, and decision-making, as it enables reliable and trustworthy results to be obtained.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (1)$$

Precision is to calculate the consistency and reproducibility of measurements or calculations. It quantifies the degree of exactness or refinement in the results obtained. Precision focuses on the level of variability or scatter in the data points or measurements, indicating how closely they cluster around each other. A high level of precision suggests minimal variation and a high degree of repeatability, while low precision indicates significant variability and lack of consistency in the measurements or calculations. Precision is important in fields such as scientific experiments, quality control, and statistical analysis, as it helps assess the reliability and consistency of the data or results being obtained.

$$\text{Precision} = \frac{TP}{(TP + FP)} \quad (2)$$

Recall is to calculate the ability of a system or model to correctly identify or retrieve relevant information or instances from a given set. It measures the completeness or the proportion of true positives that are correctly identified out of all actual positives in a dataset. Recall is particularly important in tasks such as information retrieval, classification, and pattern recognition. A high recall indicates that the system is effectively capturing a large portion of the relevant information, while a low recall suggests that there is a significant number of missed or undetected instances. Maximizing recall is crucial in scenarios where it is important to minimize false negatives and ensure comprehensive coverage of the target population or desired outcomes.

$$\text{Recall} = \frac{TP}{(TP + FN)} \quad (3)$$

F1 score is a metric that combines precision and recall into a single measure. It provides a balanced evaluation of a classification or information retrieval system's performance. The F1 score is calculated as the harmonic mean of precision and recall, ranging from 0 to 1. It is particularly useful for assessing performance in imbalanced datasets and considers both false positives and false negatives.

$$F - score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (4)$$

The CNN-LSTM model accuracy and loss were measured for both the training and validation sets at each epoch value. It allows us to assess if the model has been sufficiently learned to differentiate between various anomalies and how many data points in the validation set have been correctly identified. In addition, the accuracy and loss of the CNN and LSTM models were measured separately. **Table 4** shows model training and validation for each dataset.

Table 4 results of the LSTM-CNN model.

Algorithms	Dataset	accuracy	Precision	Recall	F1 score
CNN-LSTM	IoT-23	0.9822	0.9538	0.9041	0.9215
	CICDDoS2019	0.9947	0.9541	0.9021	0.9191
CNN	IoT-23	0.8345	0.8821	0.9026	0.8846
	CICDDoS2019	0.8136	0.7861	0.8276	0.8585
LSTM	IoT-23	0.7988	0.8536	0.7642	0.8123
	CICDDoS2019	0.8349	0.8646	0.8262	0.7724

As shown in Table 4, the difference in performance between the two datasets, IoT-23 and CICDDoS2019, is significant. The IoT-23 dataset achieved an accuracy rate of 0.9822, while the CICDDoS2019 dataset outperformed it with an accuracy rate of 0.9974. Despite both datasets being divided using the same 80/20 split for training and testing, the CICDDoS2019 dataset's considerably larger size and longer training duration likely contributed to its superior performance.

The larger size of the CICDDoS2019 dataset provided a more extensive and diverse set of samples for the deep learning model to learn from, allowing it to capture a wider range of patterns and anomalies. The increased number of samples improved the model's ability to generalize and detect anomalies accurately, resulting in a higher accuracy rate.

Moreover, the longer training duration for the CICDDoS2019 dataset allowed the model to explore and learn complex patterns more thoroughly. The additional training time enabled the model to fine-tune its parameters and adjust its internal representations, leading to improved performance and a higher accuracy rate observed.

The upward trend of the learning curve indicates that the model learns quickly and efficiently. The rate of improvement can be observed in Figures 13 and 14 below, showcasing the progressive nature of the model's learning process. The ROC curve, depicted in Figure 15, serves as a visual representation of the model's ability to effectively learn and discriminate between different classes or categories. The steep upward trend observed in the ROC curve signifies the model's rapid acquisition of accurate prediction abilities, leading to high-performance results. Consequently, the ROC curve provides valuable insights

into the model's learning capabilities and its effectiveness in accurately classifying instances.

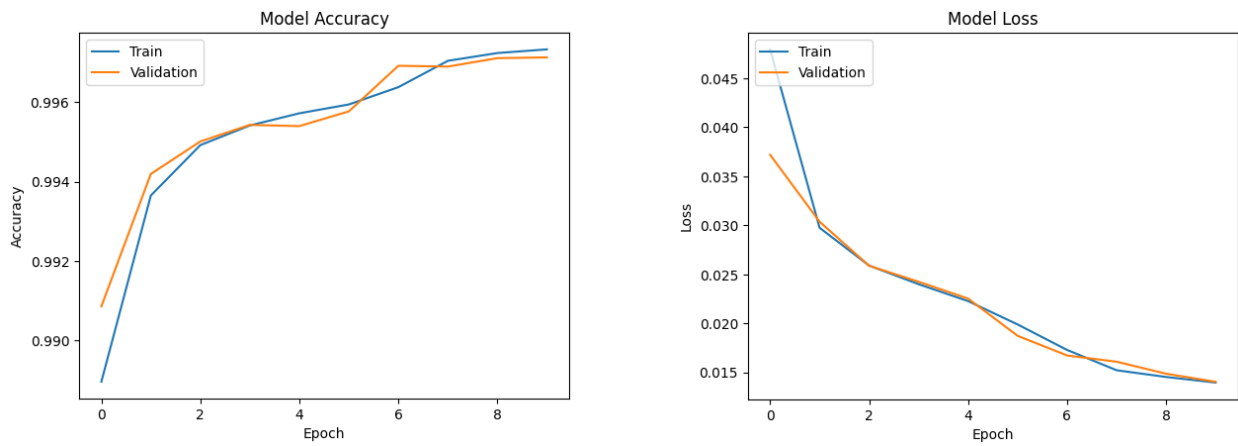


Figure 13. The CNN–LSTM model's performance for IoT-23 datasets

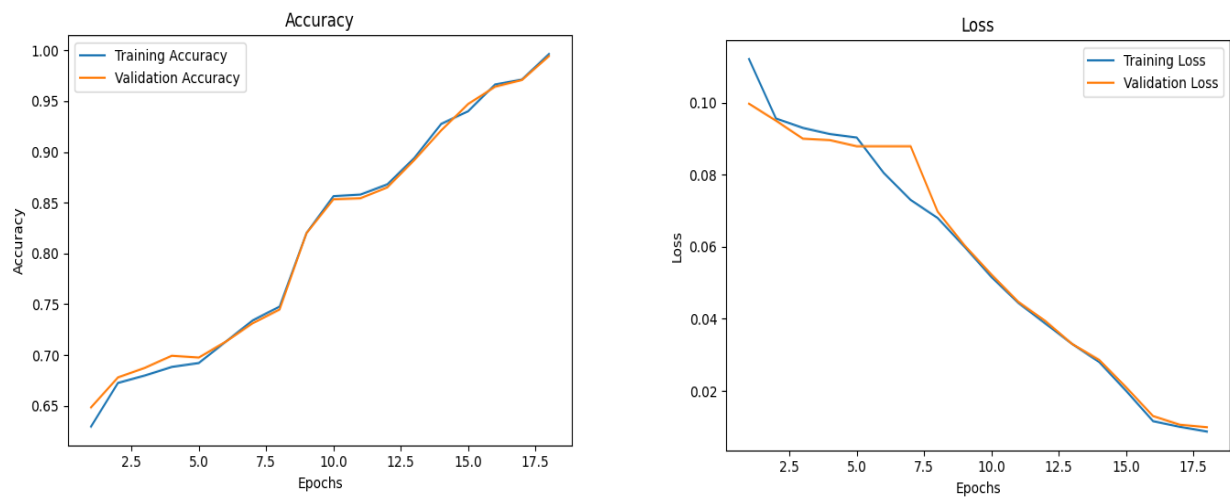


Figure 14. The CNN–LSTM model's performance for CICDDoS2019 datasets

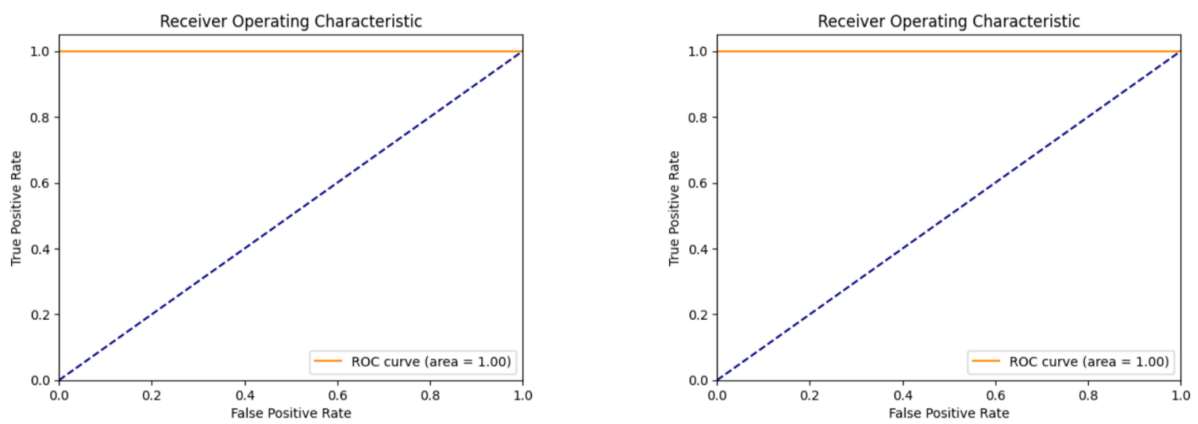


Figure 15. The ROC Curve CNN–LSTM model's performance for CICDDoS2019 datasets

The proposed CNN- LSTM models outperform all other implementations in all datasets included in this study in Table 5. This proposed model combines the strengths of LSTM and CNN to achieve exceptional performance in accurately classifying data. The LSTM component of the model excels at capturing long-term dependencies in sequential data, while the CNN component excels at extracting spatial features from the data. Together, these two components work in harmony to deliver a powerful and cutting-edge solution for data classification. Figure 16 shows a comparison between the proposed system and other systems and other state of the art systems.

Table 5. Comparison between the results of different systems in detecting DDoS attacks.

Ref.	Authors &Year	Technique	Dataset	Algorithm	Accuracy
[10]	Yizhen Jia et al 2020	FlowGuard: Intelligent edge defense mechanism against IoT DDoS attacks	CICDDoS 2019	LSTM	0.989
[11]	Marcos V. O. de Assis et al 2020	Near real-time security system applied to SDN environments in IoT networks using CNN	CICDDoS 2019	CNN	0.954
[24]	Yin, Jie, et al. 2023	anomaly traffic detection based on feature fluctuation for secure industrial IoT	IoT-23	ANN and GLM model	0.9700
[25]	Abdalgawad, Nada, et al. 2021	Generative deep learning to detect cyberattacks for the IoT-23 dataset.	IoT-23	AAE and BiGAN	0.9900
[26]	Chen, Lei, et al. 2023	An adversarial DBN-LSTM method for detecting and defending against DDoS attacks in SDN environments.	CICDDoS 2019	DBNs, LSTM, and GAN	0.9655
[27]	Dawadi, Babu R., et al 2023	Deep learning technique-enabled web application firewall for the detection of web attacks.	CICDDoS 2019	LSTM	0.9775
	The proposed model	Anomaly detection for DDoS attacks on network edge using deep learning	IoT-23 CICDDoS 2019	CNN- LSTM	0.9822 0.9947

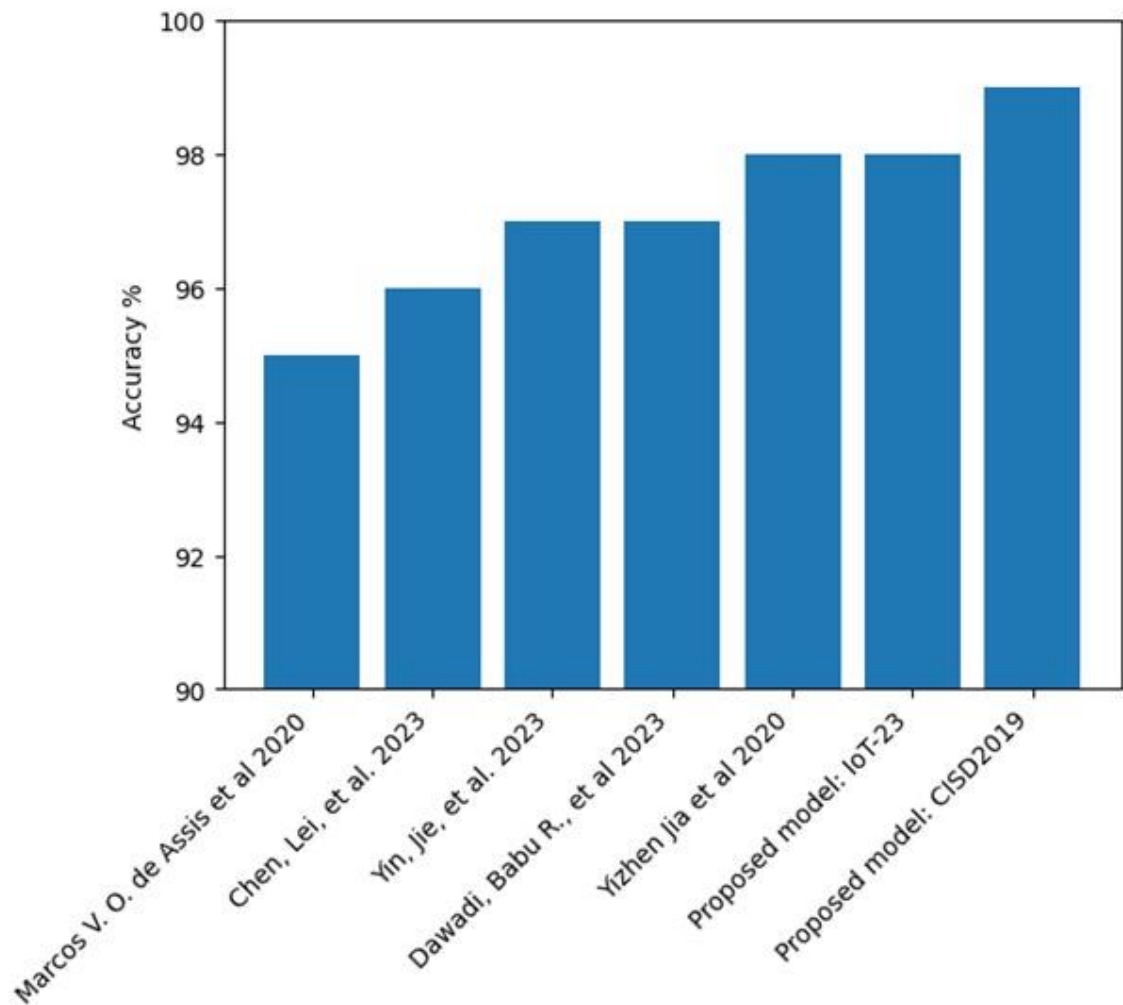


Figure 16. Visual comparison between the results the proposed mode against other models in detecting DDoS attacks.

6. Conclusions and Future Works

In this research, deep learning techniques have shown their ability to correctly classify anomalies in many areas of research. However, hackers use new and innovative techniques to launch cyberattacks. While great attempts to track down these attacks have the advantage that they continue to occur in multiple ways to cooperate with other potential attacks such as DDoS attacks and botnets. This paper proposes an anomaly detection model for IoT networks in smart agriculture using CNN-LSTM. It was performed on two different datasets, namely IoT-23 and CICDDoS2019. The proposed model achieved high accuracy for current classification strategies and modern deep learning applications. In terms of anomaly detection accuracy using the proposed model, the rate is 99.47%, and it can be more efficient if applied at the edge of the network. In future work, we will investigate further anomaly detection using various deep learning methods, such as DBN and GAN, and compare the results with those obtained with CNN-LSTM.

Author Contributions

For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “Conceptualization, B.A. and I.A.; methodology, B.A.; software, I.A.; validation, B.A., and I.A.; formal analysis, B.A.; investigation, I.A.; resources, I.A.; data curation, B.A.; writing—original draft preparation, I.A.; writing—review and editing, B.A.; visualization, I.A. All authors have read and agreed to the published version of the manuscript.

Funding

The authors would like to thank the Deanship of Graduate Studies at Jouf University for funding and supporting this research through the initiative of DGS, Graduate Students Research Support (GSR) at Jouf University, Saudi Arabia.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Conflicts of Interest

The authors declare that there is no conflict of interest in the research.

Data Availability Statement

All data presented within this manuscript is available upon request. Please contact the corresponding author for access to the data

Appendix A

A.1 The IoT-23 dataset features

['ts', 'uid', 'id.orig_h', 'id.orig_p', 'id.resp_h', 'id.resp_p', 'proto', 'service', 'duration', 'orig_bytes', 'resp_bytes', 'conn_state', 'local_orig', 'local_resp', 'missed_bytes', 'history', 'orig_pkts', 'orig_ip_bytes', 'resp_pkts', 'resp_ip_bytes', 'tunnel_parents', 'label', 'detailed-label']

A.2 The CICDDoS2019 dataset features

["Label, Packet Length Mean, Fwd Packet Length Max, Avg Fwd Segment Size, Fwd Packet Length Mean, Average Packet Size, Min Packet Length, Fwd Packet Length Min, Max Packet Length, ACK Flag Count, Protocol, Flow Bytes/s , Source Port, Init_Win_bytes_forward, Fwd IAT Mean, Flow IAT Mean, Flow IAT Std, Fwd IAT Std, Fwd IAT Max, Flow IAT Max, Idle Mean, Idle Max, Idle Min, Fwd IAT Total, Flow Duration, Subflow Fwd Bytes, Total Length of Fwd Packets, Fwd Packets/s, Idle Std, Flow Packets/s, act_data_pkt_fwd, Inbound, URG Flag Count, CWE Flag Count, min_seg_size_forward, Bwd Packet Length Min, Bwd Packet Length Mean, Avg Bwd Segment Size, Bwd IAT Min, Fwd PSH Flags, RST Flag Count, Init_Win_bytes_backward, Bwd IAT Mean, Bwd Packet Length Std, Bwd IAT Std, Bwd Packets/s, Bwd Packet Length Max, Bwd IAT Max, Packet Length Std, Fwd Header Length.1, Fwd Header Length, SimilarHTTP, Destination Port, Active Std, Total Fwd Packets, Subflow Fwd Packets, Packet Length Variance, Active Max, Bwd IAT Total, Fwd Packet Length Std, Down/Up Ratio, Total Length of Bwd Packets, Subflow Bwd Bytes, Total Backward Packets, Subflow Bwd Packets, Active Min, Flow IAT Min, Bwd Header Length, Fwd IAT Min, Active Mean, SYN Flag Count"]

References

- [1]. Yazdinejad, A., Zolfaghari, B., Azmoodeh, A., Dehghantanha, A., Karimipour, H., Fraser, E., ... & Duncan, E. (2021). A review on security of smart farming and precision agriculture: Security aspects, attacks, threats and countermeasures. *Applied Sciences*, 11(16), 7518.
- [2]. Sinha, B. B., & Dhanalakshmi, R. (2022). Recent advancements and challenges of Internet of Things in smart agriculture: A survey. *Future Generation Computer Systems*, 126, 169-184.
- [3]. de Araujo Zanella, A. R., da Silva, E., & Albin, L. C. P. (2020). Security challenges to smart agriculture: Current state, key issues, and future directions. *Array*, 8, 100048.

- [4]. Abdallah, M., Lee, W. J., Raghunathan, N., Mousoulis, C., Sutherland, J. W., & Bagchi, S. (2021). Anomaly detection through transfer learning in agriculture and manufacturing IoT systems. arXiv preprint arXiv:2102.05814. 1
- [5]. Demestichas, K., Peppes, N., & Alexakis, T. (2020). Survey on security threats in agricultural IoT and smart farming. *Sensors*, 20(22), 6458. 2
- [6]. Cheng, W., Ma, T., Wang, X., & Wang, G. (2022). Anomaly Detection for Internet of Things Time Series Data Using Generative Adversarial Networks With Attention Mechanism in Smart Agriculture. *Frontiers in Plant Science*, 13. 3
- [7]. Vo, T., Dave, P., Bajpai, G., & Kashef, R. (2022). Edge, Fog, and Cloud Computing: An Overview on Challenges and Applications. arXiv preprint arXiv:2211.01863. 4
- [8]. Catalano, C., Paiano, L., Calabrese, F., Cataldo, M., Mancarella, L., & Tommasi, F. (2022). Anomaly detection in smart agriculture systems. *Computers in Industry*, 143, 103750. 5
- [9]. Moso, J. C., Cormier, S., de Runz, C., Fouchal, H., & Wandeto, J. M. (2021). Anomaly Detection on Data Streams for Smart Agriculture. *Agriculture*, 11(11), 1083. 6
- [10]. Jia, Y., Zhong, F., Alrawais, A., Gong, B., & Cheng, X. (2020). Flowguard: An intelligent edge defense mechanism against IoT DDoS attacks. *IEEE Internet of Things Journal*, 7(10), 9552-9562. 7
- [11]. de Assis, M. V., Carvalho, L. F., Rodrigues, J. J., Lloret, J., & Proença Jr, M. L. (2020). Near real-time security system applied to SDN environments in IoT networks using convolutional neural network. *Computers & Electrical Engineering*, 86, 106738. 8
- [12]. Nandy, S., Adhikari, M., Khan, M. A., Menon, V. G., & Verma, S. (2021). An intrusion detection mechanism for secured IoMT framework based on swarm-neural network. *IEEE Journal of Biomedical and Health Informatics*, 26(5), 1969-1976. 9
- [13]. Sebastian Garcia, Agustin Parmisano, & Maria Jose Erquiaga. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]. Zenodo. <http://doi.org/10.5281/zenodo.4743746> 10
- [14]. de Araujo Zanella, A. R., da Silva, E., & Albin, L. C. P. (2022). CEIFA: A multi-level anomaly detector for smart farming. *Computers and Electronics in Agriculture*, 202, 107279. 11
- [15]. Chaganti, R., Varadarajan, V., Gorantla, V. S., Gadekallu, T. R., & Ravi, V. (2022). Blockchain-based cloud-enabled security monitoring using Internet of Things in smart agriculture. *Future Internet*, 14(9), 250. 12
- [16]. Kumar, P., Gupta, G. P., & Tripathi, R. (2021). PEFL: Deep privacy-encoding-based federated learning framework for smart agriculture. *IEEE Micro*, 42(1), 33-40. 13
- [17]. Chen, S., Xu, F., Wang, S., Huang, Y., Wen, P., Huang, D., & Zhao, S. (2021, June). A Smart Agricultural Monitoring System Based on Cloud Platform of Internet of Things. In 2021 IEEE International Conference on Prognostics and Health Management (ICPHM) (pp. 1-8). IEEE. 14
- [18]. Adkisson, M., Kimmell, J. C., Gupta, M., & Abdelsalam, M. (2021, December). Autoencoder-based anomaly detection in smart farming ecosystem. In 2021 IEEE International Conference on Big Data (Big Data) (pp. 3390-3399). IEEE. 15
- [19]. Rodríguez, J. P., Montoya-Munoz, A. I., Rodríguez-Pabon, C., Hoyos, J., & Corrales, J. C. (2021). IoT-Agro: A smart farming system to Colombian coffee farms. *Computers and Electronics in Agriculture*, 190, 106442. 16
- [20]. Yoa, S., Lee, S., Kim, C., & Kim, H. J. (2021). Self-supervised learning for anomaly detection with dynamic local augmentation. *IEEE Access*, 9, 147201-147211. 17
- [21]. Chukkappalli, S. S. L., Pillai, N., Mittal, S., & Joshi, A. (2021, November). Cyber-physical system security surveillance using knowledge graph based digital twins-a smart farming usecase. In 2021 IEEE International Conference on Intelligence and Security Informatics (ISI) (pp. 1-6). IEEE. 18
- [22]. Tukur, Y. M., Thakker, D., & Awan, I. U. (2021). Edge-based blockchain enabled anomaly detection for insider attack prevention in Internet of Things. *Transactions on Emerging Telecommunications Technologies*, 32(6), e4158. 19
- [23]. Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., & Ahmad, R. (2022). CNN-LSTM: hybrid deep neural network for network intrusion detection system. *IEEE Access*, 10, 99837-99849. 20
- [24]. Yin J, Zhang C, Xie W, Liang G, Zhang L, Gui G. Anomaly traffic detection based on feature fluctuation for secure industrial internet of things. *Peer Peer Netw Appl*. 2023 Apr 26:1–16. doi: 10.1007/s12083-023-01482-0. Epub ahead of print. PMID: PMC10131526. 21
- [25]. Abdalgawad, N., Sajun, A., Kaddoura, Y., Zualkernan, I. A., & Aloul, F. (2021). Generative deep learning to detect cyberattacks for the IoT-23 dataset. *IEEE Access*, 10, 6430-6441. 22
- [26]. Chen, L.; Wang, Z.; Huo, R.; Huang, T. An Adversarial DBN-LSTM Method for Detecting and Defending against DDoS Attacks in SDN Environments. *Algorithms* 2023, 16, 197. <https://doi.org/10.3390/a16040197> 23
- [27]. Dawadi, B.R.; Adhikari, B.; Srivastava, D.K. Deep Learning Technique-Enabled Web Application Firewall for the Detection of Web Attacks. *Sensors* 2023, 23, 2073. <https://doi.org/10.3390/s23042073> 24
- [28]. Santos, R., Souza, D., Santo, W., Ribeiro, A., & Moreno, E. (2020). Machine learning algorithms to detect DDoS attacks in SDN. *Concurrency and Computation: Practice and Experience*, 32(16), e5402. 25
- [29]. Available online: <https://www.unb.ca/cic/datasets/ddos-2019.html> (accessed on 2 Jan 2023) 26
- [30]. Salim, M. M., Rathore, S., & Park, J. H. (2020). Distributed denial of service attacks and its defenses in IoT: a survey. *The Journal of Supercomputing*, 76, 5320-5363. 27
- [31]. Allugunti, V. R. (2022). A machine learning model for skin disease classification using convolution neural network. *International Journal of Computing, Programming and Database Management*, 3(1), 141-147. 28



1

Copyright: © 2023 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

2