Journal of Edge Computing, **2024**, 3(1), pp. 28-42 https://doi.org/10.55056/jec.648

A long short-term memory based approach for detecting cyber attacks in IoT using CIC-IoT2023 dataset

Akinul Islam Jony, *Arjun Kumar Bose* Arnob

American International University-Bangladesh, 408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh

Abstract. The growth of Internet of Things (IoT) gadgets has ushered in a new era of connectedness and convenience, but it has also sparked worries about security flaws. Long Short-Term Memory (LSTM) networks are used in this research's use of intrusion detection as a novel strategy to strengthen IoT security. The proposed LSTM-based model excels in detecting both known and evolving cyber-attack patterns with an accuracy rate of 98.75% and an F1 score of 98.59% in extensive experimental evaluations using the vast CIC-IoT2023 dataset, representing a varied array of IoT network traffic scenarios. This research contributes significantly to IoT security while addressing the urgent need for adaptable intrusion detection systems to defend against changing cyber threats. It is an important step toward ensuring IoT technology's long-term development and dependability in a world that is becoming more interconnected.

Keywords: IoT, cyber-attacks, cybersecurity, threats, LSTM, CIC-IoT2023 dataset

1. Introduction

Since the last two centuries, the growth rate in human development has been accelerating due to the use of various technologies. The exponentially growing power of computers is one of the most promising technologies. The number of connections and networks with various devices like desktops, laptops, smartphones, PDAs, etc., is increasing and will significantly expand. Besides, it is expanding its reach as more linked devices are spread across cities to create smarter systems. These can reduce the need for human work and make human life more intelligent [19]. The IoT is the term used to describe how different entities of an object will communicate with one another [4]. IoT has experienced remarkable expansion, necessitating appropriate security and privacy rules to guard against system vulnerabilities or threats. IoT devices are vulnerable to cyberattacks because they are numerous, heterogeneous, require limited computing power, and usually operate on the edge of computer networks [2]. According to Pajouh et al. [14], IoT devices transmit data over wireless media, so they are a simpler target for attacks. Also, IoT device connectivity opens new access ports for hackers, offering security and privacy issues.

Attacks on IoT systems spread across a much wider region and have disastrous repercussions on IoT sites, in contrast to regular transmission attacks on local networks, which are restricted to nodes near a small local domain. According to Illing [10], the attacks on the IoT have

akinul@aiub.edu (A. I. Jony); arjunkumarbosu@gmail.com (A. K. B. Arnob)
https://cs.aiub.edu/profile/akinul (A. I. Jony); https://github.com/arjunkumarbose (A. K. B. Arnob)
0000-0002-2942-6780 (A. I. Jony); 0009-0003-2244-2328 (A. K. B. Arnob)

greatly grown and sophisticated. A secure IoT network is necessary for combating cybercrime. Furthermore, a few other crucial issues in the IIoT (Industrial Internet of Things) include stability, scalability, and power consumption. Conventional security measures are not always appropriate in this situation. The security measures that are currently in place are made vulnerable by IoT devices. IoT cyberattacks typically take the form of brute force attacks, physical interference, cloud-related problems, botnets, and man-in-the middle attacks. Due to the possibility of targeted assaults using personal data from linked devices, data and identity theft are potentially serious issues. Therefore, the primary goal of this study is to concentrate on the numerous security assaults that may be classified according to the IoT-related objects of attack. A handful of these assaults are equally crucial in an IIoT setting. Researchers, practitioners, and businesspeople in the industry would be able to determine which assaults are pertinent to their application domain with this object-based classification of attacks [19]. These cutting-edge electronic devices have made enormous strides, but because they have recently been the subject of cyberattacks, they have also brought forth brand-new issues.

On the other hand, deep learning (DL) approaches are useful for identifying DDoS attacks because of their ability to classify the data and extract features from datasets. A system for detection that can cope with data unavailability is required in the modern environment. Labels for legitimate traffic are commonly available, whereas labels for fraudulent traffic are less frequent [21]. Long Short-Term Memory Networks (LSTM)- simply explained! [12], What are Recurrent Neural Networks? [22] mentioned that recurrent neural networks (RNNs) have a variant called LSTM models used to find patterns in data sequences. To create predictions, RNNs can use the context and location of a sequence. Regular RNNs, on the other hand, only have short-term memory and have trouble with longer sequences. LSTM models address this problem by including both long-term and short-term memory. According to Dolphin [5], LSTM networks were created to address the recurrent neural networks' long-term reliance issue. LSTMs have feedback connections, which, in contrast to conventional feedforward neural networks, enable them to handle data sequences while preserving knowledge about prior data points. This makes them especially useful for processing time series, text, and speech data. The ability of LSTM networks to efficiently capture long-term relationships in the data is the first advantage [5]. This capability is essential for identifying complicated assault patterns that may include several time steps. In IoT contexts, where assaults can be subtle and develop over time, this is extremely crucial. LSTM networks are flexible enough to accommodate the dynamic nature of IoT data since they can accept variable-length sequences [1]. This makes it possible to detect assaults regardless of how long the data sequence is being analyzed. Therefore, an LSTM-based model has been proposed in this research study to identify cyber-attacks/threats in IoT and create the model CIC-IoT2023. The dataset is used as a sample.

The rest of the article is structured as follows: section 2 mentions some existing works related to this study, section 3 describes the methods and materials used in this study, section 4 outlines the results and findings of this research work, and finally, section 5 concludes the article with some concluding remarks.

2. Related works

The central emphasis of this research is on IoT security, so it is acknowledged that broader cybersecurity strategies may provide insightful perspectives and possible relevance to IoT environments. For example, methodologies such as anomaly detection and network behaviour analysis, which have been thoroughly investigated in conventional cybersecurity fields, could be modified to bolster security frameworks for the IoT. The rationale for focusing on literature specific to the IoT arises from IoT devices' distinctive attributes and limitations, including restricted computational capabilities and a wide range of communication protocols. These factors require customized security measures. Nevertheless, forthcoming versions of this study will endeavour to narrow down this disparity by integrating a comparative examination of conventional cybersecurity techniques and their applicability to the IoT environments, utilizing seminal contributions from the discipline [1, 2]. Adopting a holistic approach will guarantee a thorough comprehension of the cybersecurity environment and implications for the perpetually changing IoT ecosystem.

De La Torre Parra et al. [3] claim that DDoS-based flooding attacks utilize ICMP, DNS protocol packets, TCP, and UDP to sever communications from registered users at the network/transport level. DDoS flooding attacks target the application layer of the target server and seek to exhaust its storage capacity, disk/database, I/O bandwidth, and ports [18]. Cyber attackers frequently target IoT devices with limited resources because they are easy targets. Additionally, malevolent IoT objects can take part in more significant attacks. Hence, according to Sahu et al. [18], the applied CNN approach based on the dataset consists of CC, FileDownload, HeartBeat, PartofHorizontalPortScan, Torii, Okiru, Mirai, DDoS, and Benign, and the suggested model has a 96% F-Measure for detecting malicious network traffic. Once identified, The traffic can be further analyzed and identified using the CNN-based sub-classification network.

According to Al-Garadi et al. [1], monitoring IoT devices can effectively provide a defence against new or zero-day threats (as demonstrated in figure 1). DL and ML (Machine Learning) are effective techniques for data exploration to comprehend "normal" and "abnormal" behaviours concerning how IoT gadgets and parts interact. Each IoT system component's input data can be collected and evaluated to identify common patterns of interaction, enabling the early identification of malicious conduct. Additionally, because ML/DL approaches may learn from prior examples to intelligently forecast impending unknown attacks, they may be essential for predicting upcoming assaults, typically variants of earlier attempts. To be effective and secure, IoT systems must advance beyond enabling safe interaction among gadgets to security-based expertise supported by DL/ML approaches.

Bi-directional LSTM, a variation on the RNN model, was presented as an IoT attack detection mechanism by Roy and Cheung [17]. The model offers 95% accuracy. However, the model is trained using a single dataset and 5451 test samples. In addition, it does not provide any comparisons to other modern models. RNNs have been further investigated by HaddadPajouh et al. [7] to detect IoT malware. On 32-bit ARM-based processors, they gathered malware samples. The OpCodes were used to generate their dataset, and three different iterations of the LSTM model were examined [6, 8]. They trained their model with 281 malicious and 270 benign programs, which had a 98% accuracy rate. The dataset is small and mimicked, as can be seen, and hence, the model needs to be evaluated using the large new datasets that are

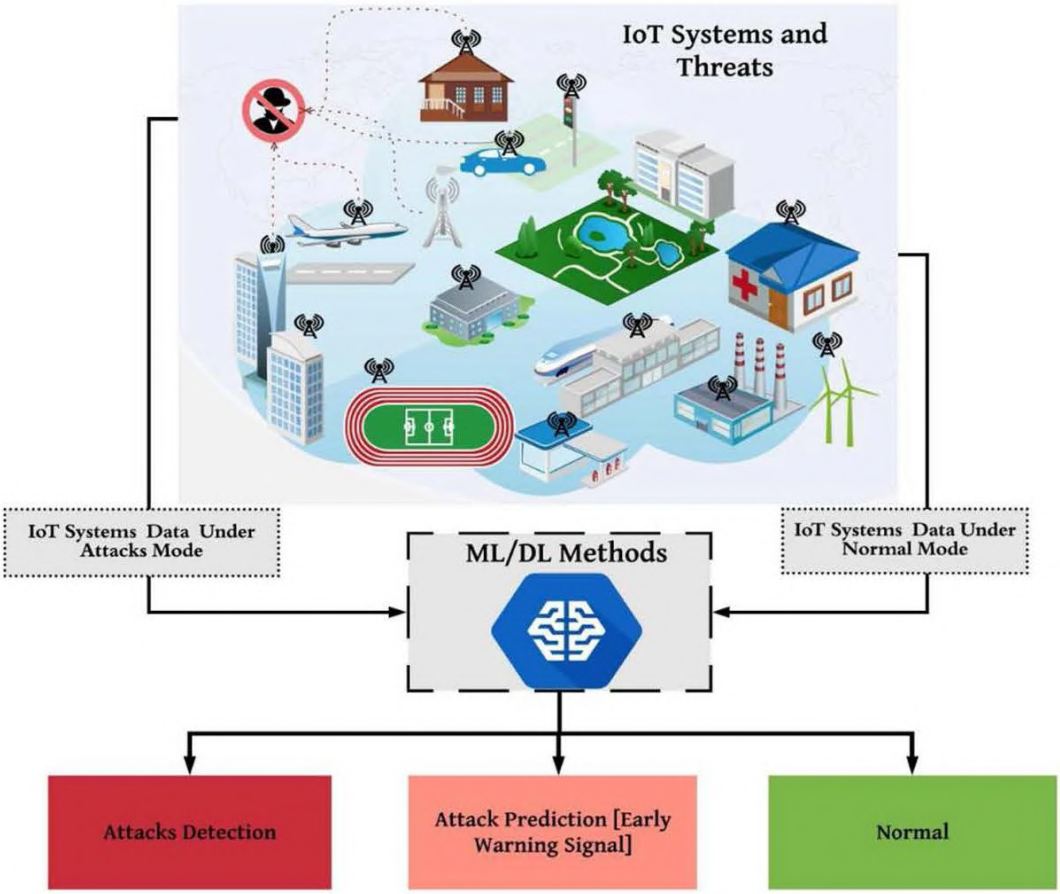

Figure 1: Illustration of the potential role of ML/DL in IoT security [1].

accessible. Tran et al. [20] trained and tested the proposed DL method to create an effective DL model using a real-time dataset taken by an intelligent CNC machine in different cutting process scenarios. A bogus dataset is randomly added to the real-time dataset of the smart CNC machine to represent a cyber-attack, and it was discovered that the linear SVM classifier, which has an accuracy rate for classification of 93.33%, makes the accuracy possible. By using nearest neighbours using a count of three, the KNN structure increases the precision of classification to 98.3%. Incorporating a signal-hidden layer of 10 neurons into the ANN model can raise the classification accuracy to 98.6%. On the other hand, the enhanced proposed deep neural network (DNN) performs superiorly to other conventional machine learning approaches in classifying various milling process states with 99.47% accuracy. This demonstrates how the deep learning network can automatically extract representative features from the dataset after learning the pattern. The best features inside the dataset had to be designed and chosen using deep expertise and signal processing skills for classical feature learning.

Ibitoye, Shafiq and Matrawy [9] utilize the BoT-IoT dataset given by the Cyber Range Lab

of the UNSW Canberra Cyber Center. A scaled-down version of roughly 3.6 million records from the dataset, which includes over 72 million records of network activity in a simulated IoT environment, was used for the study [10], and the researchers evaluated the effect of adversarial samples on an Intrusion Detection System (IDS) built on deep learning in IoT networks. The IDS initially had a high accuracy of 95.1%. However, its accuracy drastically decreased to 24%, 18%, and 31%, respectively, when tested with adversarial samples made using FGSM, BIM, and PGD approaches. They then compared the performance of two IDS models, the FNN (Feedforward Neural Network) and the SNN (Self-normalizing Neural Network). Across a range of performance criteria, such as precision, recall, F1-score, and multiclassification metrics like Copen Cappa Score and MC Coefficient, the FNN IDS consistently beat the SNN IDS.

3. Methods and materials

The methods and materials used in this study are briefly discussed in this section. The model architecture is defined, and a set of evaluation criteria for evaluating the model is also presented. Furthermore, a brief introduction of the dataset used in this study is given.

3.1. Model architecture and evaluation metrics

This subsection discusses the deep learning model's structure and the suggested model's mathematical foundations.

According to Ingolfsson [11], the LSTM's structure has evolved through time, and the most prevalent architecture will be described here. An LSTM unit comprises a cell, and three gates within the cell regulate the information flow and state of the LSTM cell. An input gate, an output gate, and a forget gate comprise the trio of gates. Then, the LSTM chains these cells together, each LSTM cell acting as a memory module.

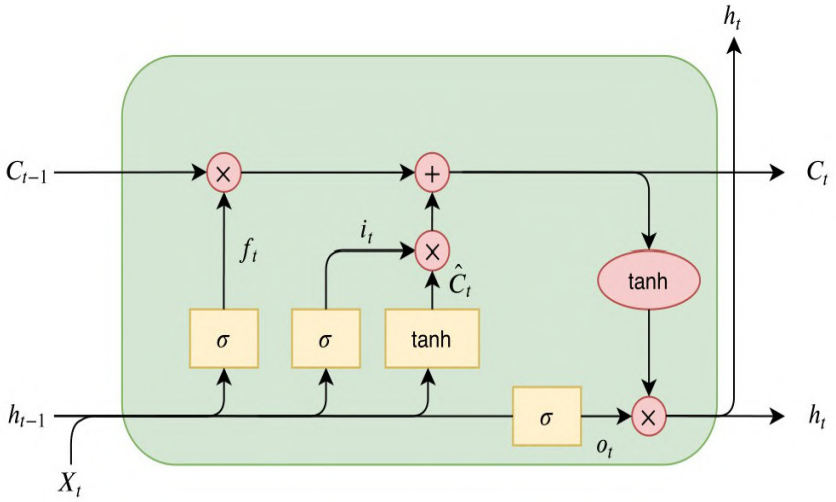

Figure 2: LSTM cell architecture [11].

Here's the LSTM cell architecture. It stands for input time step, X_t for output, h_t for output, C_t for cell state, f_t for forget gate, i_t for input, o_t for output, and t for internal cell state. Operations are pointwise inside the light red circle. In figure 2, the three gates—forget, input, and output—are denoted by the letters f_t, i_t, and O_t, respectively. The gates are based on straightforward intuition:

- The forget gate instructs the cell which information from its internal state to "forget" or discard.
- The cell is instructed by the input gate, which new information should be stored in the internal cell state.
- The cell then emits what is known as the output gate, a filtered representation of the internal state of the cell.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, X_t] + b_f\right) \tag{1}$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, X_t] + b_i\right) \tag{2}$$

$$O_t = \sigma\left(W_O \cdot [h_{t-1}, X_t] + b_O\right) \tag{3}$$

$$\hat{C}_t = \tanh\left(W_C \cdot [h_{t-1}, X_t] + b_C\right) \tag{4}$$

Then, the internal cell state is computed as

$$C_t = i_t \cdot \hat{C}_t + f_t \cdot C_{t-1} \tag{5}$$

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

$$h_t = o_t \cdot \tanh(C_t) \tag{6}$$

Weights and biases are coupled to each gate, just like neural networks. To enable the LSTM cell to learn, these weight matrices are combined with gradient-based optimization. In the equations above, weight matrices and biases are denoted by W_f, b_f, W_i, b_i, W_o, b_o, and W_C, b_c, respectively. The RNN/LSTM network may preserve data from earlier time steps and create time-series predictions thanks to the chaining of these cells, as seen in figure 3. The network can solve the vanishing gradient issue using the LSTM cell topology. Older RNN designs were unable to make excellent time-series predictions because of this.

The following valuation metrics are used for assessing the model of this study for detecting cyber-attacks in IoT using the CIC-IoT2023 Dataset this study.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1 - Score = \frac{Precision + Recall}{2} \tag{10}$$

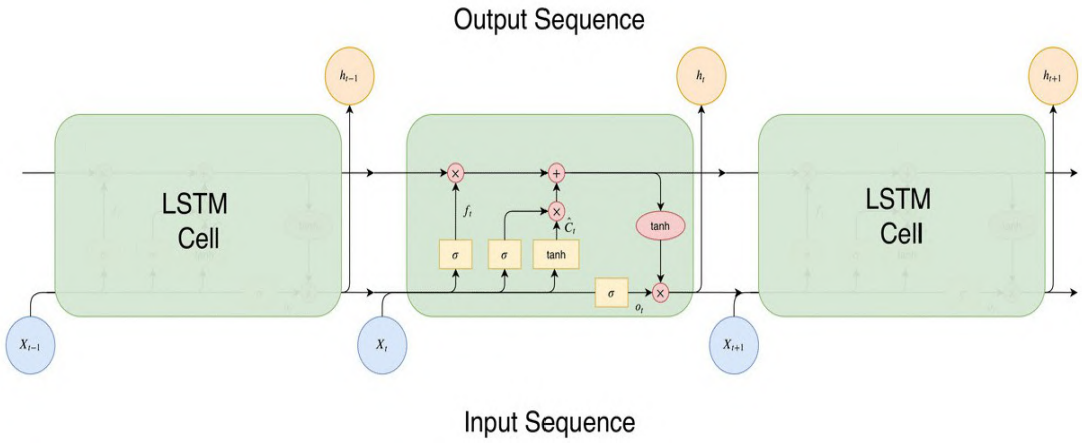

Figure 3: Standard LSTM model [11].

Rectified Linear Unit, or ReLU, is a popular activation function in neural networks. It generates nonlinearity by producing the input value when it is positive or zero and 0 otherwise. ReLU is a common option for DL models since it is computationally effective and helps solve the vanishing gradient issue [16]. Deep networks with ReLUs can be optimized more quickly than deep networks with sigmoid or tanh units because gradients can flow when the input to the ReLU function is positive. ReLU has emerged as the deep learning community's go-to activation function due to its ease of use and potency. The LSTM layer is activated using the "relu" activation function. For multi-class classification problems, softmax is a common activation function used in the output layer of neural networks. It converts a vector of real numbers into a probability distribution, ensuring that the output vector's values add up to 1. Each component of the output vector denotes the estimated likelihood that the input belongs to a particular class. Softmax is essential for allocating probability to several classes and identifying the predicted class in multi-class classification issues. Qin, Kim and Gedeon [15] mentioned that it is common practice to train neural network classifiers using the softmax and cross-entropy combination. It calculates the cross-entropy between the neural network's output and the ground truth label y. Then, the network's parameters are changed using backpropagation to lessen the cross-entropy. The network's goal in modelling the relationship between input x and label y via this loss function, i.e., softmax with cross-entropy, is still unclear, even if it makes sense to lower the cross-entropy between labels and projected probabilities. In the output layer of the LSTM model, the "softmax" activation function has been used.

This study uses a batch training method with a batch size of 1000 to train and test the LSTM model over a sequence of 50 epochs. During the training process, batches of training data made up of input features and associated target labels are fed to the model to update its parameters repeatedly. A progress bar shows the current epoch and tracks the training loss using the 'train_on_batch' method during each epoch to visualize the training progress. After that, each epoch is followed by an evaluation phase, during which the evaluation dataset is partitioned into batches, and the model's performance is evaluated using the 'test_on_batch' method. This method guarantees effective computing and offers insightful information about the dynamics of

the model's learning. The trained LSTM model was reviewed thoroughly during the research's evaluation phase to determine how well it performed at classifying data. It is possible to understand the model's capabilities comprehensively by using various common evaluation criteria, such as accuracy, F1 score, recall, and precision. The test dataset's predictions from the model produced class probabilities that were stored in the variable y_pred. Then, these probabilities are divided into predicted classes, represented by y_pred_classes, by choosing the class with the highest likelihood for each data point. In parallel, the test dataset's true class labels were retrieved and saved in the variable y_true.

3.2. Dataset overview

The dataset used in this study is a very recent dataset called the CIC IoT Dataset 2023, which was created to support the creation of security analytics software for the IoT environment [13]. It consists of 33 assaults carried out over 105 IoT devices, divided into seven categories: DDoS, Denial of Service (DoS), Recon, Web-based, Brute Force, Spoofing, and Mirai. These assaults are carried out by IoT devices that are maliciously aimed at other IoT devices.

Figure 4 demonstrates a wide variety of IoT cyberattacks, along with a corresponding number of rows in the dataset, pose a threat to the availability and integrity of computer systems and networks in cybersecurity. A variety of strategies, including flooding assaults like UDP and ICMP Floods and fragmentation-based attacks, are included in DDoS attacks. DoS attacks cause service disruptions by flooding a single source with traffic. Web-based attacks use techniques like SQL Injection and XSS to target web applications. Through numerous tests, brute force attacks try to acquire unauthorized access. Attacks that spoof entities or change network traffic are known as spoofing. Finally, Mirai attacks use strategies like GREIP Flood and UDPPlain attacks, which mostly target IoT devices. It also presents a thorough overview of network assaults, their corresponding frequencies shown by the number of rows, and their classification into more general attack types. To gauge the gravity of these threats, the row counts represent the frequency of each distinct assault type within the dataset.

A thorough and cautiously arranged set of features from network traffic data is presented in table 1, providing a detailed look at the traits and behaviours of packets inside a network. The "Timestamp" gives each packet's specific recording time. "Flow Duration" provides information on how long a packet has been flowing, and "Protocol Type" classifies packets according to their network protocols, which include well-known ones like IP, UDP, and TCP. Indicators for other application layer protocols, including "HTTP", "HTTPS", "DNS" and others, are also included, making it possible to spot particular application-level behaviour in network traffic. Data throughput is provided, as well as the packet transmission rate under "Rate". The counts of several flags, including "FIN", "SYN", "RST", "PSH", "ACK", "ECE", and "CWR", provide information on particular packet-level interactions and any anomalies. To understand how the lengths of incoming and outgoing packets connect, statistical metrics like "Covariance" and "Variance Ratio" evaluate the variability in packet lengths. "Weight" measures the sum of the incoming and departing packet counts, providing a comprehensive picture of traffic patterns. Network specialists can learn important details about the network's performance and security posture thanks to additional elements like "Magnitude", "Radius", "Standard Deviation", "Packet Length", "Inter-Arrival Time", and "Packet Count" that provide depth to the research.

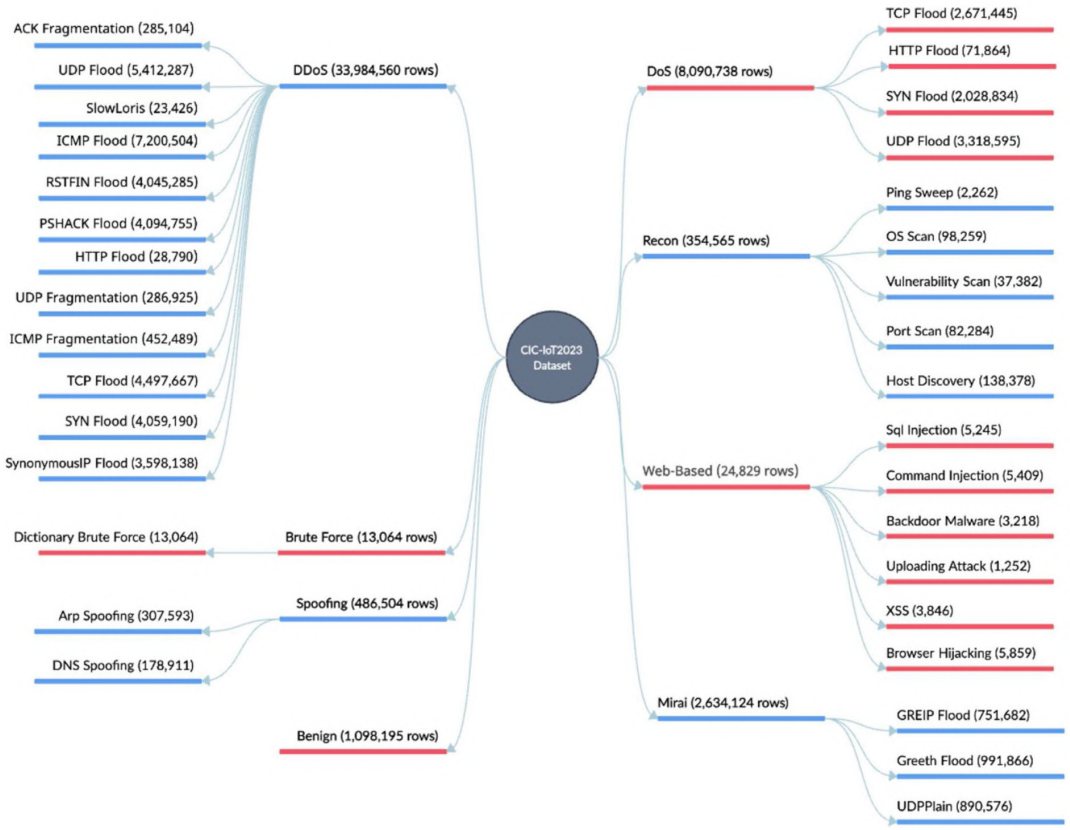

Figure 4: Dataset overview.

For network analysts and security professionals, this set of features is a valuable resource for network traffic analysis, anomaly identification, and optimization tasks.

Table 1
Features extracted from the network traffic [13].

Feature	Description
Timestamp	Time when the packet was recorded
Flow Duration	Duration of the packet's flow
Header Length	Length of the packet's header
Protocol Type	Type of protocol (e.g., IP, UDP, TCP)
Time-to-Live (TTL)	Time-to-Live value of the packet
HTTP	Indicates if the application layer protocol is HTTP
HTTPS	Indicates if the application layer protocol is HTTPS
DNS	Indicates if the application layer protocol is DNS
Telnet	Indicates if the application layer protocol is Telnet

Continued on next page

Table 1 – continued from previous page

Feature	Description
SMTP	Indicates if the application layer protocol is SMTP
SSH	Indicates if the application layer protocol is SSH
IRC	Indicates if the application layer protocol is IRC
TCP	Indicates if the transport layer protocol is TCP
UDP	Indicates if the transport layer protocol is UDP
DHCP	Indicates if the application layer protocol is DHCP
ARP	Indicates if the link layer protocol is ARP
ICMP	Indicates if the network layer protocol is ICMP
IP	Indicates if the network layer protocol is IP
LLC	Indicates if the link layer protocol is LLC
Total Packet Length	Summation of packet lengths in the flow
Min Packet Length	Minimum packet length in the flow
Max Packet Length	Maximum packet length in the flow
Average Packet Length	Average packet length in the flow
Packet Count	Number of packets in the flow
Rate	Packet transmission rate in the flow
Outbound Rate (Srate)	Outbound packet transmission rate in the flow
Inbound Rate (Drate)	Inbound packet transmission rate in the flow
FIN Flag Count	Count of packets with FIN flag set in the flow
SYN Flag Count	Count of packets with SYN flag set in the flow
RST Flag Count	Count of packets with RST flag set in the flow
PSH Flag Count	Count of packets with PSH flag set in the flow
ACK Flag Count	Count of packets with ACK flag set in the flow
ECE Flag Count	Count of packets with ECE flag set in the flow
CWR Flag Count	Count of packets with CWR flag set in the flow
CK Packet Count	Number of packets with ACK flag set in the flow
SYN Packet Count	Number of packets with SYN flag set in the flow
FIN Packet Count	Number of packets with FIN flag set in the flow
URG Packet Count	Number of packets with URG flag set in the flow
RST Packet Count	Number of packets with RST flag set in the flow
Covariance	Covariance of the lengths of incoming and outgoing packets
Variance Ratio	Variance of the lengths of incoming packets divided by variance of the lengths of outgoing packets
Weight	Number of incoming packets multiplied by the number of outgoing packets
Magnitude	Average of the lengths of incoming and outgoing packets in the flow
Radius	Variance of the lengths of incoming and outgoing packets in the flow
Standard Deviation	Standard deviation of packet length in the flow
Packet Length	Length of the packet
Inter-Arrival Time	The time difference with the previous packet

4. Results and findings

The datasets were arranged alphabetically and divided into training and test sets, with the top 70% of the datasets serving as training sets to allow for reliable experimentation and evaluation of the LSTM model's effectiveness on various data subsets. The research established a thorough set of column names for feature extraction and labelling inside the datasets. These columns included information about many aspects of network traffic, such as flow time, protocol type, flag counts, and details about protocols like HTTP, HTTPS, DNS, etc. The 'label' column was set aside for class labels to make supervised learning tasks easier. During the data preprocessing phase, several crucial actions were carried out to prepare the dataset for the LSTM model's training and testing. To enable subsequent processing, the raw data, which consisted of sequences and the labels that went with them, was originally transformed into NumPy arrays. Then, standardization was used to ensure that the characteristics had a mean of 0 and a standard deviation of 1, improving the model's performance. After that, label encoding was used to transform categorical labels into numerical values so that machine learning methods could be used. The training and testing datasets were subsampled using a predefined data fraction, where a fraction of 0.1 indicated the consumption of 10% of the data for managing the computing needs and speeding up experimentation. These preprocessing methods cleared the way for efficient LSTM model training and evaluation by ensuring the data were formatted and scaled correctly for the best model performance.

A thorough investigation of the LSTM model's performance throughout 50 epochs throughout the training and evaluation phases has been performed. A steady decline is observed in the model's loss during training, with both the training and evaluation losses declining. This shows that the model successfully discovered the underlying relationships and patterns in the data. The weights and biases of the model were iteratively updated during the training process to reduce the loss function and finally adjust the model's parameters. Exceptional results are achieved in the LSTM model's final evaluation.

To evaluate the effectiveness of the LSTM-based intrusion detection model, a confusion matrix (as shown in figure 5) is used to thoroughly analyse the classification outcomes, enabling us to assess the model's precision and misclassifications. The genuine class labels are represented by each row and each column represents the anticipated class labels in the confusion matrix. The matrix's elements stand for how many occurrences fall into each category. While off-diagonal elements signify incorrect classifications (false positives and false negatives), diagonal elements represent accurate predictions (true positives and true negatives). The confusion matrix's visual depiction offers important insights into the model's classification accuracy and any potential improvement areas.

Table 2 shows the model's outstanding accuracy of 98.75%, indicating that it could accurately and precisely classify cases. The model must have this extraordinary accuracy to be dependable and useful in practical applications. Another impressive statistic that shows a well-balanced trade-off between recall and precision is the F1 score of 0.9859. This shows that the model successfully detects meaningful cases while reducing false positives and false negatives. When memory and precision are both crucial, a high F1 score is especially important. Further demonstrating the model's success in accurately recognizing positive events, the recall score of 0.9875 shows that it excels at capturing actual positive instances. Additionally, the model's accuracy

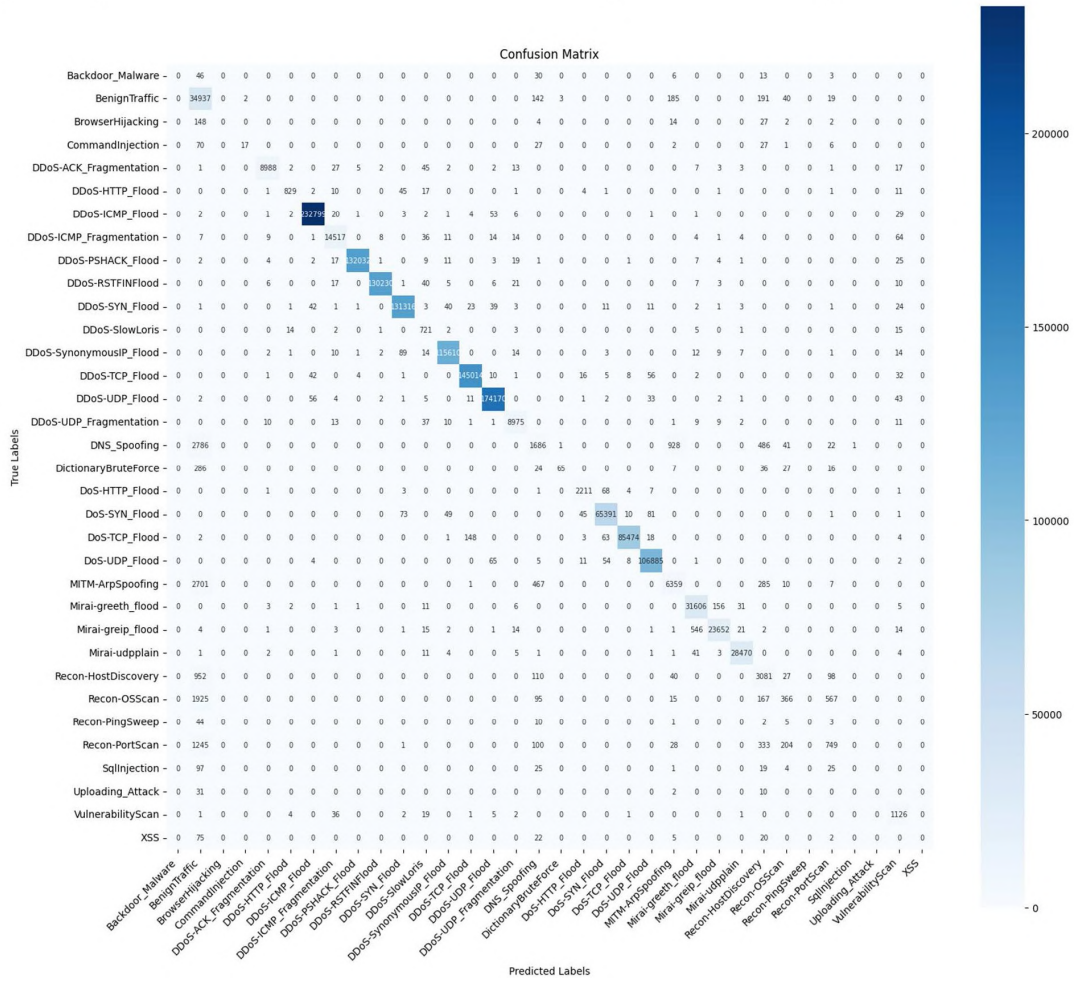

Figure 5: Confusion matrix based on true labels and predicted labels.

Table 2
Evaluation matrices of the proposed model.

Evaluation matrices	Output
Accuracy	0.9875
F1 Score	0.9859
Recall Score	0.9875
Precision Score	0.9866

in correctly classifying objects is shown by its precision score of 0.9866. These outstanding evaluation outcomes highlight the LSTM model's dependability and robustness for the given task. Such high-performance measures are invaluable in a wide range of applications, especially those where precision and dependability are critical, including, but not limited to, medical

diagnosis, fraud detection, and natural language processing. These results support the LSTM model's usefulness and effectiveness in the research environment.

5. Conclusion

The implementation of the LSTM approach in the context of IoT security has been thoroughly explored in this study. The results showed that the LSTM model performs remarkably well when given the right architecture and instruction. Notably, with an F1 score of 0.9859, recall of 0.9875, and precision of 0.9866, this model obtained a remarkable accuracy rate of 98.75%. These findings highlight the LSTM model's dependability and durability, making it a significant tool in situations requiring exact categorization, which is essential for IoT security. Additionally, it provided a thorough summary of the CIC-IoT2023 dataset, which served as the basis for the study and is a useful tool for additional research in this area.

Even though the model has performed exceptionally well, there is a need for more study to improve its understandability and interpretability, especially in situations where model assessments are needed for precise comprehension and validation. In addition, scalability and efficiency issues are crucial, particularly in cases involving extensive IoT deployments.

Future research should broaden the application and accessibility of LSTM models by including more data types and domains in their scope. Optimizing these models requires investigating methods like model compression, quantization, and hardware acceleration. Innovative applications and solutions can be produced by collaboration with domain specialists in healthcare, finance, and natural language processing. IoT security has significantly advanced, thanks to LSTM-based methods, and this study is a big step in that direction. It also opens up intriguing new directions for future research into improving model performance and real-world application.

6. Data availability statement

A public dataset is used in this study, which can be found at https://www.unb.ca/cic/datasets/iotdataset-2023.html [13].

7. Funding

This research received no funding.

References

[1] Al-Garadi, M.A., Mohamed, A., Al-Ali, A.K., Du, X., Ali, I. and Guizani, M., 2020. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Communications Surveys & Tutorials*, 22(3), pp.1646–1685. Available from: https://doi.org/10.1109/COMST.2020.2988293.

[2] Chesney, S., Roy, K. and Khorsandroo, S., 2021. Machine Learning Algorithms for Preventing IoT Cybersecurity Attacks. In: K. Arai, S. Kapoor and R. Bhatia, eds. *Intelligent Systems and Applications*. Cham: Springer International Publishing, pp.679–686. Available from: https://doi.org/10.1007/978-3-030-55190-2_53.

[3] De La Torre Parra, G., Rad, P., Choo, K.K.R. and Beebe, N., 2020. Detecting Internet of Things attacks using distributed deep learning. *Journal of Network and Computer Applications*, 163, p.102662. Available from: https://doi.org/https://doi.org/10.1016/j.jnca.2020.102662.

[4] Deogirikar, J. and Vidhate, A., 2017. Security attacks in IoT: A survey. *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. pp.32–37. Available from: https://doi.org/10.1109/I-SMAC.2017.8058363.

[5] Dolphin, R., 2021. LSTM Networks | A Detailed Explanation. Medium. Available from: https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9.

[6] Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R. and Schmidhuber, J., 2017. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), pp.2222–2232. Available from: https://doi.org/10.1109/TNNLS.2016.2582924.

[7] HaddadPajouh, H., Dehghantanha, A., Khayami, R. and Choo, K.K.R., 2018. A deep Recurrent Neural Network based approach for Internet of Things malware threat hunting. *Future Generation Computer Systems*, 85, pp.88–96. Available from: https://doi.org/10.1016/j.future.2018.03.007.

[8] Hochreiter, S. and Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation*, 9(8), pp.1735–1780. Available from: https://doi.org/10.1162/neco.1997.9.8.1735.

[9] Ibitoye, O., Shafiq, O. and Matrawy, A., 2019. Analyzing Adversarial Attacks against Deep Learning for Intrusion Detection in IoT Networks. *2019 IEEE Global Communications Conference (GLOBECOM)*. pp.1–6. Available from: https://doi.org/10.1109/GLOBECOM38437.2019.9014337.

[10] Illing, D., 2023. Common Cyber-Attacks in the IoT | GlobalSign. GlobalSign. Available from: https://www.globalsign.com/en/blog/common-cyber-attacks-in-the-iot.

[11] Ingolfsson, T.M., 2021. Insights into LSTM architecture. Available from: https://thorirmar.com/post/insight_into_lstm/.

[12] Long Short-Term Memory Networks (LSTM)- simply explained!, 2022. Available from: https://databasecamp.de/en/ml/lstms.

[13] Neto, E.C.P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R. and Ghorbani, A.A., 2023. CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. *Sensors*, 23(13). Available from: https://doi.org/10.3390/s23135941.

[14] Pajouh, H.H., Javidan, R., Khayami, R., Dehghantanha, A. and Choo, K.K.R., 2019. A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks. *IEEE Transactions on Emerging Topics in Computing*, 7(2), pp.314–323. Available from: https://doi.org/10.1109/TETC.2016.2633228.

[15] Qin, Z., Kim, D. and Gedeon, T., 2020. Rethinking Softmax with Cross-Entropy: Neural Network Classifier as Mutual Information Estimator. 1911.10688, Available from: https://doi.org/10.48550/arXiv.1911.10688.

[16] Ramachandran, P., Zoph, B. and Le, Q.V., 2017. Searching for Activation Functions. 1710.05941, Available from: https://doi.org/10.48550/arXiv.1710.05941.

[17] Roy, B. and Cheung, H., 2018. A Deep Learning Approach for Intrusion Detection in Internet

of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network. *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*. pp.1–6. Available from: https://doi.org/10.1109/ATNAC.2018.8615294.

[18] Sahu, A.K., Sharma, S., Tanveer, M. and Raja, R., 2021. Internet of Things attack detection using hybrid Deep Learning Model. *Computer Communications*, 176, pp.146–154. Available from: https://doi.org/10.1016/j.comcom.2021.05.024.

[19] Sengupta, J., Ruj, S. and Das Bit, S., 2020. A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT. *Journal of Network and Computer Applications*, 149, p.102481. Available from: https://doi.org/10.1016/j.jnca.2019.102481.

[20] Tran, M.Q., Elsisi, M., Liu, M.K., Vu, V.Q., Mahmoud, K., Darwish, M.M.F., Abdelaziz, A.Y. and Lehtonen, M., 2022. Reliable Deep Learning and IoT-Based Monitoring System for Secure Computer Numerical Control Machines Against Cyber-Attacks With Experimental Verification. *IEEE Access*, 10, pp.23186–23197. Available from: https://doi.org/10.1109/ACCESS.2022.3153471.

[21] Van, N.T., Thinh, T.N. and Sach, L.T., 2017. An anomaly-based network intrusion detection system using Deep learning. *2017 International Conference on System Science and Engineering (ICSSE)*. pp.210–214. Available from: https://doi.org/10.1109/ICSSE.2017.8030867.

[22] What are Recurrent Neural Networks?, 2021. Available from: https://databasecamp.de/en/ml/recurrent-neural-network.