NABEA AL-AJYAL FOUNDATION PRESS
Online ISSN: 3078-5413
CyberSystem Journal (CSJ)
csj.nabea.pub
CyberSystem Journal Research Article

Leveraging deep Learning for Efficient Intrusion Detection in IoT Networks

Noor Thamer Mahmood [1*], Suhad Hatem Jihad[2], Sumar Mohamed Khaleel[3] and Ahmed Saleem Abbas [4]

[1,2,3] Computer Center, University of Babylon, Babylon, Hilla, 51001,Iraq
[4] Prof. PhD, Software Department. College of information technology, university of Babylon, Science of College, University of Hilla; Babylon, Hilla, 51001,Iraq

* Corresponding Author: **Noor Thamer Mahmood,** Email: nour.thamer95@uobabylon.edu.iq.

Abstract. Internet of Things (IoT) technology is experiencing rapid development and increasing use in a variety of applications, making it a potential target for cyber-attacks. Machine learning and deep neural network techniques are an effective way to address these challenges and improve IoT security. This research aims to design a deep learning techniques for intrusion detection in an Internet of Things environment with limited resources. The research focuses on improving the efficiency and effectiveness of current model using artificial intelligence and LSTM algorithms, ensuring reliable and effective security in the IoT environment. The proposed model is evaluated using a realistic data set, Canadian Institute for Cybersecurity Internet of Things 2023 Dataset (CICIoT2023) devices, and using performance metrics, namely Accuracy, Precision, F1 Score, and Recall. The results show its compatibility and effectiveness in a real environment, with 99.1% accuracy recorded. This paper is considered an important contribution to the field of IoT security and provides an effective methodology for developing advanced security solutions in the IoT environment that enhance traffic analysis, identify abnormal behavior, and take the necessary measures.

Access this article online

Keywords: Deep learning, CICIoT2023 dataset, IoT, balancing methods.

1. Introduction

The Internet of Things (IoT) illustrated in Figure 1, represents a paradigm shift in modern computing, connecting billions of devices—including sensors, actuators, and embedded systems—to share data and perform tasks autonomously. The IoT is now deeply integrated into many sectors, such as healthcare, manufacturing, transportation, and smart cities. However, the increasing ubiquity and complexity of IoT systems have also introduced significant security vulnerabilities, particularly due to their resource-constrained nature, heterogeneous protocols, and limited built-in defense mechanisms.

As IoT networks continue to expand, the cyber threats targeting them have become more sophisticated, frequent, and difficult to detect. Common attacks include denial of service (DoS), data exfiltration, botnet deployment, and unauthorized access. Traditional intrusion detection systems (IDS), which rely heavily on static rules or signature-based detection, often fail to generalize well to sophisticated and previously unseen attacks. This creates an urgent need for intelligent, adaptive, and lightweight IDS solutions that operate efficiently in real-time and under the resource constraints typical of IoT environments.

Current intrusion detection techniques suffer from limitations in IoT environments due to high false positive rates, poor adaptability to new threats, and a lack of contextual learning from sequential traffic patterns. Furthermore, IoT datasets are often imbalanced and noisy, impairing the learning ability of traditional models [1].

Received February 12, 2025; Revised March 2, 2025; Accepted May 10, 2025; Published June 30, 2025
https://doi.org/10.57238/csj.2025.1006

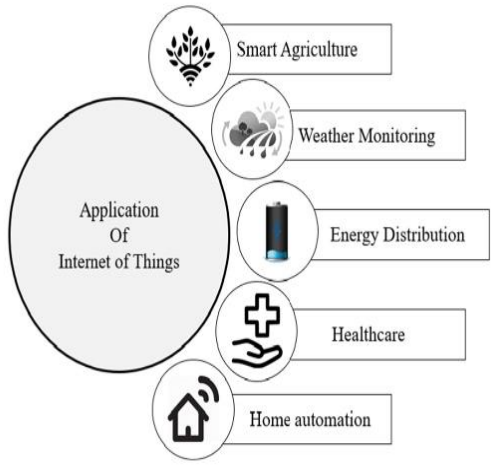

Figure 1 Applications of Internet of Things [1].

This paper aims to leverage deep learning, specifically the Long Short-Term Memory (LSTM) architecture, to build a robust IDS framework specifically designed for IoT networks. LSTM networks are known for their ability to learn temporal dependencies in sequential data and are well-suited for modeling network traffic patterns over time and distinguishing between normal and malicious behavior. The goal is to train and evaluate an LSTM-based model on the CICIoT2023 dataset—a modern, realistic IoT traffic dataset—by applying a structured preprocessing pipeline to address missing values, balancing the class distribution using Synthetic Minority Oversampling Technique (SMOT), and normalizing features using a min-max metric.

The paper hypothesizes that an LSTM model, when trained on a properly preprocessed and balanced IoT dataset, can achieve high detection accuracy with minimal false positives, outperforming traditional rule-based or shallow learning models in intrusion detection tasks.

The remaining sections of the paper are organized as follows:

- Section II provides a detailed review of related work and current IoT intrusion detection techniques.
- Section III outlines the proposed methodology, including dataset preparation, preprocessing steps, and LSTM model configuration.
- Section 4 presents experimental results and analyses, including performance evaluation and comparison with related models.
- Section 5 concludes with key findings and future research directions

1.1 Internet of Things Architecture

Especially in large IoT networks, where there are challenges related to data integrity and confidentiality. The number of security concerns such as exposing them to cyber-attacks has increased [2]. Constant, unmanaged exposure to the Internet can leave devices and the underlying network vulnerable to various types of attacks. As Internet-connected IoT devices grow, the point of attack and the potential risk of these devices being compromised and exploited in unwanted cyberattacks also increases.

The three-layer IoT architecture, as shown in figure 2, consists of the following main layers: perception layer, network layer, and application layer. We will highlight each layer in the following context.

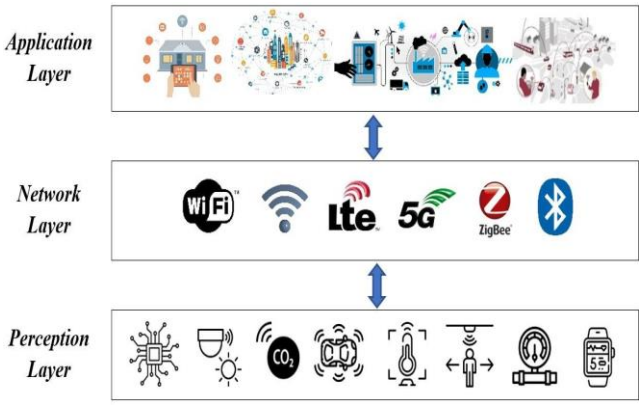

Figure 2. IoT layers Architecture [3].

1.1.1 Perception layer

The perception layer consists of devices, such as sensors, that enable it to optimize its surrounding environment. This layer is like the sensory system of things, where devices collect information about the surroundings and send it across the network. Devices in this layer include, for example, temperature and humidity sensors and medical devices. Besides sensing functions, this layer also includes action execution devices, such as actuators, that react and execute required commands[4].

1.1.2 Network layer

The network layer is a means of transferring data from the perception layer to the application layer via specific paths. Its primary function is to receive data from devices in the perception layer and route it through integrated networks to the application layer. Network and mobile technologies used include IEEE802.11, 4G, 5G, Bluetooth, and Zigbee. In addition, this layer also includes the network management process to ensure the smooth and error-free operation of IoT systems[5].

1.1.3 Application layer

This layer provides application services to users and subscribers. The layer uses the context collected from the lower layers to provide smart applications such as smart homes, e-health, and smart transportation to end users. This layer is an essential part of the IoT system as it collects information from the underlying technologies to provide useful and easy-to-use applications to end users.

The Internet of Things aims to connect things to a network and collect and process the information provided by these things. IoT networks are enabled by capabilities that are logically classified into two main categories: device capabilities and gateway capabilities. Hardware capabilities include direct and indirect interaction with the network, while gateway capabilities include support for multiple interfaces and protocol switching [3-6].

1.1.4 Security Attacks in the IoT Network – Perception Layer Environment

Security challenges in the perception layer environment of the Internet of Things become really important because of the device and connection diversity plus a large number of connected devices. Since their main function is to collect data, then they become vulnerable to data forgery and theft as well as attacks on the information gathered by the IoT devices; also, it may lead to damage to the IoT devices and make the network completely unavailable.

1. Waterway attacks: A compromised node in the perception layer advertises tempting false power, computation, and communications capabilities to the nearby nodes such that they divert data traffic to the node. This then permits the attacker to collect data traffic or distort data traffic before delivering it to the application layer.
2. Node capture breaches: This attack discloses sensitive information, such as group keys and radio keys from the compromised node, therefore compromising the security of the entire system.
3. Injecting malicious codes: By exploiting vulnerabilities and injecting malicious codes, the attacker could assume control of a node or device at the perception layer, thereby leading to unauthorized operations.
4. 4. False data injection attacks: Gain on the node or device is taken; thereafter, instead of real data, false data is injected, negatively impacting the system's effectiveness.
5. Replay attacks: Successful legitimate identification information acquisition from the source host enables a follow-up attack.
6. Eavesdropping: Wireless device-to-device communication can be utilized by an attacker to capture useful data.
7. Sleep Deprivation Attacks: The smart node in the perception layer has its battery drained through increased power consumption or actions it was not supposed to take during its sleep period, which is how sleep deprivation attacks work. These attacks would require strong security strategies to be considered in the list of actions that need to be taken to secure against security threats in the environment of the Internet of Things perception layer. An entity, whether legitimate or not, can assert its identity through multiple identifiers at the disposal of it, thereby muddling the perception layer. E.g., an illegitimate entity can communicate with several other entities to enhance its standing and even con the entire system into drawing false conclusions.
8. In a black hole attack, the attacker tries to create artificial packet loss at the perception layer. For this, packets are delivered by the compromised node that represents an IoT where packets are delivered, which should not be said forward to the next node. This can be very harmful when coupled with a denial of service attack because due to one more such attacks, the compromised node can singly more nodes.
9. Finally, Denial of Service (DoS) attacks, the main goal is to exhaust the resources of the perception layer to make the entire IoT or a specific node unavailable. For example, jamming attacks can disrupt communication between IoT sensors and the gateway, resulting in disruption to services provided to users. These attacks can be carried out by sending high-range signals to overload the communication channel between devices, or by flooding the gateway with forged data.

In addition, there are important security requirements for IoT and gateway, such as communications security, data

management security, service delivery security, integration of security policies and technologies, mutual authentication and authorization, and security auditing. The gateway should implement these requirements to ensure security and privacy in the IoT system.[7-9].

1.1.5 IoT Security and Privacy Requirements

According to ITU-T Recommendation Y.2066, a list of security and privacy protection requirements for IoT is provided. These requirements refer to the functionality required during the capture, storage, transmission, collection, and processing of object data, as well as the provision of services involving objects. These requirements are relevant to all players in the IoT field. Below are these requirements[15-18]:

- Communications security: There must be a secure, reliable, and privacy-protected communication capability, which allows blocking unauthorized access to data content, ensures data integrity, and protects privacy-related data content during transmission or transfer over the Internet of Things.

- Data management security: There must be a safe, reliable, and privacy-protected data management capability, which allows blocking unauthorized access to data content, ensures data integrity, and secures privacy-related data content when stored or processed in the Internet of Things.

- Service delivery security: The possibility of providing a secure, reliable, and privacy-protected service must be provided, which allows blocking unauthorized access to the service and enables illicit service provision, and protects the privacy information of IoT users.

- Integration of security policies and technologies: Different security policies and technologies must be standardized to ensure consistent security control over a variety of devices and user networks in the Internet of Things.

- Mutual authentication and authorization: Before accessing the IoT, mutual authentication and authorization must be performed between the device (or IoT user) and the IoT according to defined security policies.

- Security Audit: IoT must be supported by security audit to ensure transparency, traceability, and redundancy of data transmission, storage, processing, and application access.

In addition, ITU-T Recommendation Y.2067 provides specific security requirements that a gateway should implement, which helps achieve security in an IoT system. As shown in Figure 3.

1.1.6 Security Considerations

As we increasingly rely on interconnected smart devices in our daily lives, the risk of these devices or “things” being targeted by attacks and intrusions is also increasing, potentially leading to device malfunctions and putting our privacy and safety at risk. So, security turns out to be a major challenge to consider along with safety in IoT. This follows closely on the physical world. In addition, the issue of managing IoT devices points to the challenges you may face in terms of accountability, diversity that characterizes the IoT ecosystem, and scalability issues. There are many different concerns that hinder the standardization of secure IoT ecosystems, including large-scale attacks, limited hardware resources, diversity in the ecosystem, fragmentation of standards and regulations, widespread deployment, security updates, insecure programming, and unclear obligations in liability guidance [19-20].

2. Related Works

The last ten years have seen a growing interest in securing Internet of Things devices from cyber threats. This has led to extensive research in machine learning and deep learning. A number of studies have been conducted to improve the accuracy of intrusion detection while reducing the number of false positives. For example, the study proposed a deep learning ensemble methodology that included CNN, GRU, and LSTM which when used together achieved an accuracy of 99.7% based on data from NSL-KDD. Another example is a study that proposed a lightweight dense random neural network (DnRaNN) that achieved an accuracy of 99.14% based on ToN_IoT. While another study proposed a CNN-BiLSTM model with batch normalization which improved the accuracy to 96.3% on NSL-KDD. More studies used artificial neural networks to detect IoT threats with an 84% accurate result like in the study conducted by and showed that Decision Tree and Random Forest models could hit 100% accuracy based on the CICIOT2023 dataset. Another study that used LSTM for the same purpose is conducted by Chaganty et al. in the year 2022 with a result of achieving 97.1% accuracy in intrusion detection in SDN-IoT networks. Further recent developments include a lightweight model DL-BiLSTM [3] and the architecture of Bi-GRU-CNN (2023) having results better than the traditional way of detecting malware. These

studies have collectively strengthened the security of IoT, leading to high robustness and accuracy in detecting system-intruded cyber threats[20-26].

In conclusion, studies conducted by many researchers emphasize the critical importance of securing Internet of Things (IoT) devices against cyber threats and unauthorized access. Each study proposes innovative approaches, leveraging machine learning (ML) and deep learning (DL) techniques, to enhance intrusion detection and cybersecurity in IoT environments. These methodologies address diverse challenges, such as resource constraints, data complexity, and the evolving nature of cyber threats. Specifically, studies highlight the effectiveness of ensemble-based deep learning models, lightweight neural networks, and LSTM-based methods in detecting and mitigating IoT-based attacks. Collectively, these studies contribute valuable insights and solutions to enhance the security and resilience of the Internet of Things. Ecosystems against cyber threats in an expanding digital landscape [27-28].

3. Methodology

This study designs and experimentally evaluates an intelligent intruder detection model for the Internet of Things (IoT) environment. The proposed model uses a deep neural network model known as Long Short-Term Memory (LSTM), which belongs to a subclass of recurrent neural networks specialized for time series data. The methodology begins by clarifying the research problem of the poor performance of legacy intrusion detection systems in detecting advanced and unknown cyberattacks within heterogeneous IoT networks where data sources are multiple. The real-world CICIoT2023 dataset is used, which simulates 33 different types of attacks distributed across more than 105 IoT devices; therefore, it has a highly representative and realistic data structure.

The first steps in preprocessing were to address missing data and noise by removing them, then balancing the class distribution using the synthetic minority oversampling(SMOTE) to eliminate the class imbalance problem. Applying Min-Max normalization helps standardize the range of numeric values, thus making the learning model more stable. The data is then split into two sets for training and testing at a ratio of 80/20, respectively.

The proposed model is built using an LSTM layer with 64 memory cells, a dropout layer to mitigate overfitting, and a sigmoid output layer to support binary classification. The model is trained using the Adaptive Moment Estimation **(**Adam**)** optimizer and the Binary Cross entropy function as a loss metric. Performance is evaluated using important metrics such as precision, recall, accuracy, and performance metric that balances precision and recall**(** F1 score**)** . The model was developed in Google Collab using TensorFlow, Kera's, and Scikit-learn for ease of implementation and reproducibility. Furthermore, this performance is compared to standard classification methods, including Random Forest and SVMs. This comparison aims to determine whether the proposed model outperforms in terms of its handling of time series data and generalization efficiency.

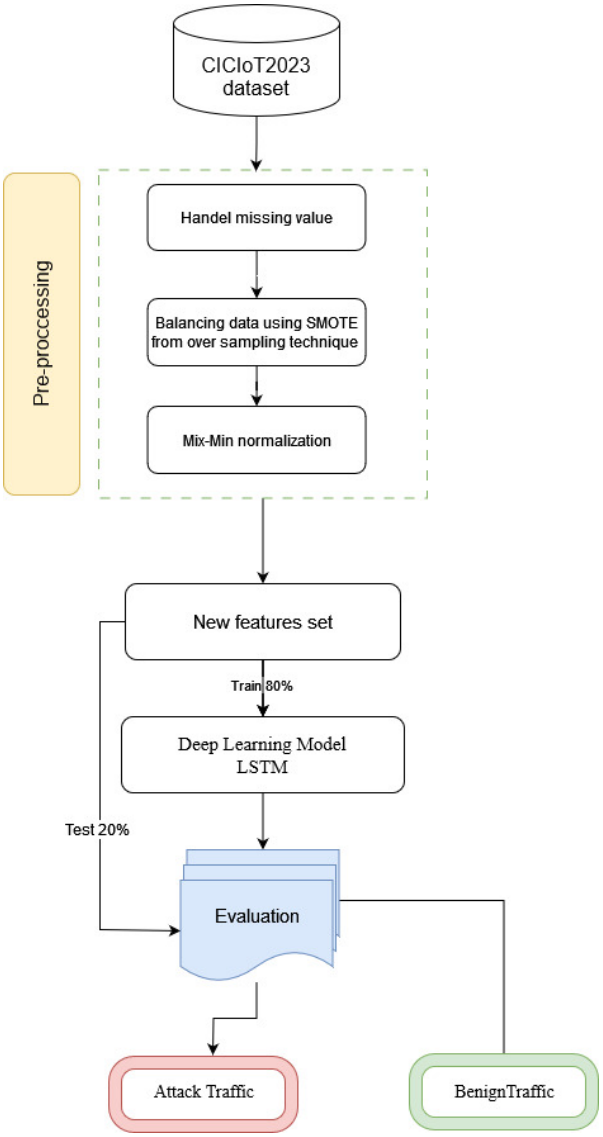

Figure 3. The layout of the proposed methodology.

Figure 3 in this section shows the proposed methodology for building an artificial intelligence model that focuses on detecting cyber-attacks and intrusions using deep learning techniques.

The main stages include defining the problem, collecting data, pre-processing for consistency, selecting relevant features for the Artificial Intelligence (AI) model, selecting an appropriate machine learning algorithm based on data characteristics, and training and evaluating the selected models. Finally, the optimized model is deployed to obtain realistic predictions in fraud detection and user behavior evaluation.

The study uses LSTM with the "CICIoT2023" dataset. Model performance is evaluated using metrics of precision, precision, recall, and F1 score for a comprehensive evaluation.

A visualization of the proposed approach is shown in Fig. This methodology provides a well-organized strategy for developing the proposed deep AI model, which contributes to early mitigation of security risks. The proposed model consists of several main stages, which will be explained in the following section:

3.1 Dataset

A dataset referred to as "CIC IoT Dataset 2023" which is considerably new, was brought into formation to support programs of analysis concerning security within the IoT environment. This dataset comprises 33 varieties of attacks carried out on more than 105 IoT devices and is split into seven main categories. The first six are DDoS, DoS, Web-based, Brute Force, Spoofing, and Mirai attacks; the malicious IoT devices on other IoT devices carry these out. The last one is Recon.

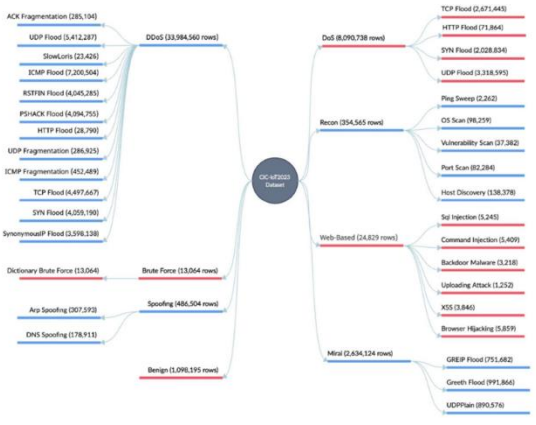

Figure 4. Dataset overview[41].

Figure 4 presents numerous IoT cyber-attacks as well as an equal number of dataset rows hence posing a threat to cybersecurity concerning the availability and integrity of computer systems and networks. The attacks comprise floods such as UDP and ICMP Floods and hash-based attacks which are incorporated in DDoS attacks. A disruption of service is caused by flooding a single source with traffic; web-based attacks use SQL Injection and XSS, among many other related web application attacks. Brute force attacks try to gain unauthorized access to something by determining a username and password combination, while other attacks try to impersonate network traffic or actual entities. In the end, Mirai attacks typically focus on IoT devices and include GRE IP, and UDP Plain attacks.

The dataset gives a full detailed preview about the attributes and behaviors about network traffic packets-timestamp, flow duration, protocol type, data transfer rate, number of tags, etc., so that it will help network professionals to comprehensively and accurately examine the network performance and security posture [29] .

3.2 Preprocessing

" CIC IoT Dataset 2023" is imbalanced nature. Figure 5 shows unbalancing " CIC IoT Dataset 2023" dataset.

```
Out[5]: target
        1    1401208
        0      33836
        Name: count, dtype: int64
```

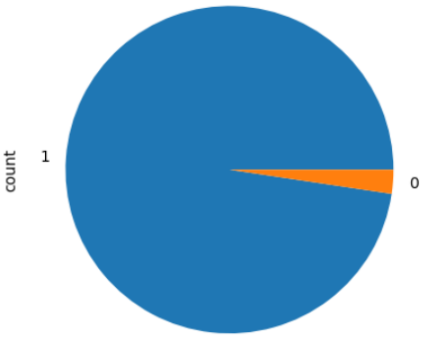

Figure 5. distribution of classes in unbalanced dataset

To address class imbalance in the CICIOT2023 dataset, the SMOTE technique is applied. First, the minority class within the data set is identified, which typically represents the least frequent class. Preprocessing involves separating features and the target variable. SMOTE is then used. Artificial samples are created for the minority class. The original dataset is then combined with these synthetic samples to create a balanced dataset[30]. This approach aims to mitigate the class imbalance problem, which may improve the model's performance on tasks such as intrusion detection or classification. Figure 6 shows the distribution of classes in the dataset after applying the SMOT method.

```
target
1    1401208
0    1401208
Name: count, dtype: int64
```

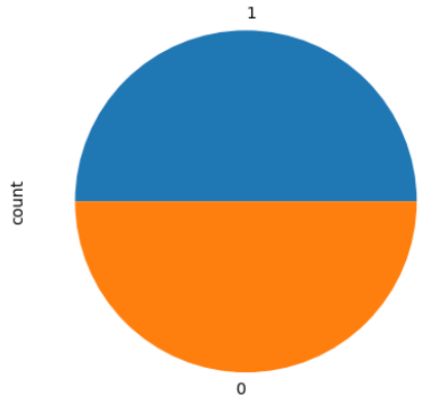

Figure 6. distribution of classes in balanced dataset.

3.3 Normalization

An additional preprocessing procedure involves min-max normalization, often referred to as feature scaling. This method entails applying a linear transformation to the data, effectively rescaling it within a range of (0, 1) [31]. The normalization process is carried out in accordance with equation (1):

$$x\ new = \frac{x - min\ (x)}{(x)\ - min\ (x)} \quad (1)$$

Where x new represent normalized x.

3.4 Splitting dataset

The data set is divided into two parts: a training part and a validation part. The training portion (80%) is used to train the neural network of the proposed model. Training data is fed to the neural network, and the weights and parameters are updated during the training process using capture and optimization algorithms. Between training rounds, the remaining portion (20%) allocated to validation is used to evaluate the network's performance and verify its ability to handle new data that was not used in training. Performance metrics, such as accuracy rate and confusion matrix, are used to evaluate the model's performance and effectiveness in dealing with the tasks at hand.

4. Machine learning

Machine learning, which is the basic part of AI, enables algorithms to learn from data and make decisions. These comprise supervised and unsupervised learning and reinforcement learning. In supervised learning, models are built for making predictions based on labeled data; in unsupervised learning, data is used for finding patterns, and in reinforcement learning, the model makes optimum decisions based on the interaction between an agent and an environment. The supervised methods covered include support vector machines, decision trees, and logistic regression, with the last being used when the output variable is continuous. This structure gives very good tools for data analysis, prediction, and decision-making in different application [30].

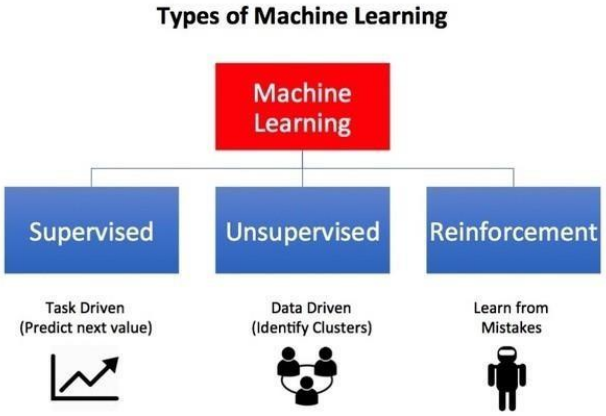

Figure 7. Type of Machine learning [36].

The deep learning algorithm LSTM was chosen, as described in the next section.

4.1 Long Short-Term Memory (LSTM)

The architecture of long-term memory (LSTM) neural networks has evolved over time and is described by the most common architecture shown in figure 8. The LSTM unit comprises within the cell core three gates, which govern the information flow and state of the cell. These gates are the input gate, output gate, and forget gate. Those cells are interconnected with each other which means that all information which is used as a memory has to be inside one LSTM cell.

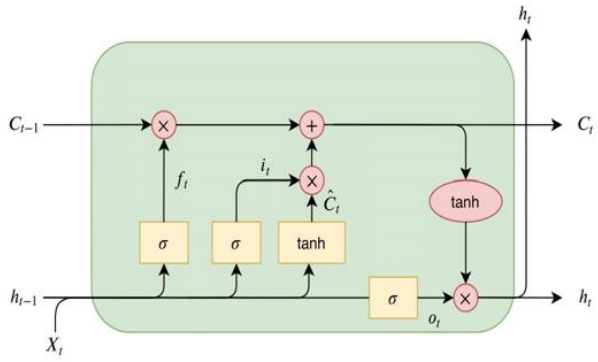

Figure 8. LSTM cell architecture [33].

This is the structure of an LSTM cell representing the input-output time step, where X is the input, h is the output, C is the cell state, and f is the forget gate. The processes are shown as dots inside the light red circle. These gates are based on common sense:

a) The forget gate directs the cell to "forget" or discard information from its internal state.

b) The cell is routed through an input gate, where new information is stored in the cell's internal state.

c) Next, the cell emits what is known as an output gate, which is a filtered output of the cell's internal state.

$$f = \sigma(Wf \cdot [h-1, X] +) \quad (2)$$

$$= \sigma(W \cdot [h-1, X] + b) \quad (3)$$

$$O = \sigma(WO \cdot [h-1, X] + bO) \quad (4)$$

$$C\hat{} = \tanh(WC \cdot [h-1, X] + bC) \quad (5)$$

Then, the internal cell state is computed as

$$C = \cdot C\hat{} + f \cdot C - 1 \quad (6)$$

The final output from the cell, or h, is then filtered with the internal cell state as

$$h = \cdot tanh(C) \quad (6)$$

Weights and biases are associated with each gate, much like in neural networks. To enable an LSTM cell to learn, these weight matrices are integrated with gradient-based optimization. In the above equations, weight matrices and biases are denoted by symbols , bf , W, b, W, b, and WC, bc, respectively. An Recurrent Neural Network(RNN)/LSTM network retains data from previous time steps and generates predictions for time series by sequencing these cells, as illustrated in Figure 9. The network can address the vanishing gradient problem by utilizing the LSTM cell topology. Older RNN designs struggled to provide accurate predictions for time series due to this issue.

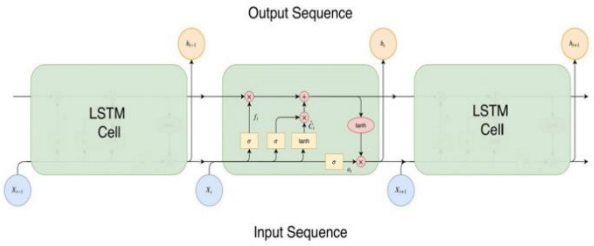

Figure 9. Standard LSTM model [33].

Rectified Linear Unit is a rather new activation function in neural networks. Here is the math for it: output = 0 if input < 0, output = input. The choice of Non-linearity for DL models, as well as being computationally efficient, in practice turns out to make vanishing gradient less likely to occur. Essentially, deep networks that have ReLUs in them learn much faster as the gradients are backpropagated much more effectively in deep networks with a positive regime due to the flow whenever the input is positive. Owing to this, ease of programming, and possible highest effectiveness, ReLU has turned out to be the activation function of choice for the deep learning community. The only change made to the above code is that the function used to activate the LSTM layer is "relu". For multi-class classification problems, the "softmax" function is a common activation function for the output layer of neural networks. It maps real-valued vector scores to probability scores, though the actual mathematical equation computed by the softmax function involves normalizing the real-valued vector scores. It then goes on to interpret the components of the output vector as probabilities of membership to classes. The requirement of softmax arises when multiple classes need to be considered and the decision lies with one of them based on the highest probability in multi-class classification problems. According to Chen, Kim, and Gideon, in most cases, a combination of softmax and cross-entropy is used to train neural network classifiers. This is just the cross-entropy between the output of the neural network and the ground truth label, and then its parameters are adjusted with back propagation to minimize cross entropy with changed parameters. that the network learns to model the relationship between input and label via a loss function softmax with cross-entropy, is not apparent, even if it is sensible to minimize the cross-entropy between labels and predicted probabilities. In the output layer of the LSTM model, the “softmax” activation function was used [31].

5. Result Analysis and Discussion

This study trains and tests the model that is being proposed using the LSTM algorithm with a batch size of 512,

over a period of two epochs. In the training phase, the input features and target labels are packed into batches of training data to constantly update the model's parameters iteratively. A simple progress bar belonging to the current epoch and tracking the training loss using 'train_on_batch' method, helps visualize the progress of the training phase. This is followed by an evaluation phase for each epoch. In this, the evaluation dataset is divided into batches to speed up the computation and have insight into the dynamical process. Moreover, classification metrics based on the confusion matrix distributions are applied and presented below. Figures [10-14] and table 1 show the results of the proposed model on the CICIoT2023 dataset.

```
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_6 (LSTM)               (None, 46, 50)            10400

 dropout_18 (Dropout)        (None, 46, 50)            0

 simple_rnn_12 (SimpleRNN)   (None, 46, 50)            5050

 dropout_19 (Dropout)        (None, 46, 50)            0

 simple_rnn_13 (SimpleRNN)   (None, 50)                5050

 dropout_20 (Dropout)        (None, 50)                0

 dense_12 (Dense)            (None, 50)                2550

 dense_13 (Dense)            (None, 1)                 51

=================================================================
Total params: 23101 (90.24 KB)
Trainable params: 23101 (90.24 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/2
70061/70061 [==============================] - 1570s 22ms/step - loss: 0.0119 - accuracy: 0.9906 - val_loss: 0.0076 - val_accu
acy: 0.9919
Epoch 2/2
70061/70061 [==============================] - 1575s 22ms/step - loss: 0.0102 - accuracy: 0.9920 - val_loss: 0.0081 - val_accu
acy: 0.9909
70061/70061 [==============================] - 501s 7ms/step - loss: 0.0079 - accuracy: 0.9911

accuracy: 99.11%
```

Figure 10. LSTM model training

Figure (10) depicts the model training process across two training cycles. It demonstrates clear stability in the process of minimizing the error function value over a short period of time, indicating efficient data preprocessing and good tuning of training parameters such as the learning rate and the choice of the number of cells in the LSTM layer.

```
17516/17516 [==============================] - 132s 8ms/step
17516/17516 [==============================] - 134s 8ms/step
Accuracy: 0.990913
Precision: 0.992163
Recall: 0.989621
F1 score: 0.990890
Cohens kappa: 0.981826
ROC AUC: 0.990913
[[278397   2188]
 [  2905 276994]]
```

Figure 11. LSTM model testing.

Figure (11) displays the results of testing the model on data not used during training. The model demonstrates excellent ability to generalize the acquired learning to new data, confirming the model's robustness in real-world applications in IoT environments.

Table 1 Performance of proposed model.

Details	Result Result %%
Accuracy score	**99.1%**
Precision score	**99.2%**
Recall score	**98.96%**
f1_score	**99.08%**

These results in table 1 indicate a very strong classification performance for the proposed model. The high specific accuracy (99.2%) reflects the model's ability to avoid false positives, an important aspect in security systems to avoid false alarms that can weaken the system's response. In contrast, the recall rate (98.96%) indicates the model's high ability to capture most real attack cases, reducing the likelihood of missing a real threat (false negatives), a pivotal point in intrusion detection.

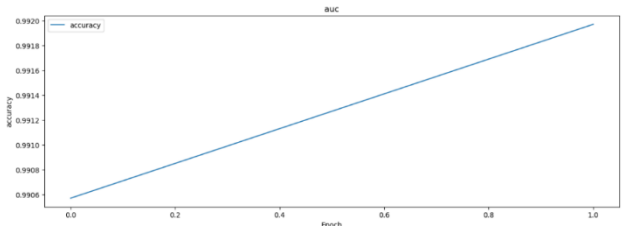

Figure 12. The proposed system train carve.

The training curve in Figure (12) shows a rapid and consistent decline in the loss value across each training batch, demonstrating that the model is learning effectively without signs of overfitting, a finding also confirmed by the high test results.

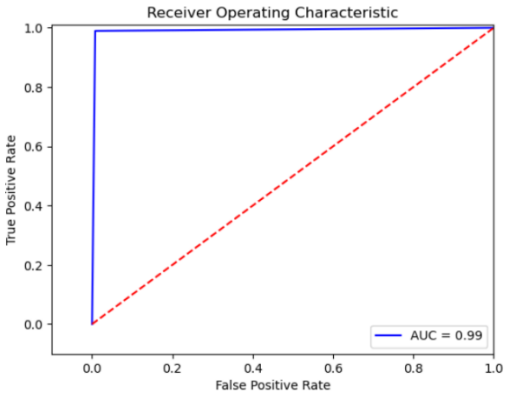

Figure 13. The ROC for proposed system

Figure (13) displays the Receiver Operating Characteristic (ROC) curve, which measures the model's ability to discriminate between classes (TPR to FPR ratio). A high Area Under the Curve (AUC) indicates high classification accuracy and strong confidence in the model's decisions across various classification threshold.

	precision	recall	f1-score	support
0	0.99	0.99	0.99	280585
1	0.99	0.99	0.99	279899
accuracy			0.99	560484
macro avg	0.99	0.99	0.99	560484
weighted avg	0.99	0.99	0.99	560484

Figure 14. The confusion matrix.

The confusion matrix in Figure (14) highlights the number of cases that were correctly or incorrectly classified. The distribution shows that the percentage of errors (False Positives and False Negatives) is very low compared to the percentage of correct classifications (True Positives and True Negatives), enhancing the model's reliability in security-sensitive scenario.

The performance metrics of the proposed model were evaluated, for the classification task. Experimental results of the, proposed model showed an F1 score of 99.08, a recall of,98.96, and a precision of 99.2, indicating strong, classification performance. Overall, this study contributes to enhancing IoT security through LSTM-based methods and lays the foundation for further exploration in this research. These results demonstrate the effectiveness of the LSTM model in processing and analyzing complex and noisy IoT traffic data. The model's high performance demonstrates its suitability for early detection of attacks and reducing the likelihood of human or technical error in intrusion detection systems. Compared to traditional algorithms, the proposed model demonstrates clear superiority, especially in handling time series data and understanding contextual relationships between features. Although the number of training cycles used was small (2 epochs), the results indicate that the model's architecture, preprocessing techniques, and class balance all contributed to accelerating the learning process and achieving accurate results in a short time. However, increasing the number of cycles in the future could further improve performance and allow for further improvements in model stability.

6. Conclusion

LSTM approach is employed within the realm of IoT security, leveraging its ability to capture long-term dependencies in sequential data. The analysis is conducted using the CIC-IoT2023 dataset, purposefully curated for IoT security analytics. The LSTM model, meticulously configured for optimal performance, exhibits exceptional accuracy, with an F1 score of 99.08, recall of 0.98.96, and precision of 99.2, highlighting its reliability in categorizing IoT security threats. The study aims to underscore the significance of LSTM models in enhancing IoT security while emphasizing the need for further research to enhance their interpretability, scalability, and efficiency, particularly in large-scale IoT deployments. Future research directions include expanding the application scope of LSTM models, optimizing their performance through techniques like model compression and hardware acceleration, and fostering interdisciplinary collaboration to tackle IoT security challenges effectively. Overall, this study contributes to advancing IoT security using LSTM-based methods and sets the stage for further exploration in this domain.

References

[1] W. Kassab and K. A. Darabkh, "A–Z survey of Internet of Things: Architectures, protocols, applications, recent advances, future directions and recommendations," *J. Netw. Comput. Appl.*, vol. 163, p. 102663, Aug. 2020, doi: 10.1016/j.jnca.2020.102663.

[2] M. Alaa, A. A. Zaidan, B. B. Zaidan, M. Talal, and M. L. M. Kiah, "A review of smart home applications based on Internet of Things," *J. Netw. Comput. Appl.*, vol. 97, pp. 48–65, Nov. 2017, doi: 10.1016/j.jnca.2017.08.017.

[3] P. Guo, K. Xiao, X. Wang, and D. Li, "Multi-source heterogeneous data access management framework and key technologies for electric power Internet of Things," *Global Energy Interconnection*, 2024, doi: 10.1016/j.gloei.2024.01.009.

[4] X. Liang and Y. Kim, "A survey on security attacks and solutions in the IoT network," in *Proc. IEEE 11th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2021, pp. 853–859, doi: 10.1109/CCWC51732.2021.9376174.

[5] M. A. Ferrag, L. Maglaras, A. Argyriou, D. Kosmanos, and H. Janicke, "Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes," *J. Netw. Comput. Appl.*, vol. 101, pp. 55–82, Jan. 2018, doi: 10.1016/j.jnca.2017.10.017.

[6] X. Yang *et al.*, "Towards a low-cost remote memory attestation for the smart grid," *Sensors*, vol. 15, no. 8, pp. 20799–20824, Aug. 2015, doi: 10.3390/s150820799.

[7] J. Lin, W. Yu, and X. Yang, "Towards multistep electricity prices in smart grid electricity markets," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 286–302, Jan. 2016, doi: 10.1109/TPDS.2015.2388479.

[8] G. Gomez, F. J. Lopez-Martinez, D. Morales-Jimenez, and M. R. McKay, "On the equivalence between interference and eavesdropping in wireless communications," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5935–5940, Dec. 2015, doi: 10.1109/TVT.2014.2387475.

[9] N. Zhao, F. R. Yu, M. Li, and V. C. M. Leung, "Anti-eavesdropping schemes for interference alignment (IA)-based wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 8, pp. 5719–5732, Aug. 2016, doi: 10.1109/TWC.2016.2568188.

[10] K. Zhang, X. Liang, R. Lu, and X. Shen, "Sybil attacks and their defenses in the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 372–383, Oct. 2014, doi: 10.1109/JIOT.2014.2344013.

[11] K. N. Qureshi, S. S. Rana, A. Ahmed, and G. Jeon, "A novel and secure attacks detection framework for smart cities industrial Internet of Things," *Sustain. Cities Soc.*, vol. 61, p. 102343, Oct. 2020, doi: 10.1016/j.scs.2020.102343.

[12] A. Mayzaud, R. Badonnel, and I. Chrisment, "A taxonomy of attacks in RPL-based Internet of Things," *Int. J. Netw. Secur.*, vol. 18, no. 3, pp. 459–473, 2016, doi: 10.13140/RG.2.2.23255.62885.

[13] I. Butun, P. Osterberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 616–644, Jan. 2020, doi: 10.1109/COMST.2019.2953364.

[14] M. El-Hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A survey of Internet of Things (IoT) authentication schemes," *Sensors*, vol. 19, no. 5, art. no. 1141, Mar. 2019, doi: 10.3390/s19051141.

[15] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2483–2495, Aug. 2018, doi: 10.1109/JIOT.2017.2767291.

[16] ITU-T, "Y.2066: Common requirements of the Internet of Things," *ITU-T Recommendation Y.2066*, Jun. 22, 2014, doi: 11.1002/1000/12169.

[17] J. Asharf *et al.*, "A review of intrusion detection systems using machine and deep learning in Internet of Things: Challenges, solutions and future directions," *Electronics*, vol. 9, no. 7, art. no. 1177, Jul. 2020, doi: 10.3390/electronics9071177.

[18] ITU-T, "Y.4101: Common requirements and capabilities of a gateway for Internet of Things applications," *ITU-T Recommendation Y.4101/Y.2067*, Oct. 29, 2017, doi: 11.1002/1000/13384.

[19] ENISA, "Baseline security recommendations for Internet of Things in the context of critical information infrastructures," *ENISA Rep.*, Nov. 2017. [Online]. Available: https://data.europa.eu/doi/10.2824/03228

[20] A. Odeh and A. Abu Taleb, "Ensemble-based deep learning models for enhancing IoT intrusion detection," *Appl. Sci.*, vol. 13, no. 21, art. no. 11985, Nov. 2023, doi: 10.3390/app132111985.

[21] S. Latif *et al.*, "Intrusion detection framework for the Internet of Things using a dense random neural network," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6435–6444, Sep. 2022, doi: 10.1109/TII.2021.3130248.

[22] A. Li and S. Yi, "Intelligent intrusion detection method of industrial Internet of Things based on CNN-BiLSTM," *Secur. Commun. Netw.*, vol. 2022, art. no. 5448647, 2022, doi: 10.1155/2022/5448647.

[23] S. Hanif, T. Ilyas, and M. Zeeshan, "Intrusion detection in IoT using artificial neural networks on UNSW-15 dataset," in *Proc. IEEE 16th Int. Conf. Smart Cities: Improving Quality Life Using ICT, IoT and A.I. (HONET-ICT)*, Charlotte, NC, USA, Oct. 2019, pp. 152–156, doi: 10.1109/HONET.2019.8908122.

[24] N. Thereza and K. Ramli, "Development of intrusion detection models for IoT networks utilizing CICIoT2023 dataset," in *Proc. 3rd Int. Conf. Smart Cities, Autom. Intell. Comput. Syst. (ICON-SONICS)*, Bali, Indonesia, 2023, pp. 66–72, doi: 10.1109/ICON-SONICS59898.2023.10435006.

[25] V. Ravi, R. Chaganti, and M. Alazab, "Deep learning feature fusion approach for an intrusion detection system in SDN-based IoT networks," *IEEE Internet Things Mag.*, vol. 5, no. 2, pp. 24–29, 2022, doi: 10.1109/IOTM.003.2200001.

[26] Z. Wang *et al.*, "A lightweight intrusion detection method for IoT based on deep learning and dynamic quantization," *PeerJ Comput. Sci.*, vol. 9, art. no. e1569, 2023, doi: 10.7717/peerj-cs.1569.

[27] R. Chaganti, V. Ravi, and T. D. Pham, "Deep learning based cross architecture Internet of Things malware detection and classification," *Comput. Secur.*, vol. 120, art. no. 102779, 2022, doi: 10.1016/j.cose.2022.102779.

[28] E. C. P. Neto *et al.*, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, art. no. 5941, Jul. 2023, doi: 10.3390/s23135941.

[29] B. Nemade, V. Bharadi, S. S. Alegavi, and B. Marakarkandy, "A comprehensive review: SMOTE-based oversampling methods for imbalanced classification techniques, evaluation, and result comparisons," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 9S, pp. 790–803, Jul. 2023. [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/3268

[30] L. Huang *et al.*, "Normalization techniques in training DNNs: Methodology, analysis and application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 10173–10196, Aug. 2023, doi: 10.1109/TPAMI.2023.3250241.

[31] R. Alkhatib, W. Sahwan, A. Alkhatieb, and B. Schütt, "A brief review of machine learning algorithms in forest fires science," *Appl. Sci.*, vol. 13, no. 14, art. no. 8275, Jul. 2023, doi: 10.3390/app13148275.

[32] A. I. Jony and A. K. B. Arnob, "A long short-term memory based approach for detecting cyber attacks in IoT using CIC-IoT2023 dataset," *J. Edge Comput.*, 2024, doi: 10.55056/jec.648.

How to cite this article

N. T. Mahmood, S. H. Jihad, S. M. Khaleel, and A. S. Abbas, "Leveraging deep Learning for Efficient Intrusion Detection in IoT Networks," CyberSystem Journal, vol. 2, no. 1, pp. 53–64, 2025.doi: 10.57238/csj.2025.1006

Access this article online