International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(21s) 2026 pp. 1652-1669
https://doi.org/10.70917/ijcisim-2026-5917
Article

Deep Learning Model for IOT-Integrated WSN Network for Security Threat Classification

Tanuja Saini[1], Dr. Rajpreet Singh[2], Dr. Kritika Raj Sharma[3]

[1]Electronics & Communication Engineering, Chandigarh University,sainitanuja28@gmail.com
[2]Electronics & Communication Engineering, Chandigarh University,rajpreet.ece@cumail.in
[3]Electronics & Communication Engineering, Chandigarh University,kritika.ece@cumail.in

Abstract: Worldwide, industry has seen a radical transformation because of technological advances, i.e., real-time automation systems that employ sensors to gather physical parameter data and transmit that data to remote computers via wireless sensor networks. Even while such technology offers many advantages like higher operational effectiveness, real-time data visualization for better decision-making, cost reduction, etc., it's also subjected to significant data security threats. For this, conventional security methods like cryptography and authentication & access control are also challenging to implement due to the variety of connections and the constraints on compute, energy, and storage capability of industrial equipment. Therefore, in order to safeguard privacy, accessibility, and data integrity when collecting data, transferring, storing, etc., strong security measures must be implemented. Derived from this, the author has assessed the efficacy of different deep learning (DL) model to verify their validity in terms of how well it work when simulated using a digital threat dataset so that they can be proposed for real-time IoT-integrated WSN data security systems. It has been found that LSTM model for Wireless Sensor Network (WSN) attack categorization performs remarkably well, attaining high recall (98.68%), accuracy (98.68%), and precision (98.61%).
Keywords: Wireless sensor network, IoT, Security threats, Machine learning, Deep learning etc

1. Introduction

Applications for In today's world, wireless sensor networks (WSN) are expanding globally; some examples include manufacturing control systems, medical systems, and agricultural automation [1]. The number of electronic devices having Internet connectivity has increased at a never-before-seen rate worldwide as a result of these advancements in wireless network technology. According to the World Data Corporation, there will be 30 billion sensor-enabled objects connected to networks and between 50 billion and 1 trillion smart devices by 2020. These include over 0.5 billion sensor systems in US businesses, 1.2 million home automation systems with 200 million sensors, 0.110 billion integrated cars with 5.5 billion sensors, and 212 billion accessible sensors [2]. By 2030, There will be around 24.1 billion IoT devices worldwide or around four gadgets per person [3]. Due to the notable developments in wireless sensor technology, the issue of managing, handling and protecting data has grown significantly. In recent years, the business has seen a sharp rise in data breaches and hacking efforts. According to a 2021 report, only within a healthcare industry, breaches have increased by 55.1% from 2019 and 2020. In 2020 alone, there were almost 600 data breaches [4].

WSNs are often composed of many sensor nodes positioned in certain areas of a network. In this context, "IoT" refers to tangible objects that are linked to the internet using wireless sensors [5]. Possessing sensing and wireless communication capabilities, these devices are straightforward but naturally intelligent [6-7]. However, such networks have potential security threats that could jeopardize the effectiveness of these networks, like risks threat while data is transmitted [8]. As its well-known, wireless communication distances are easily insecure and unconstrained. Numerous In wireless applications, sensors gather physical information from the surroundings and transmit it to the server. Attacks on the sensor might happen while data is being sent. For example, physical data sent across wireless

Copyright: © 2026 by the authors

networks may be collected and altered by a third party. After that, he or she could provide access to a distant server that contains compromised data, putting the system in jeopardy.

Another safety risk is the dangers of routing in WSNs, i.e., Code Blue and MEDiSN, etc., designs for the exploratory situation [9]. These architectures require multi-hop surroundings, or between two nodes, in order to connect body sensors to a distant server. Thus, a malicious user might target the network layer. The attacker might take the packets and edit them, then send the changed packets to the distant centre, or back-end, which could result in a false alarm. Before sending captured packets to the next hop, an attacker may change their address information. This could cause the route to be misguided or even create an unending routing loop. Similarly, Denial-of-Service (DoS) threats [10]. It is any occurrence that reduces or removes a network's ability to perform its intended purpose. Because industrial applications require constant monitoring, denial-of-service attacks may create even more disruptions. DoS attacks are detrimental to all applications.

Such unauthorized network attacks are becoming increasingly varied and powerful. Many technologies that identify forceful threats have been created in response to the incursions. Normally, professionals refer to these kinds of systems as security systems (IDS). Only a small percentage of security systems can identify and stop modern networks or cyber-attacks, whereas some can identify several types of unauthorized access. To prevent network packets from being sniffed by third parties, a traditional system is deployed on the network. A detection system typically examines data packets and sounds an alarm if it notices any anomalies. Models for machine learning allow the auditing of massive data sets in order to structurally identify and document a variety of irregularities. Hence, the same is applied in the present work.

The following lists the main novelty-related contribution of the current study.

1. Using deep learning model, the publicly accessible CIC-DDoS2019 dataset has been examined for improved threat detection.

2. A test was were out to Analysed the model's performance using the confusion matrix's overall accuracy and efficacy properties.

There are several sections in this article. The second section provides an overview of previously published material.

2. Related Work

Because of the vulnerability to numerous attacks, such as denial-of-service, routing manipulation, and data tampering, it is important to secure IoT-integrated. Comparing to still ML techniques which are simplistic and easily implemented, in contrast, DL techniques are more precise and capable of extracting features, and new techniques are trying to find a balance between decentralization, scalability and privacy. It has critically discussed these types and their strengths, weaknesses and suitability use of the various applications. It also indicates missing points in the research that are the driving forces of the present work.

2.1 Methods for IoT-WSN Security Based on Machine Learning (ML)

In [11], Al-Quayed et al. (2024) Your training will be until the month of October in the year 2023. The researchers developed a situational forecasting model that can be used by Industry 4.0 WSN systems in detecting and preventing intrusions. The system formed its dual classification system using machine learning and deep learning classifiers which allowed the system to be able to adapt to different network situations thereby enabling the system to identify threats before they became dangerous. The architecture used was predictive analytics whereby the possible attack scenarios were predicted, which enhanced system defense mechanisms. The strategy is shown to have advantages in its ability to detect attacks that are comparable to the current attack patterns and its capability to learn new attack patterns by observing the current patterns. It is a challenge to the system since it uses more processing resources and time to process the complex activities which are not applicable in real-time running of the IoT. The

method can be utilized in Industry 4.0 where there is resource availability, but the resource-intensive Internet of Things nodes are resource-strained.

The framework proposed by Alshehri [12] will also offer a mechanism of identifying and preventing wormhole attacks yet it is applicable to the IoT as well as the WSN system. The model used the application of ML algorithms to analyze traffic flow, detect abnormalities in the routing paths and put in place corrective measures to ensure that communication channels are secured. The results of the testing showed that the system had had higher rates of packet delivery and in addition, had reduced the efficiency of the attack attempts. The method exhibits excellent safeguarding against one of the most hazardous routing assaults in WSN systems. This method involves a structured data set to validate it, which is difficult because practical applications of it in the real world seldom provide such a structured data set. The method works well when applied to WSN systems that experience fixed or partially fixed situations but is difficult to use in the context of mobile sensor networks that are subject to continual structural changes.

Rani and Taneja [13] The researchers supplied an extensive analysis, which demonstrated the process of implementing machine learning for smart application development in healthcare and industrial automation and environmental monitoring through Internet of Things wireless sensor network systems. The authors showed that machine learning technologies should be used to enhance both resource distribution and anomaly detection and forecasting decision procedures in heterogeneous Internet of Things environments. Their book contained multiple case studies that demonstrated how machine learning could address specific problems in different fields. The research demonstrates its strengths because it can be used in different industries and multiple domains of research. The system needs to handle scalability problems and deploy machine learning models at Internet of Things network edge points, which generates energy consumption challenges. The approach works well for smart city and smart healthcare purposes, but it does not apply to extremely lightweight wireless sensor networks.

Chandnani and Khairnar [14] Three goals were accomplished by the system's protocol: reducing needless data transfers, increasing energy efficiency, and prolonging the network's operating duration. The simulation's findings demonstrated that applying machine learning for routing decisions brought about better network performance because it created more dependable network connections. The model functions at its best because it enables wireless sensor networks to work for extended periods while delivering trustworthy data transmission. The solution requires extra processing tasks to execute at sensor nodes, which creates operational difficulties for devices that have limited energy capacity. This solution operates most effectively in operational environments that have between medium and large IoT systems, which require both efficiency and dependable performance, while it does not function properly in micro-sensor networks.

Saif et al. [15] The machine learning intrusion detection system (MLIDS) was created to protect body-area wireless sensor networks (BA-WSN) used in healthcare monitoring. The system used various classifiers to identify patient health data irregularities which protected both data integrity and patient safety. Compared to conventional intrusion detection systems, the system produced superior results because it delivered high accuracy and trustworthy intrusion detection capabilities. The method provides accurate results through its specialized healthcare system optimization. The system needs excessive training and processing power which creates difficulties with immediate system operations. MLIDS delivers excellent performance for healthcare Internet of Things systems that require high accuracy results while its effectiveness decreases in emergency response situations which need immediate processing capabilities.

2.2 Deep Learning (DL)-based Intrusion Detection Techniques

Muneer et al. [16] The study conducted an essential evaluation of artificial intelligence-based intrusion detection methods which specifically examined deep learning techniques. Autoencoders and CNNs and RNNs together with LSTM models have proven their capability to capture and detect complex traffic patterns which exist within Internet of Things networks. The work establishes its strengths through complete analysis which enables detailed comparison of various deep learning methods between multiple test benchmarks. The review shows deep learning potential for IoT resource-limited environments according to its identification of applications but it fails to

deliver useful implementation methods. The review assists researchers who need to select deep learning algorithms for their research work while offering limited value for immediate use in lightweight Internet of Things systems.

Sharma et al. [17] established a framework which enables explainable artificial intelligence (XAI) detection of deep learning intrusions within Internet of Things networks. The model allowed security analysts to get an understanding of the reasons behind an intrusion being raised through its ability to provide interpretability to the deep learning classifier decision-making process. The system was also found to be excellent in terms of detection and clear classification result. The method finds its advantages in the fact that its implementation meets the requirements of explainability as demanded by applications that require high levels of security and that cannot be delivered by black-box deep learning models. The approach is weak in that its application of interpretability layers brings about an increased processing cost. The methodology is comparable to the IoT systems that have low latency requirements but not comparable to the ultra-fast systems.

Awajan [18] The researchers created an innovative deep learning system which detects intrusions specifically for Internet of Things network environments. The model also got more detection rates and low false positive rates in that it used deep neural networks to process network data based on its output. The fact that the analysis concentrates on the traffic tendencies that are particular to the IoT and that contribute to the improvement of the quality of the detection in the environment that is characterized by limited resources makes the analysis the most powerful point of the study. The system is limited as it deals with training processes that are rather costly and it is impossible to expand to other Internet of Thing's applications. Since it can identify anomalies, but does not respond well in large active environments, which require models to be continuously updated, the system satisfies the needs of medium-sized Internet of Things installations.

Qazi et al. [19] suggested HDLNIDS, a solution for detecting network intrusions that uses deep learning together with other methods. The system implemented a combination of CNN and LSTM architectures to gather traffic data which included both temporal and spatial dimensions. The research achieved high accuracy results and showed the system's ability to withstand different types of attacks through testing on standard benchmark datasets. The system enables users to use LSTMs for temporal analysis together with CNNs for spatial analysis because of this feature. The system requires extended training periods together with an intricate model design which present two main disadvantages. The HDLNIDS system meets security requirements for Internet of Things networks while delivering precise results but it does not work well with wireless sensor networks that have very limited processing capacity.

Otoum et al. [20] In the system, the traffic patterns that were examined on the network included patterns that were abnormally behaved. Deep autoencoders and LSTM layers were used in the system to recognise the patterns present in network traffic, which exhibited abnormal behaviour. The system realized improved detection performance due to improved accuracy and power over its counterparts of detection systems. This framework performs optimally since it can adapt to new attack methods that can enable it to be used over an extended period of time in multiple environments. The system also needs a lot of training data which is a challenge as there is no specific Internet of Things application that provides it. DL-IDS has better performance than Internet of Things systems that run on large volumes of data but reduces the systems that require small volumes of data and operate on limited resources.

Finally, **Jayalaxmi et al. [21]** The researchers established a classification system for intrusion detection methods which they used to evaluate how effectively these methods handled current security threats. The survey displays its strengths through complete coverage of Integrating deep learning techniques which create a path forward for upcoming researchers. The study lacks thorough experimental testing to validate the methods which it presents as its primary limitation. The research work provides a detailed framework which researchers can use to develop systems yet it does not include direct implementation solutions for Internet of Things professionals.

2.3 Emerging Approaches: Federated, Edge, and Blockchain-Enabled Security

Abou El Houda et al. [22] The suggested solution makes advantage of a federated learning architecture which operates through blockchain technology to enable organizations to share their intrusion detection system across their

vehicle edge computing networks. The researchers combined federated learning distributed learning capabilities with blockchain technology which provides unchangeable and trustworthy data protection to enhance their system's capability to maintain data privacy during intrusion detection. The system demonstrated effective results by protecting against poisoning attacks while minimizing the need for data transfer in extensive vehicular network systems. The system provides operational benefits through its ability to scale, maintain user confidentiality, and permit vehicle-to-vehicle interactions which operate independently from central management. The system suffers from a disadvantage because it depends on blockchain consensus which creates delays for immediate vehicle network communication. The vehicular edge network system works effectively with this framework for establishing joint threat detection, yet it fails to function properly in scenarios that demand extremely short response times like autonomous vehicle operations.

Alzubi et al. [23] created a mobile edge computing security management solution. which operates through blockchain technology as described. The edge node system of their model used smart contracts with decentralized consensus methods to create a secure data exchange mechanism. The framework used blockchain technology to create multiple points of system access which increased trust between edge devices while eliminating single points of failure. The system offers three main advantages which include decentralized operations, better user visibility, and decreased chances of internal security breaches. The system's efficiency in extensive edge networks experiences negative effects because blockchain requires high power usage and transaction fees. The framework works best in mobile edge computing settings which need multiple parties to exchange protected data, but it does not function well in environments with limited system assets.

Kim et al. [24] The researchers examined how federated learning and blockchain technology impact safety measures in vehicle-to-everything (V2X) communication systems that operate on 6G networks. The researchers present their study which examines upcoming vehicular communication system vulnerabilities through their research on privacy-protecting authentication methods and secure data transmission techniques. The testing process showed that the system could withstand both jamming attacks and eavesdropping attempts. This method shows its strength through its ability to create 6G network designs which protect user privacy while maintaining secure communication processes. The approach has one major constraint because it depends on untested 6G technologies which remain incomplete and not yet operational. This method proves useful for upcoming vehicular IoT systems but fails to meet present requirements of 4G and 5G technologies.

Otoum et al., [25] The authors have created an energy trading platform that can be maintained with the assistance of blockchain technology along with federated learning systems to service Industry 4.0 energy management. Their paradigm used federated learning to predict the energy demand, and used blockchain to offer transparent and tamperproof transactions. Its findings revealed that energy efficiency, as well as the trading markets that are decentralized, were more credible due to their studies. To prove the existing strengths of the work, the two sustainable solutions can be provided which safeguard the energy resources and ensure the trading operations. A combination of federated learning and blockchain places restrictions on usage that confines their application to lightweight IoT devices. The model would be suitable to operate in Industry 4.0 ecosystems that would also include distributed energy systems but would not be suitable to be used in small-scale IoT networks.

Qu et al. [26] expounded a lot on blockchain-enabled federated learning, explaining its applications in the industry 4.0, IoT, health and automotive networks. The research study showed how blockchain could be applied to solve the data poisoning and model theft issues, and enhance the trust as well as the decentralization and auditability in federated learning. This is the strength of this piece of work; it has a broadness of coverage that has made it present an organized system of taxonomy of integration strategies. But as the work was less experimentally confirmed it is weak in terms of experimental confirmation of the theoretical discussion. In terms of its application within application, not a lot of deployment opportunities are provided in the current studies but the survey is still a good indicator of how the future studies should be conducted.

3. Material and methods

3.1 Dataset selection

The selection of the dataset is the first stage in classifying security risks in an Internet of Things integrated WSN network. It is owing to this fact that the author has decided to use a dataset of WSN DDoS attacks referred to as ICDDoS2019.

The dataset offers a complete set of network traffic data gathered in real-life IoT environments, which can serve researchers develop and test methods to identify deep learning and machine learning. ICDDoS2019 incorporates normal traffic and a combination of various DDOS attacks types, including as an HTTP floods, an ICMP floods, a TCP SYN floods and a UDP floods, to help it in the development of trustful anomaly detection models. The dataset has usually been applicable in the research of cybersecurity to experiment with the performance of an IDS, identify trends in attacks, and enhance the security of a network. Table 1, given below has shown the details description of dataset in term of title Authors, Published, Link for the dataset Sources etc.

Table1: Dataset metadata

Attribute	**Description**
Title	ICDDoS2019 Dataset
Authors	Dataset provided by academic researchers and cybersecurity experts
Published in	Cybersecurity research community, available through open research platforms
URL for the dataset	https://www.unb.ca/cic/datasets/ddos-2019.html
Sources	IoT-based networks, simulated DDoS attack scenarios, and real-world traffic data
Attack Types	Normal traffic, UDP, HTTP, ICMP, TCP-SYN, and UDP floods
Number of Records	Large-scale dataset (Exact count depends on the version)
Features Included	Packet length, protocol type, source/destination IP, source/destination port, timestamp, attack label, etc.
Data Format	CSV or PCAP (Packet Capture) format
Purpose	IDS and anomaly detection research, machine learning model training
Labeling	Supervised (Labeled as attack or normal traffic)
Collection Method	Traffic captured from IoT networks under controlled attack scenarios
Applications	Network security, IoT security, cybersecurity research, ML-based IDS development

3.3 Data cleansing

The dataset was selected to guarantee that the raw data would be suitable for the analytical stage of the study. The data cleansing and scrubbing procedure identified and fixed any potential flaws in the dataset that might have an impact on the prediction models. Unevenly distributed data, data impacted by noise, missing data, etc. are examples of these defects. Appropriate data normalization has been used for this. In order for the suggested models to forecast the assault, an error-free, pre-processed dataset has been suggested as input.

3.4 Dataset splitting

The neural network uses a learning-based approach and an existing dataset, as is commonly recognized. Consequently, the provided dataset has to be divided into two sets in order to be efficiently learned.

The first set is referred to as a training set, while the second is considered a testing set. One half of a two-part dataset is used to train an algorithm, and the The remaining portion is utilized for statistical analysis or testing.

Data science requires data degradation, especially when creating models from data. The datasets were split into test and training sets in a 3:1 report for the sake of this investigation.

3.5 LSTM model

The model's architecture targets four concurrent goals: limited computational capacity, speed, precision, and memory. These qualities are determined by the program's features, which necessitate that a diagnosis system be quicker and use less memory than a client-server application because of its more interactive nature. Lastly, in this instance, accuracy is impacted by the expenses related to false positives or false negatives. The architectural design of self-devised LSTM model has been presented in illustration 1.

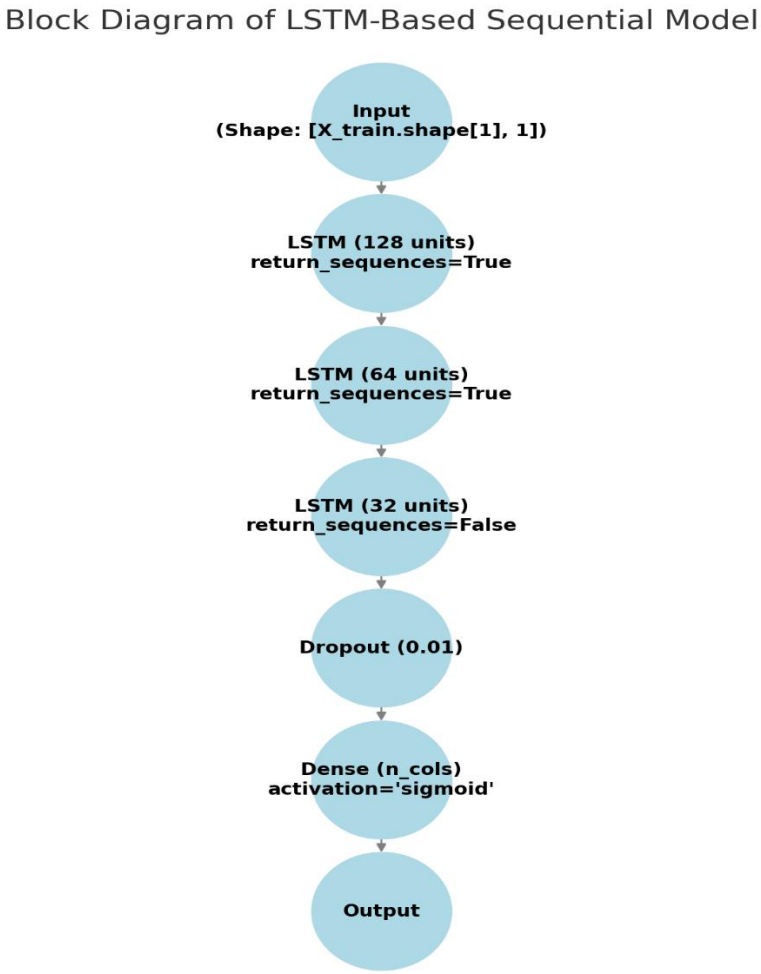


Figure 1: Architectural design of self-devised LSTM model

3.6 Layer's description

The suggested model is made up of dense abandonment and escape layers and successive layers of an artificial neural networks (ANN) short-term memory (LSTM). The architecture is made to handle sequential data efficiently while reducing over fitting through regularization.

The 1st layer is a 128-unit LSTM layer that uses 66,560 trainable parameters to learn intricate temporal relationships from an input sequence. It produces a form sequence (77, 128). With 64 units, the second LSTM layer contributes 49,408 trainable parameters while further refining the learnt features and reducing the output size to (77, 64). With 12,416 parameters, the third LSTM layer reduces the complexity by compressing the feature representations into a 32-dimensional vector.

After the last LSTM layer, to aid the model, a Dropout layer is introduced generalize more effectively by randomly dropping some neurones during training in order to prevent over-fitting. With only 264 parameters added The final one functions as the output layer of the network., which has eight units and is dense, perhaps for a classification or regression task with 8 possible outputs. The model includes 128,648 trainable parameters in total, and each layer balances feature extraction, compression, and generalization while helping the sequential data helps the model learn.

Table 2: Layers that describe the proposed model or ANN

Model: "sequential"

Layer (type)	Output Shape	Param
lstm (LSTM)	(None, 77, 128)	66,560
lstm_1 (LSTM)	(None, 77, 64)	49,408
lstm_2 (LSTM)	(None, 32)	12,416
dropout (Dropout)	(None, 32)	0
dense (Dense)	(None, 8)	264

Total params: 128,648 (502.53 KB)
Trainable params: 128,648 (502.53 KB)
Non-trainable params: 0 (0.00 B)

3.7 Model's training parameters

3.7.1 Learning Rate

In an optimization method, the tempo of learning is a tuning parameter that establishes the step length for each iteration to obtain a minimal gradient descent. Additionally, the learning rate of the network affects its capacity to merge to a local optimum, or reach the best accuracy. Therefore, the model training process should just take a few minutes if we execute it right from the beginning. The recommended LSTM model uses a learning rate of 1.00E-03.

3.7.2 The Epoch and Batch Size

The training specification's hyperparameters, batch size and epochs, may have an effect on the network's total training or instruction. The number of training dataset traversals completed by the proposed deep learning network is known as the epoch. However, the number of training samples utilized in a single cycle depends on batch sizes.

3.7.3 Optimizer Specifications

Many deep learning optimizers exist, including Adam, Adamax, RMSProp, Adadelta, SGD, and Adagrad. According to study reports, among different deep learning optimizers, the Adam optimizer outperforms the others in classification tasks. In order to overcome the present classification challenge of identifying the attacks in WSN networks, the researchers have introduced the Adam optimizer. Table 3 provides a thorough explanation of the Adam optimizer that was utilized to train the proposed LSTM model.

Table 3: Optimizer's parameters

Optimizers	Specification
Name	Adam
LR	0.001
Beta1	0.9
Beta2	0.999
Epsilon	1×10^{-8}

Table 4 lists the model's numerous other parameters.

Table4: Training parameters of Model

Parameters	**Value**
Model	Sequential
Layers	05
Hidden Layer Activation Function	relu
Activation Function for fully connected Layers	sigmoid
Loss Function	binary_crossentropy
Optimizer	Adam
Metrics	Accuracy
Batch size	100
Epochs	50

3.8 System configuration

The back-propagation algorithm was trained over 50 epochs with a batch size of 100. To optimize performance, the decay rate, dropout rate, and learning rate were set to 1e-6, 0.5, and 1e-3, respectively. All experiments were conducted on a system equipped with an Intel® Core™ i5-6200U processor running at 2.30 GHz, 8 GB of RAM, and Python 3.6, utilizing TensorFlow 1.12.0 as the backend and Keras 2.2.4. Additionally, experiments were performed on a Google Colab environment with a T4 GPU, running continuously for up to 12 hours. Training time varied across epochs depending on the GPU load at the time of execution.

Table 5: Virtual hardware details

Hardware		Details
Platform	---	Google Colab
GPU	---	T4
Programming version	---	Python 3
RAM	---	14 GB
ROM	---	64GB

3.1.8 Performance measurement

Knowing how frequently the model in question is inaccurate is frequently insufficient. Knowing how frequently it is unable to correctly anticipate a particular outcome has been increasingly important. The performance evaluation's parameters provide helpful information about how well the model can control a specific outcome. Three distinct criteria have been used to develop the suggested methodology. In many situations, it is not enough to know the frequency of incorrectness of the model in question. Knowing how often it is unable to correctly anticipate a particular result will be more important. The confusion matrix offers helpful details on how well the model forecasts a certain result.

A confusion matrix consists of four possible real-world outcomes, broadly classified into two categories: positive and negative. True Positive (TP) refers to the number of positive cases correctly identified as positive, while True Negative (TN) represents the number of negative cases accurately classified as negative. False Positive (FP) occurs when negative instances are incorrectly labelled as positive, whereas False Negative (FN) refers to positive instances that are mistakenly classified as negative. From these four values, several performance metrics can be derived, including precision, recall, F1-score, true positive rate, and false positive rate.

4. Results and analysis

This section presents the findings and a detailed evaluation of the proposed work. This section's main goal is to assess how effectively deep learning models recognize the attack when simulated using a certain dataset. The section methodically analyzes the models using a range of performance and assessment factors to identify the most effective and reliable solution for secure WSN.

4.1 Training results of LSTM

The training results demonstrate the model's exceptional accuracy and loss performance. across 50 epochs, demonstrating its high predictive precision and reliability. The results obtained from the LSTM model are presented below.

Table 6: Training per epoch LSTM

```
Epoch 1/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16s 18ms/step - accuracy:
0.9842 - loss: 0.0420 - val_Accuracy: 0.9839 - val_Loss: 0.0434
Epoch 2/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15s 17ms/step - accuracy:
0.9842 - loss: 0.0406 - val_Accuracy: 0.9856 - val_Loss: 0.0418
Epoch 3/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21s 17ms/step - accuracy:
0.9839 - loss: 0.0427 - val_Accuracy: 0.9853 - val_Loss: 0.0420
Epoch 4/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20s 17ms/step - accuracy:
0.9847 - loss: 0.0398 - val_Accuracy: 0.9847 - val_Loss: 0.0412
Epoch 5/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24s 21ms/step - accuracy:
0.9850 - loss: 0.0396 - val_Accuracy: 0.9852 - val_Loss: 0.0408
Epoch 6/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17s 18ms/step - accuracy:
0.9839 - loss: 0.0398 - val_Accuracy: 0.9854 - val_Loss: 0.0397
Epoch 7/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15s 18ms/step - accuracy:
0.9842 - loss: 0.0418 - val_Accuracy: 0.9854 - val_Loss: 0.0401
Epoch 8/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20s 17ms/step - accuracy:
0.9843 - loss: 0.0400 - val_Accuracy: 0.9852 - val_Loss: 0.0421
Epoch 9/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15s 17ms/step - accuracy:
0.9855 - loss: 0.0387 - val_Accuracy: 0.9840 - val_Loss: 0.0421
Epoch 10/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15s 17ms/step - accuracy:
0.9852 - loss: 0.0400 - val_Accuracy: 0.9854 - val_Loss: 0.0423
Epoch 11/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21s 17ms/step - accuracy:
0.9843 - loss: 0.0409 - val_Accuracy: 0.9855 - val_Loss: 0.0411
Epoch 12/50
```

```
877/877 ━━━━━━━━━━━━━━━━━━━━ 20s 17ms/step - accuracy:
0.9846 - loss: 0.0392 - val_Accuracy: 0.9845 - val_Loss: 0.0410
Epoch 13/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 24s 21ms/step - accuracy:
0.9844 - loss: 0.0388 - val_Accuracy: 0.9847 - val_Loss: 0.0414
Epoch 14/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 18s 18ms/step - accuracy:
0.9850 - loss: 0.0403 - val_Accuracy: 0.9856 - val_Loss: 0.0420
Epoch 15/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 20s 18ms/step - accuracy:
0.9847 - loss: 0.0397 - val_Accuracy: 0.9854 - val_Loss: 0.0412
Epoch 16/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 15s 17ms/step - accuracy:
0.9846 - loss: 0.0410 - val_Accuracy: 0.9860 - val_Loss: 0.0412
Epoch 17/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 16s 18ms/step - accuracy:
0.9856 - loss: 0.0380 - val_Accuracy: 0.9866 - val_Loss: 0.0399
Epoch 18/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 20s 18ms/step - accuracy:
0.9840 - loss: 0.0410 - val_Accuracy: 0.9866 - val_Loss: 0.0398
Epoch 19/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 23s 21ms/step - accuracy:
0.9841 - loss: 0.0410 - val_Accuracy: 0.9826 - val_Loss: 0.0476
Epoch 20/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 18s 17ms/step - accuracy:
0.9854 - loss: 0.0385 - val_Accuracy: 0.9828 - val_Loss: 0.0493
Epoch 21/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 19s 21ms/step - accuracy:
0.9854 - loss: 0.0381 - val_Accuracy: 0.9860 - val_Loss: 0.0438
Epoch 22/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 18s 18ms/step - accuracy:
0.9850 - loss: 0.0395 - val_Accuracy: 0.9862 - val_Loss: 0.0394
Epoch 23/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 23s 20ms/step - accuracy:
0.9848 - loss: 0.0395 - val_Accuracy: 0.9860 - val_Loss: 0.0404
Epoch 24/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 17s 17ms/step - accuracy:
0.9850 - loss: 0.0389 - val_Accuracy: 0.9866 - val_Loss: 0.0392
Epoch 25/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 23s 20ms/step - accuracy:
0.9854 - loss: 0.0379 - val_Accuracy: 0.9845 - val_Loss: 0.0419
Epoch 26/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 18s 17ms/step - accuracy:
0.9846 - loss: 0.0404 - val_Accuracy: 0.9854 - val_Loss: 0.0418
Epoch 27/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 20s 17ms/step - accuracy:
0.9852 - loss: 0.0385 - val_Accuracy: 0.9849 - val_Loss: 0.0439
Epoch 28/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 21s 18ms/step - accuracy:
0.9842 - loss: 0.0408 - val_Accuracy: 0.9848 - val_Loss: 0.0412
Epoch 29/50
877/877 ━━━━━━━━━━━━━━━━━━━━ 20s 18ms/step - accuracy:
0.9854 - loss: 0.0388 - val_Accuracy: 0.9846 - val_Loss: 0.0423
Epoch 30/50
```

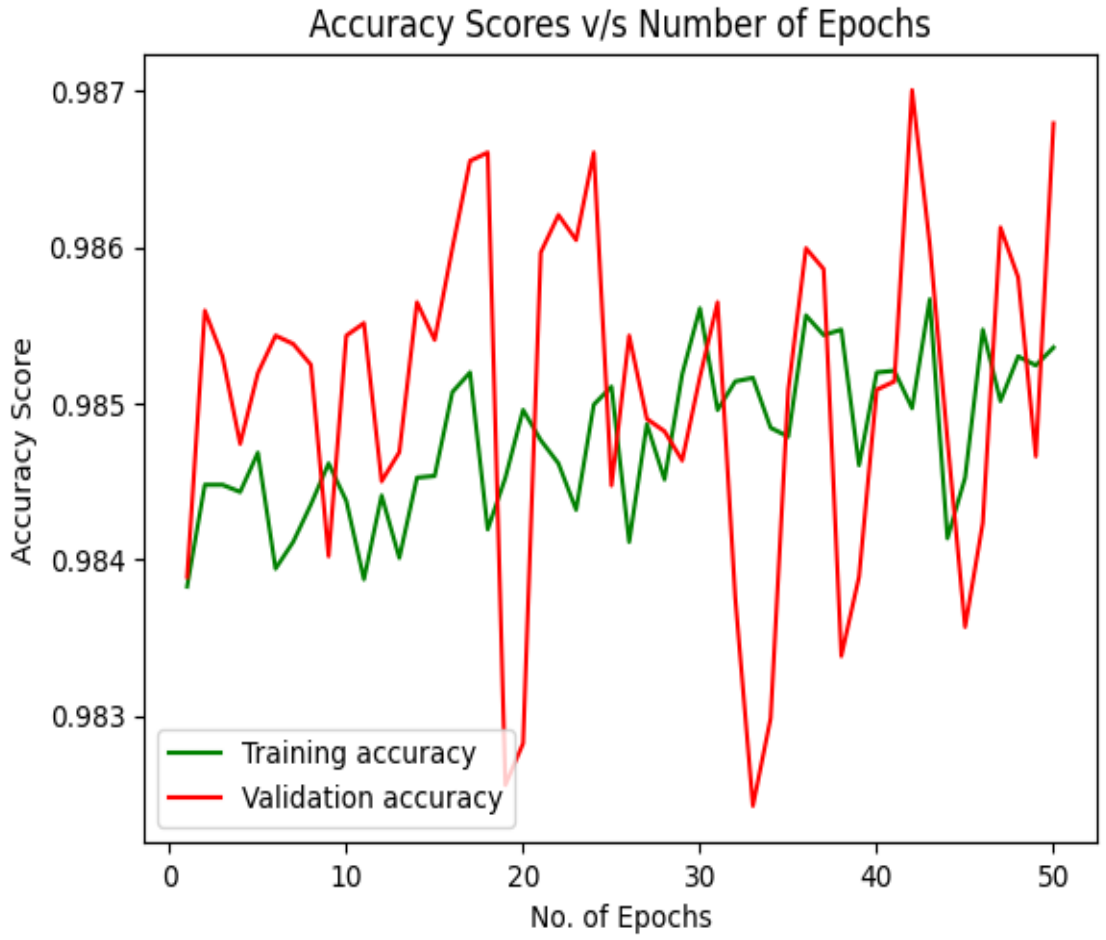
```
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20s 18ms/step - accuracy:
0.9854 - loss: 0.0385 - val_Accuracy: 0.9852 - val_Loss: 0.0394
Epoch 31/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20s 17ms/step - accuracy:
0.9854 - loss: 0.0388 - val_Accuracy: 0.9856 - val_Loss: 0.0405
Epoch 32/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15s 17ms/step - accuracy:
0.9846 - loss: 0.0393 - val_Accuracy: 0.9838 - val_Loss: 0.0435
Epoch 33/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15s 17ms/step - accuracy:
0.9853 - loss: 0.0381 - val_Accuracy: 0.9824 - val_Loss: 0.0491
Epoch 34/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21s 18ms/step - accuracy:
0.9848 - loss: 0.0396 - val_Accuracy: 0.9830 - val_Loss: 0.0457
Epoch 35/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20s 17ms/step - accuracy:
0.9841 - loss: 0.0402 - val_Accuracy: 0.9851 - val_Loss: 0.0414
Epoch 36/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17s 19ms/step - accuracy:
0.9854 - loss: 0.0386 - val_Accuracy: 0.9860 - val_Loss: 0.0417
Epoch 37/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15s 17ms/step - accuracy:
0.9853 - loss: 0.0379 - val_Accuracy: 0.9859 - val_Loss: 0.0426
Epoch 38/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21s 18ms/step - accuracy:
0.9849 - loss: 0.0391 - val_Accuracy: 0.9834 - val_Loss: 0.0460
Epoch 39/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15s 17ms/step - accuracy:
0.9833 - loss: 0.0404 - val_Accuracy: 0.9839 - val_Loss: 0.0429
Epoch 40/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15s 17ms/step - accuracy:
0.9849 - loss: 0.0378 - val_Accuracy: 0.9851 - val_Loss: 0.0433
Epoch 41/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 23s 20ms/step - accuracy:
0.9853 - loss: 0.0371 - val_Accuracy: 0.9851 - val_Loss: 0.0443
Epoch 42/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 18s 18ms/step - accuracy:
0.9853 - loss: 0.0395 - val_Accuracy: 0.9870 - val_Loss: 0.0388
Epoch 43/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20s 17ms/step - accuracy:
0.9855 - loss: 0.0371 - val_Accuracy: 0.9860 - val_Loss: 0.0390
Epoch 44/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21s 17ms/step - accuracy:
0.9841 - loss: 0.0427 - val_Accuracy: 0.9848 - val_Loss: 0.0422
Epoch 45/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20s 17ms/step - accuracy:
0.9849 - loss: 0.0380 - val_Accuracy: 0.9836 - val_Loss: 0.0446
Epoch 46/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21s 18ms/step - accuracy:
0.9850 - loss: 0.0367 - val_Accuracy: 0.9842 - val_Loss: 0.0436
Epoch 47/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20s 17ms/step - accuracy:
0.9851 - loss: 0.0376 - val_Accuracy: 0.9861 - val_Loss: 0.0394
Epoch 48/50
```

```
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15s 17ms/step - accuracy:
0.9860 - loss: 0.0364 - val_Accuracy: 0.9858 - val_Loss: 0.0413
Epoch 49/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21s 18ms/step - accuracy:
0.9852 - loss: 0.0368 - val_Accuracy: 0.9847 - val_Loss: 0.0425
Epoch 50/50
877/877 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15s 17ms/step - accuracy:
0.9847 - loss: 0.0383 - val_Accuracy: 0.9868 - val_Loss: 0.0397
```

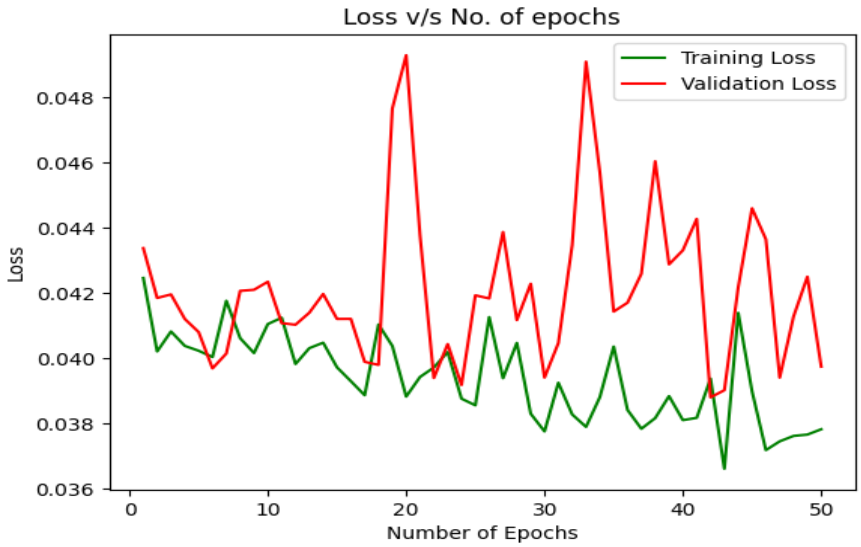
With the validation accuracy coming very close to the training accuracy, the model has a continuously high accuracy which levels off at around 98.5%. The loss values can be taken as an indication of good learning and reflect slight changes, but generally decrease with epochs. Conversely, occasional spikes in the loss of validation can be attributed to model instability or slight over fitting.

4.2 Graphical analysis of LSTM

This is where in the figure 2 accuracy curve of LSTM has been applied. Fig. 2 The graph demonstrates that when there are more than two epochs, then the model can distinguish between the positive and negative classification. Firstly, training and validation scores are high, being close to the upper limit of 1.0 which is good predictability.



(a)



(b)

Figure 2: (a) Accuracy curve (b) Loss curve

The LSTM model's accuracy plot demonstrates a steady rise in training precision with only minor changes. Despite minor fluctuations, the training accuracy and validation accuracy stay comparatively close, indicating that the model is learning effectively.

4.3 Confusion matrix results

The confusion matrix for LSTM has been presented in figure 3. As per the figure, the classification performance of the model is good.

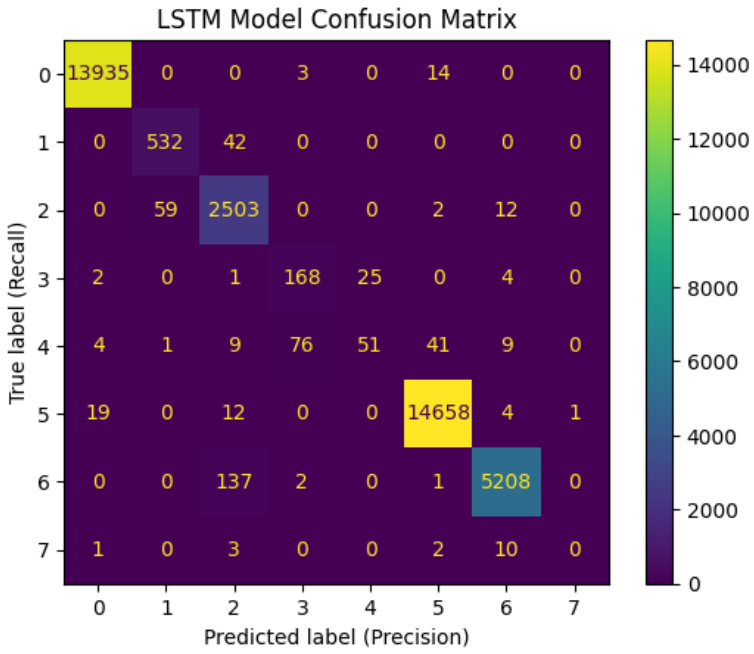


Figure 3: CM for LSTM

The LSTM model has a high level of success with attack classification in the Wireless Sensor Networks (WSN) at 98.68 per cent. The model is highly accurate and therefore very reliable in detecting the attacks. This confirms the fact that almost all the data points are well categorised. This model almost always sufficiently predicts the attacks with a 98.61 accuracy that reduces the number of false positives to a minimal level. The model has a very good recall of 98.68% which is actually very good in distinguishing virtually all actual assaults and making sure that virtually none on actual assaults is missed. This moderate accuracy and probability of recall that allows the model to detect the threats without causing any false alarm is reflected in its F1 Score of 98.59. Combined, these indicators demonstrate that the LSTM model has the ability to detect the existence of WSN attacks with a huge level of accuracy and minimal errors.

5. Conclusion

To simulate the model developed, the ICDDoS2019 Dataset was used. The classification of wireless sensor network (WSN) attack using the LSTM model has been found to perform quite well with high recall (98.68%), accuracy (98.68%), and precision (98.61%). These findings illustrate that the model is able to bracket between an attack and non-assault situations with precision and minimum errors. The excellent recall demonstrates Because of the high level of accuracy of the model, virtually all real strikes can be identified, and attacks are nearly always correct. The model has a fare weighted average of the recall and precision at 98.59% F1 Score which makes it a good attack detection tool. These results demonstrate that the LSTM model with its resilience and accuracy in classification of an attack can be effectively implemented in the actual WSN systems. Its performance suggests that it could have a tremendous effect on security as it could reduce the likelihood of an attack.

REFERENCES

1. Xxxxxx, M. (2021). From the wireless sensor networks (WSNs) to the Web of Things (WoT): an overview. *Journal of Intelligent Systems and Internet of Things*.
2. Beirão, M. L. M., Maia, A. H., Mota, E., Rangel, E., Costa, D. G., Jurgut, D., & Villas, L. A. (2021). A traffic data clustering framework based on fog computing for VANETs. *Vehicular Communications*, *31*, 100370.
3. Bahrami, A. M., Gia, T. N., Negash, B., Anzanpour, A., Azimi, I., Jiang, M., & Liljeberg, P. (2018). Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. *Future Generation Computer Systems*, *78*, 641-658.
4. Stone, T. (2022) The importance of Healthcare Data Security: Solutions & Tips, Prime TSR. Available at: https://primetsr.com/insights/the-importance-of-healthcare-datasecurity/#:~:text=Data%20security%20is%20currently%20one,breaches%20occurred%20in%202020%20alone. (Accessed: 13 January 2025).
5. Nagakannan, M., Inbaraj, C. J., Kannan, K. M., & Ramkumar, S. (2018, August). A recent review on IoT based techniques and applications. In *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2018 2nd International Conference on* (pp. 70-75). IEEE.
6. Chopra, K., Gupta, K., & Lambora, A. (2019, February). Future internet: The internet of things-a literature review. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)* (pp. 135-139). IEEE.
7. Lu, H., Li, J., & Guizani, M. (2013). Secure and efficient data transmission for cluster-based wireless sensor networks. *IEEE transactions on parallel and distributed systems*, *25*(3), 750-761.
8. Anup, J. A., & Shah, A. S. (2017). Routing and Security Issues in U-Healthcare Mobile, Ubiquitous and Wireless Body Area Network (WBAN). *Int J Adv Sci Technol*, *109*, 23-34.
9. Islam, M. N. U., Fahmin, A., Hossain, M. S., & Ahsan-uz-zaman, M. (2021). Denial-of-service attacks on wireless sensor network and defense techniques. *Wireless Personal Communications*, *116*(3), 1993-2021.
10. Al-Qaysi, F., Ahmad, Z., & Hamzah, M. (2024). A situation based predictive approach for cybersecurity intrusion detection and prevention using machine learning and deep learning algorithms in wireless sensor networks of industry 4.0. *Ieee Access*, *12*, 34800-34819.

11. Akbehri, A. H. (2024). Wormhole attack detection and mitigation model for Internet of Things and WSN using machine learning. *PeerJ Computer Science*, *10*, e2257.
12. Rani, S., & Taneja, A. (Eds.). (2024). *WSN and IoT: An integrated approach for smart applications*. CRC Press.
13. Chandnani, N., & Khairnar, C. N. (2023). A Reliable Protocol for Data Aggregation and Optimized Routing in IoT WSNs based on Machine Learning. *Wireless Personal Communications*, *130*(4).
14. Saif, S., Karmakar, K., Biswas, S., & Neogy, S. (2022). MLIDS: Machine learning enabled intrusion detection system for health monitoring framework using BA-WSN. *International journal of wireless information networks*, *29*(4), 491-502.
15. Muneer, S., Farooq, U., Athar, A., Ahsan Raza, M., Ghazal, T. M., & Sakib, S. (2024). A critical review of artificial intelligence based approaches in intrusion detection: A comprehensive analysis. *Journal of Engineering*, *2024*(1), 3909173.
16. Sharma, B., Sharma, L., Lal, C., & Roy, S. (2024). Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning based approach. *Expert Systems with Applications*, *238*, 121751.
17. Awajan, A. (2023). A novel deep learning-based intrusion detection system for IOT networks. *Computers*, *12*(2), 34.
18. Qazi, E. U. H., Faheem, M. H., & Zia, T. (2023). HDLNIDS: hybrid deep-learning-based network intrusion detection system. *Applied Sciences*, *13*(8), 4921.
19. Otoum, Y., Liu, D., & Nayak, A. (2022). DL-IDS: a deep learning–based intrusion detection framework for securing IoT. *Transactions on Emerging Telecommunications Technologies*, *33*(3), e3803.
20. Jayalaxmi, P. L. S., Saha, R., Kumar, G., Conti, M., & Kim, T. H. (2022). Machine and deep learning solutions for intrusion detection and prevention in IoTs: A survey. *IEEE Access*, *10*, 121173-121192.
21. Abou El Houda, Z., Moudoud, H., Brik, B., & Khoukhi, L. (2024). Blockchain-enabled federated learning for enhanced collaborative intrusion detection in vehicular edge computing. *IEEE Transactions on Intelligent Transportation Systems*, *25*(7), 7661-7672.
22. Alzubi, J. A., Alzubi, O. A., Singh, A., & Mahmood Alzubi, T. (2023). A blockchain-enabled security management framework for mobile edge computing. *International Journal of Network Management*, *33*(5), e2240.

23. Kim, M., Oh, I., Yim, K., Sahlabadi, M., & Shakur, Z. (2023). Security of 6G-enabled vehicle-to-everything communication in emerging federated learning and blockchain technologies. *IEEE access*, *12*, 33972-34001.

24. Otoum, S., Al Ridhawi, I., & Mouftah, H. (2022). A federated learning and blockchain-enabled sustainable energy trade at the edge: A framework for industry 4.0. *IEEE Internet of Things Journal*, *10*(4), 3018-3026.