Received: 22 December 2024 | Revised: 15 July 2025 | Accepted: 14 September 2025 | Published online: 2 October 2025

RESEARCH ARTICLE

Artificial Intelligence and Applications
2025, Vol. 00(00) 1-19
DOI: 10.47852/bonviewAIA52025066

BON VIEW PUBLISHING

IDS-IoT: Intrusion Detection System for the Internet of Things Using Enhanced Long-Short Term Memory

Gaurav Meena[1,*], Ajay Indian[1]

[1] *Department of Computer Science, Central University of Rajasthan, India*

Abstract: Network security and intrusion detection have become significant challenges with the emergent inclusion of Internet of Things (IoT) devices across several domains. In this article, we proposed an enhanced long-short term memory (E-LSTM) method for detecting intrusions in IoT-based datasets for the designing of more resilient and competent intrusion detection systems (IDSs) in the dynamic domain of IoT environments, as well as for the thoughtful selection of models appropriate for various dataset characteristics. Four distinct datasets were used in this study: KDD-Cup'99, NSL-KDD, UNSW-NB15, and CICIoT2023. The aim was to estimate and compare the performance across datasets. We provide subtle insights into model behaviors and their capacity to adjust to the particulars of each dataset through rigorous analysis. The proposed enhanced LSTM approach revealed significant differences in precision, recall, accuracy, and F1-score compared to other approaches like AdaBoost, DNN, RNN, and Logistic Regression. It was discovered that, for every dataset, the accuracy rate exceeded 95%.

Keywords: security, IoT, intrusion detection system, LSTM, deep learning

1. Introduction

Information is being produced exponentially in almost every technological and commercial domain because of the speedy expansion of distributed and Internet-based techniques such as cloud computing and the Internet of Things (IoT) [1–3]. As part of the IoT, various objects, sensors, and intelligent nodes that offer self-sufficient communication capabilities are being integrated [4]. Smart sensors that monitor and gather social media and other device data are examples of hardware resources that are part of the IoT [5, 6]. A network of physical entities that can be seen and tamed online is called the IoT [7, 8]. In addition, loads of items are accessible through wireless sensor networks (WSN) using various Internet-connected actuators and sensors [9, 10]. There is a need to calculate, save, retrieve, and evaluate IoT data, since IoT sensors often have little memory, low power, battery constraints, and network limits. So, they must be calculated, saved, retrieved, and evaluated [11, 12].

Additionally, a platform is required to handle the expanding volume of heterogeneous data and objects [13, 14]. Our lives are beginning to change because of the quick evolution of the IoT. Furthermore, with a linked and collaborative system, security has emerged as one of the most thought-provoking problems owing to the explosive expansion of IoT [15]. Numerous risky behaviors can arise from data vulnerability, which can directly lower the security of IoT [16]. Thus, eavesdropping is possible in many wireless transmissions without restrictions. The unauthorized use of computer systems targets intrusion detection systems (IDS) [17]. Among these access control methods and networks are ad hoc networks, WLANs, WPANs, and LANs. Within the WPAN family, WSN, mobile phones, and RFID are the most widely used networks [18]. Consequently, security has turn out to be a major problem in IoT [19]. Current security techniques include cryptographic security bases and systemic security systems [20]. Additionally, modern cyber threats have grown more intricate, sophisticated, unconventional, and insistent due to the fast growth of adversarial tactics. In light of these considerations, identifying and thoroughly mapping cyber threats is a crucial but sometimes disregarded step before implementing effective and robust cybersecurity remedies [21].

Intrusion detection systems (IDS) come in three primary varieties: specification-based, anomaly-based, and misuse-based. Loads of research has been done on anomaly-based and misuse-based IDSs by both industry and academics [22]. Due to the growth of IoT and the increase in cyberattacks, several IDS approaches have been developed in the IoT area [23]. Furthermore, an IDS is essential for identifying and stopping intrusions into the network and giving a strong security framework [24]. The complete number of sensors in a system has limitations related to storage, interoperability, and scalability, while the requirement for a significant volume of labeled data for performance, intrusion detection, and attack detection hampers security methods such as IDSs. IDSs are a basic and necessary security technique that protects IoT settings against various security threats. IDSs often have four primary types. The first type of IDS outlines the traits for every type of assault that can be identified and is signature-based. An alert sounds when the anomalous behavior aligns with the pattern [25], a straightforward technique for identifying known attacks. The anomaly-based IDS's initial phase gathers data on the predicted actions of the discovered system. Subsequently, the IDS set a threshold that the suspicious activity must surpass for the warning to trigger. Even though this system can detect attack attempts, processing data needs lots of memory and computing [26]. An IDS combined with a signature-based IDS can potentially balance the computing burden and false-positive signal issue of an anomaly-based IDS with the storage cost of a signature-based detection system [27]. An IDS uses a specification-driven methodology to compare current operations with its identified common system activities [28].

***Corresponding author:** Gaurav Meena, Email: gaurav.meena@curaj.ac.in

Intrusion is any illegal or unwelcome entry, exit, or activity in a computer system, network, or digital environment. Usually, it entails people or organizations trying to overcome security barriers to access, alter, or steal private information; interfere with system performance; or harm themselves. Various incursions, including hacking, malware infection, breaches of data, or any other act, can violate security, privacy, or access to digital resources. Therefore, intrusion detection is crucial to protect against cyber threats. Given the growing adoption of IoT devices and evolving threat landscapes, it is vital to assess the efficacy of intrusion detection techniques for safeguarding IoT systems. However, various datasets exist for evaluating intrusion detection algorithms, each capturing different aspects of the network behavior and attacks. The research problem is to systematically analyze diverse datasets, specifically KDD-Cup'99, NSL-KDD, UNSW-NB15, and CICIoT2023, to evaluate the performance of different intrusion detection methods in an IoT network. Deep learning (DL) methods have been demonstrated to improve prediction and classification. There is sufficient interest in deep learning across several fields to address various issues. The machine learning methods go beyond some of the limitations of traditional approaches, like support vector machines. The fact that the effectiveness of IDS relies mostly on the dataset used for testing presents another issue that must be resolved. Several benchmark datasets had varying attributes and insufficient results. These issues have led to the growth of automated and reliable IDSs that shed light on the adaptability and generalizability of intrusion detection solutions in IoT scenarios by revealing the benefits and shortcomings of various procedures when applied to these datasets. The following contributions aided the completion of this research.

1) We suggested an enhanced LSTM based method for detecting intrusions in IoT-based datasets to design more resilient and competent IDSs.
2) Diversity of Datasets: This research involves studying four distinct datasets, each with its own characteristics, attack patterns, and network behaviors. This diversity reflects the heterogeneous nature of real-world IoT environments.
3) Performance Evaluation: This study evaluated performance of intrusion-detection methods across different datasets using various metrics.
4) IoT Context: The research problem is framed explicitly in the reference of IoT, acknowledging the unique challenges and features of IoT networks, for instance resource constraints, device heterogeneity, and evolving attack vectors.
5) Consider the KDD-Cup'99, NSL-KDD, UNSW-NB15, and CICIoT2023 datasets for IoT-based IDS.
6) Their performance was evaluated on various datasets to determine the merits and demerits of intrusion-detection methods in diverse situations.

The rest of the proposed study is structured as follows. Section 2 covers the history of IoT-IDS and common terminology. Section 3 discusses the IoT Intrusion Detection framework and techniques. The experimental setup and findings are explained in Section 4. Various topics were covered, including the properties of the utilized dataset, the preprocessing techniques used, and details of the alternative models included in the research. Section 5 presents the final components of our study.

2. Related Work

Some scholars have developed IDSs. To bridge the gap among the current state-of-art and its prospective future direction, we began by examining survey research articles on IDS. The idea behind the IoT uses sensors, software, and connections to connect physical equipment, cars, apps, and other items. They can gather and share data online. These gadgets can communicate with one another and their surroundings to facilitate intelligent decision making, long-range monitoring, and data-centric automation. An IDS is a standalone component in the security domain that is crucial for information security. The necessity of security and maintaining the system's awareness of dangerous activities is growing, along with the ease with which people over a broad region may use the internet. The conventional IDS has limits like poor detection accuracy and high false alarms [29].

The rising usage of IoT devices has revolutionized the efficiency and ease of daily living. Nevertheless, new security issues have arisen owing to this expansion. The number of cyberattacks has surged owing to the growing usage of the internet and network technologies, which has boosted interest in IDS among academics. Network intrusion detection (NID) has become a critical component of contemporary security architectures for securing IoT networks. Machine-learning techniques have recently demonstrated potential as IDS solutions. However, the functional and physical variety of IoT-IDS systems poses obstacles, making the total feature usage unfeasible. Therefore, effective feature selection is crucial. An novel feature selection scheme for anomaly-based NIDS was proposed in this study [30, 31].

Machine learning (ML) approaches have been adopted in IDS to identify and categorize security vulnerabilities in various applications, such as artificial intelligence, IoT, smart city infrastructure, and fifth-generation networks [32]. In this article, we used the KDD-CUP dataset to classify the intrusions. It compares its performance with state-of-the-art results for many ML algorithms, for example Random Forest (RF) and Classification and Regression Trees (CART). Researchers have created IDSs [33] that use ML techniques to overcome these problems. These techniques can recognize unusual assaults and accurately distinguish between normal and abnormal data. One subset of ML termed as deep learning has become a hot topic for study. This review offers classification based on data objects for classifying and reviewing IDS- and ML-based deep learning publications. SVM with normalization performs better than SVM without normalization when identifying interruption data from KDD99 and Min–Max, as demonstrated in refs. [34, 35]. Normalization does well than the other techniques in terms of speed, number of support vectors, and cross-validation accuracy.

The increase in the accuracy of intrusion classification with less input data for the input subsets of the NSL-KDD-Cup'99 dataset was examined using the SVM classifier [36]. The suggested method produced a 91% accuracy rate with only three features, and a 99% accuracy rate with 36 features. Notably, an accuracy rate of 99% was achieved using all 41 training characteristics. Anomalous occurrences in network traffic flow were identified, and machine learning was applied. SVM, Naive Bayes, RF, and Decision Tree classification, which process data using Apache Spark, were examined in this study. The Random Forest outperformed others in network intrusion detection by utilizing all 42 features, as per the experimental results achieved on the new publicly available dataset UNSW-NB15 [37].

The authors in refs. [38, 39] suggested a framework designed to raise the efficacy of IDS by using data derived from Internet of Things (IoT) settings. This context employs DL methods and meta-heuristic optimization procedures to facilitate feature mining and selection. A straightforward convolutional neural network (CNN) is the primary feature extractor. A feature selection strategy, fixed in the Reptile Search Algorithm, is proposed to focus only on the most essential features derived from the CNN's extracted features. The proposed framework performed well across classification metrics in assessments of diverse datasets.

In ref. [40], the authors focused on software-defined networking (SDN), a paradigm change which improves network openness and flexibility. However, this change has created a vulnerable environment with significant hazards such as network complexities and online fraud. In recent years, research has been an essential advancement owing to the combination of SDN and clever ML techniques in IDS. In this context, the authors suggest an HFS-LGBM IDS tailored for SDN that uses LightGBM and Hybrid Feature Selection algorithms to recognize and categorize threats. Evaluations exhibited that the suggested system outpaces existing approaches. In ref. [41], the authors provided a novel method for identifying network intrusions, specifically illegal actions in computer networks. This section describes the operation and effectiveness of the suggested approach. We tested it on a standard dataset called KDD-Cup'99. Our approach focuses on deep learning carried out using specific software applications. A high accuracy of 99.65% was achieved. This approach can be helpful for DL-based detection, classification, and network security research.

Detecting intrusions and irregular activities in computer networks suggests a machine learning technique termed AE-LSTM. Unexpected behavior is well identified by a method that integrates an LSTM and an autoencoder (AE) model. It uses a standard scalar preprocessing technique to ensure the model functions effectively if the input is imbalanced. This outcome aids in the removal of inappropriate data. The main objective of AE-LSTM is to find out whether network activity is regular. They tested it with the NSL-KDD dataset and found it to be more accurate to a greater extent than the earlier methods. It achieved 98.69% and 98.70% accuracy for different intrusions, and 98.78% for differentiating malicious from regular network traffic [42].

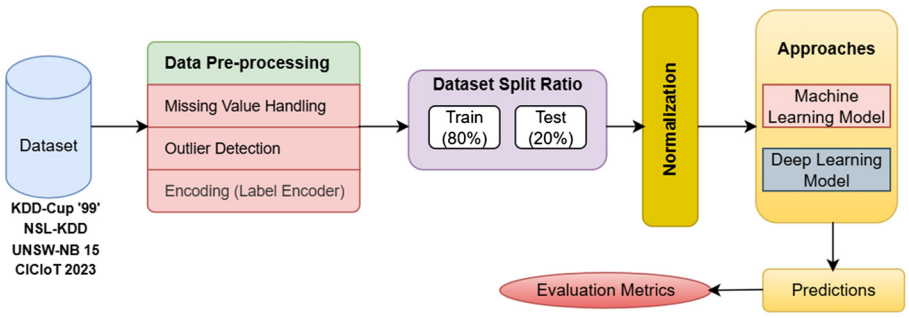
They utilized a feature selection technique (CFS-DE) to determine the essential data properties and the Weighted Stacking method to precisely differentiate between regular and suspected network activities to build a system to identify network intrusions. By streamlining and enhancing the categorization process, this strategy can increase the accuracy. They evaluated the precision of identifying intrusion into computer networks by applying the NSL-KDD and CSE-CIC-IDS2018 datasets [43]. They classified several DL methods for detecting intrusion in their work and reviewed the pertinent literature. They then evaluated and tested four crucial DL models to identify intrusions, comprising feed-forward algorithms, autoencoders, deep belief networks, and LSTM networks, using both more current datasets (CIC-IDS2017, CIC-IDS2018) and older datasets (KDD-Cup'99, NSL-KDD). The findings showed that deep feed-forward neural network models perform better than shallow neural networks on all four datasets, offering high accuracy and F1 score while being efficient for neural network training and decision-making. Interestingly, autoencoders and deep belief networks, two popular semi-supervised models, underperform supervised feed-forward neural networks [44, 45]. In ref. [46], the authors developed a unique ID using a novel tree-CNN hierarchical algorithm and soft-root-sign (SRS) activation approach. This technology rapidly recognizes attacks such as DDoS, infiltration, brute force, and online attacks. The new hierarchical method substantially speeds up the detection procedure by approximately 36%, while maintaining a mean detection rate of 98% across all tested cyberattacks.

A comparative assessment of current works is shown in Table 1. Based on previous studies, several comparative analyses of IDS models have been conducted. Though, there is a gap for further development in terms of accuracy. It provides a comprehensive overview of different studies efforts and methods for intrusion detection systems based on datasets — KDD-Cup'99, NSL-KDD, UNSW-NB15, and CICIoT2023. Each entry includes information about the authors, publication year, dataset used, techniques for extracting essential features or reducing data, type of classifier used, and the reported accuracy. Various approaches have been employed, such as the SVM, DNN, and other ML and DL models. The DNN model shows significant potential in complex IoT environments, such as CICIoT2023, achieving noteworthy accuracy. It is essential to remember that the choice of the dataset and the specific characteristics of the intrusion detection problem significantly influence which method works best. The DNN shows promising results; the optimal choice determines by the elements such as dataset size, complexity, and patterns of intrusions.

Table 1
Comparative review of existing works

Authors	Years	Dataset	Feature extraction used	Method	Accuracy
S. Mukherjee, et al. [47]	2012	NSL-KDD	Feature vitality-based	Naive Bayes	97.78
R. M. Elbasiony, et al. [48]	2013	KDD-Cup'99	-	Random Forests + Weighted K-means	98.3
E. D. L. Hoz, et al. [49]	2015	KDD-Cup'99	PCA-FDR	PSOM	88 %
Farnaaz, N., et al. [50]	2016	NSL-KDD	Symmetrical uncertainty	Random Forest	99.67
Belavagi et al. [51]	2016	NSL-KDD	-	Multiple (RF, LR, SVM, and Naive Bayes)	99% with RF
Thaseen, I. S., et al. [52]	2017	NSL-KDD	Chi-square	SVM	98
Meena, G., and Choudhary [44]	2017	KDD-Cup'99, NSL-KDD	-	Naive Bayes	92.71
S. Ahmad, F. Arif, Z. Zabee-hullah [54]	2020	KDD-Cup'99	relu, SoftMax	DNN	99.91
A. R. Gupta, et al. [55]	2020	NSL-KDD	-	CNN	Not reported
R. Abou Khamis, et al. [56]	2020	UNSW-NB15	PCA	DNN	92% without PCA 93% with PCA
H. Hindy, et al. [57]	2020	NSL-KDD	-	Autoencode	92.96
Euclides Carlos Pinto Neto, et al. [58]	2023	CICIoT 2023	Features extracted from Pcap files	Random Forest	99.16

Figure 1
Proposed framework for intrusion detection based on enhanced LSTM approach



The diversity of the methods used highlights the intricate nature of intrusion detection and emphasizes the need to tailor techniques to suit the unique features of each dataset. While random forests and DNN have emerged as strong options, their performance depends on the dataset's specific attributes and the types of intrusions being detected. Ultimately, selecting the most suitable method requires a thorough understanding of the dataset's characteristics, ensuring that the chosen approach aligns effectively with the complexities of the intrusion detection task.

The research gap in this study reveals the lack of comprehensive comparisons of various intrusion detection techniques across multiple datasets. There is a dearth of research that compares various methodologies using a variety of datasets, although individual studies have independently evaluated certain datasets and methods. Such an approach would result in greater knowledge of techniques that work well in various situations.

Furthermore, studies addressing the difficulties associated with spotting breaches in IoT networks are lacking. Given their increasing use, it is necessary to address the particular security risks of IoT devices. However, current research has not examined the function of the intrusion detection technique regarding the IoT structure and the types of assaults that target these devices. The research gap requires a thorough study that contrasts various intrusion detection techniques using numerous datasets and explores the difficulties in protecting IoT networks against intrusions.

3. Proposed Framework

We used the following workflow for our proposed enhanced LSTM-based model and other ML-based models, which are used here for the comparative study. The workflow steps include data gathering, preprocessing, feature extraction, model calibration, and emotion identification. Figure 1 shows the complete process.

Built on the specific analytic objectives or research goals, the number of methodologies used within the "KDD-Cup'99," "NSL-KDD," "UNSW-NB15," and "CICIoT2023" datasets might vary. Researchers have used each dataset for various purposes, including intrusion detection, classification, and evaluation of security approaches. Our study employed the proposed enhanced LSTM approach for all datasets and compared it with other ML approaches for instance logistic regression and AdaBoost. By identifying intricate patterns in the data and attaining high accuracy, intrusion detection models that categorize network traffic as malicious or normal have been developed using enhanced LSTM.

3.1. Dataset description

The research gap reveals the lack of thorough comparisons of various intrusion detection techniques, specifically the deep learning approach, across multiple datasets. To fill that gap, we have used the "KDD-Cup'99," "NSL-KDD," "UNSW-NB15," and "CICIoT2023" datasets, which consist of a variety of data that leads to the design of more effective and robust methods for IDS-IoT. Moreover, we have proposed using an enhanced LSTM based on the Deep Learning approach, which has not been used earlier, as deep learning approaches have their benefits in classification problems.

3.1.1. KDD-Cup'99 dataset

The KDD-Cup'99 dataset[1] was collected in 1999 for the Third International KDD Tools Competition (KDD-Cup) to assess IDSs. It comprises around 4.9 million records, making it one of the largest datasets available for intrusion detection. It includes 41 attributes such as connection type, duration, destination IP addresses, and various flags and counts related to network packets. The dataset comprises multiple attack types: DoS, R2L, and U2R. The dataset has been criticized for its imbalance, redundancy, and outdated nature, leading to improved datasets such as NSL-KDD.

3.1.2. NSL-KDD dataset

The NSL-KDD dataset[2] was developed to resolve the limits and challenges of the original KDD-Cup'99 dataset. It contains 1.8 million records, covering wide-ranging network activities. The NSL-KDD dataset shares attributes identical to those of the KDD-Cup'99 with more features to enhance realism and mitigate issues in the original dataset. NSL-KDD is similar to KDD-Cup'99 and includes different type of attacks, categorized into DoS, Probe, R2L, and U2R classes. NSL-KDD aims to provide a more balanced distribution of classes and reduce redundancy compared to the KDD-Cup'99 dataset.

3.1.3. UNSW-NB15 dataset

The UNSW-NB15 dataset[3] solves the constraints of preceding datasets, providing a more current and accurate picture of network traffic. The collection includes around 2.5 million records, providing ample data for research. The UNSW-NB15 dataset's 49 features cover an extensive range of data about source and destination IP addresses, service categories, and flags. The dataset contains ordinary traffic and several attack methods, like DoS, Probe, Remote-to-Local, and user-to-root. UNSW-NB15 improved network traffic realism and balanced normal and attack scenarios better.

3.1.4. CICIoT2023 dataset

The CICIoT2023 dataset's[4] main goal is to offer a thorough and accurate dataset for researching intrusion detections in IoT

[1] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[2] http://www.unb.ca/cic/datasets/nsl.html
[3] http://doi.org/ 10.1109/MilCIS.2015.7348942
[4] https://doi.org/10.3390/s23135941

Table 2
Performance comparison summary of all four datasets

Dataset	Year of release	Volume	Total no. of features	No. of features used
KDD-Cup'99	1999	4,900,000	41	41
NSL-KDD	2009	4,898,431	41	41
UNSW-NB15	2021	2,57,673	49	49
CICIoT2023	2023	2,540,044	47	47

environments, which is meant to help practitioners and researchers create and assess security solutions for Internet of Things systems. At the University of New Brunswick in Canada, the Faculty of Computer Science has produced a dataset. A significant amount of data from an actual IoT network was collected for CICIoT2023, guaranteeing its validity and applicability to actual situations. A thorough examination of intrusion detection techniques in IoT is made possible by the dataset's inclusion of characteristics of IoT devices, network traffic, and attack patterns. The dataset includes attributes related to IoT devices, network traffic, and attack patterns, enabling a detailed analysis of intrusion detection techniques in IoT. A comparative summary table illustrating the key parameters, characteristics, and the year of release of all four datasets is given in Table 2.

3.2. Enhanced long short-term memory for IDS-IoT

The RNN subset LSTM aims to use the dependencies in sequential data precisely. Time-series data, such as network traffic, is effective. LSTM cells include mechanisms for memorizing and forgetting information over extended sequences to manage the context and temporal patterns. When used to detect intrusions, LSTM can evaluate the sequences of network activities and learn to identify unusual behaviors that may span numerous time steps, leading to the development of more resilient and competent IDSs. LSTMs introduce a sophisticated structure comprising many gates that regulate information flow. LSTMs use three types of gates to regulate the information flow:

1) Forget Gate: decides which cell state data should be removed.

$$\sigma(W_f[h_{t-1}, x_t] + b_f) \tag{1}$$

2) Input Gate: choose which input values to use to update the state of the cell.

$$\sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2}$$

3) Output Gate: Determines the subsequent concealed state, considering the input and cell states.

$$\sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3}$$

Where:

σ is the sigmoid function, W_x Weight for respective gate(x) neurons, h_(t-1) Output of the preceding LSTM block, x_t Input at the current timestamp, and b_x biases for the respective gates.

An LSTM cell's main parts are as follows:

Cell State (c_t): The network's memory can store data across various time steps.

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \hat{c}_t \tag{4}$$

Hidden state (h_t): The LSTM cell's output at a particular time step serves as the input for the following time step.

$$h_t = o_t \cdot \tan h(c_t) \tag{5}$$

Table 3
Confusion matrix

	Actual	
Predicted	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Table 4
Performance assessment matrices

Matric	Formula
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F-Measure	$\frac{2*Precision*Recall}{Precision+Recall}$
AUC	$\left(Recall - \frac{FP}{FP+TN} + 1\right)/2$

Table 5
Model evaluation metrics - KDD-Cup'99 dataset

	ML models		**DL models**		
Metric	**Logistic Regression**	**Adaboost**	**DNN**	**RNN**	**E-LSTM**
Accuracy	0.9983	0.7858	0.9991	0.9991	0.9994
Precision	0.9982	0.6783	0.9988	0.9989	0.9992
Recall	0.9983	0.7858	0.9989	0.9990	0.9993
F1-score	0.9982	0.7143	0.9988	0.9989	0.9992

$$\hat{c}_t = f_t \cdot c_{t-1} + i_t \cdot \hat{c}_t \quad (6)$$

Where, c_t is the cell (memory) state at timestamp (t); $\hat{c}_t$ is the condition for the cell state, and others are the same as above.

In the proposed E-LSTM model for IoT-based IDS, we used two LSTM layers to extract insights from the datasets. The mined features were fed into a flatten layer, and a dense layer with a SoftMax activation function is used to classify intrusions. The proposed model employs Adam optimizer with a batch size 64 and was trained up to 20 epochs. The results and analysis section elaborates on the proposed enhanced LSTM model for different datasets. The proposed E-LSTM model is improved and modified by tuning the number of units, learning rate, dropout rate, and batch size to improve performance for an IoT-based IDS.

4. Experiments

4.1. Data pre-processing and cleaning

The pre-processing steps were designed to clean, transform, and enrich the raw data. Addressing missing values, handling outliers, and applying the necessary transformations. Tools such as Pandas, NumPy, and regular expressions for data manipulation. Data pre-processing and cleaning are necessary steps in the data "KDD-Cup'99, NSL-KDD, UNSW-NB15, and CICIoT2023" that ensure the collected

Figure 2
E-LSTM loss & accuracy graph on KDD-Cup'99

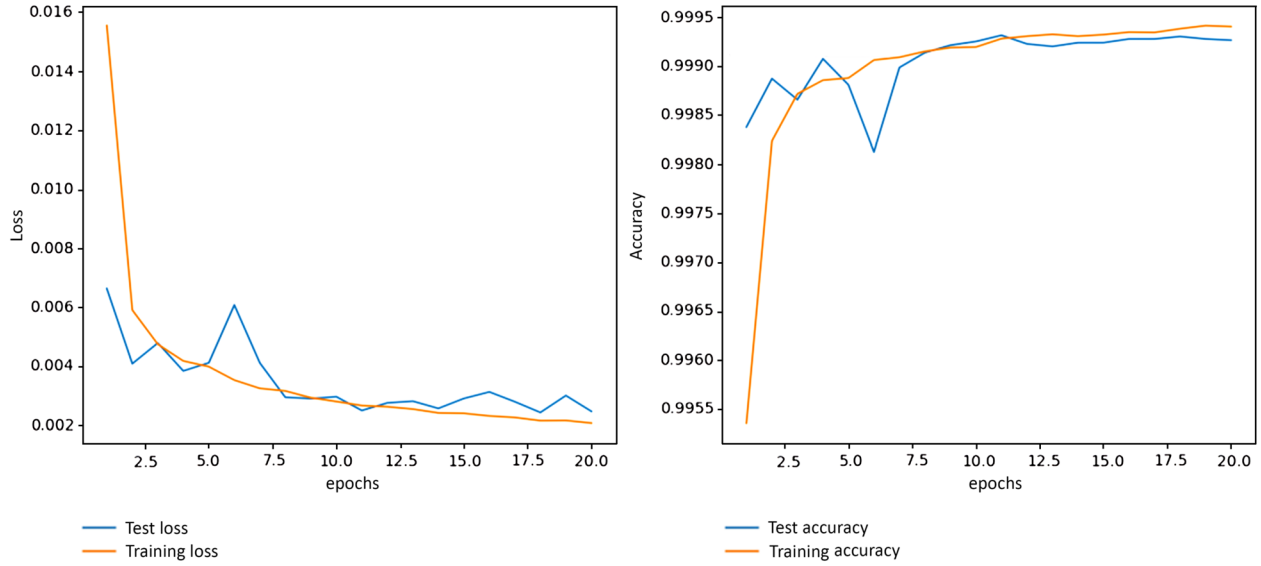


Figure 3
DNN loss & accuracy graph on KDD-Cup'99

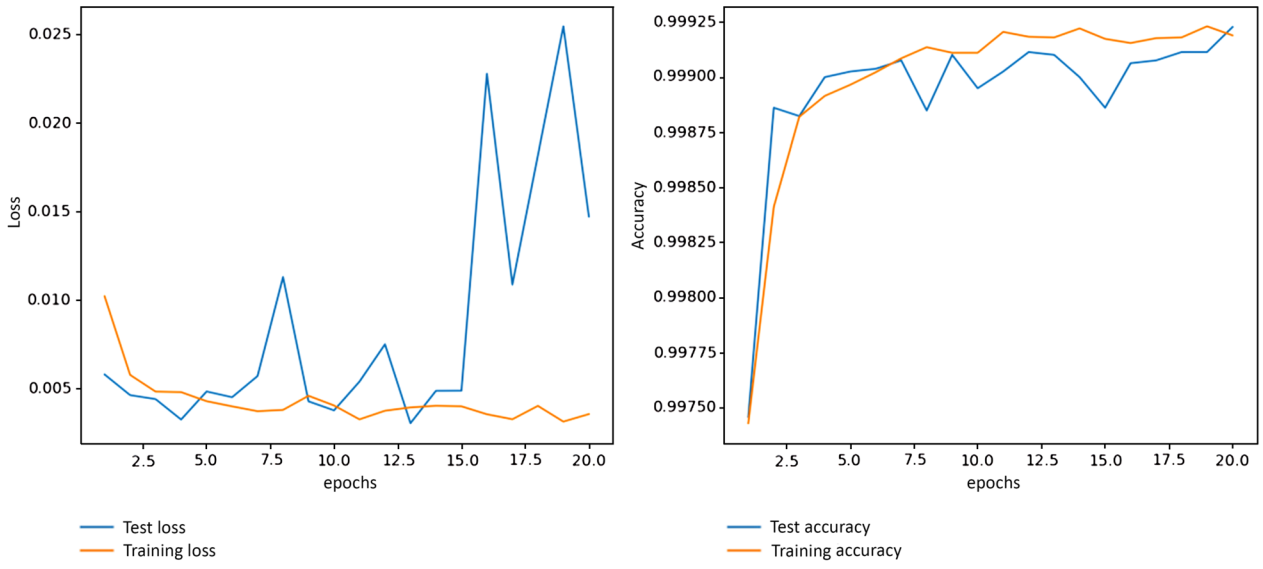


Figure 4
RNN loss & accuracy graph on KDD-Cup'99

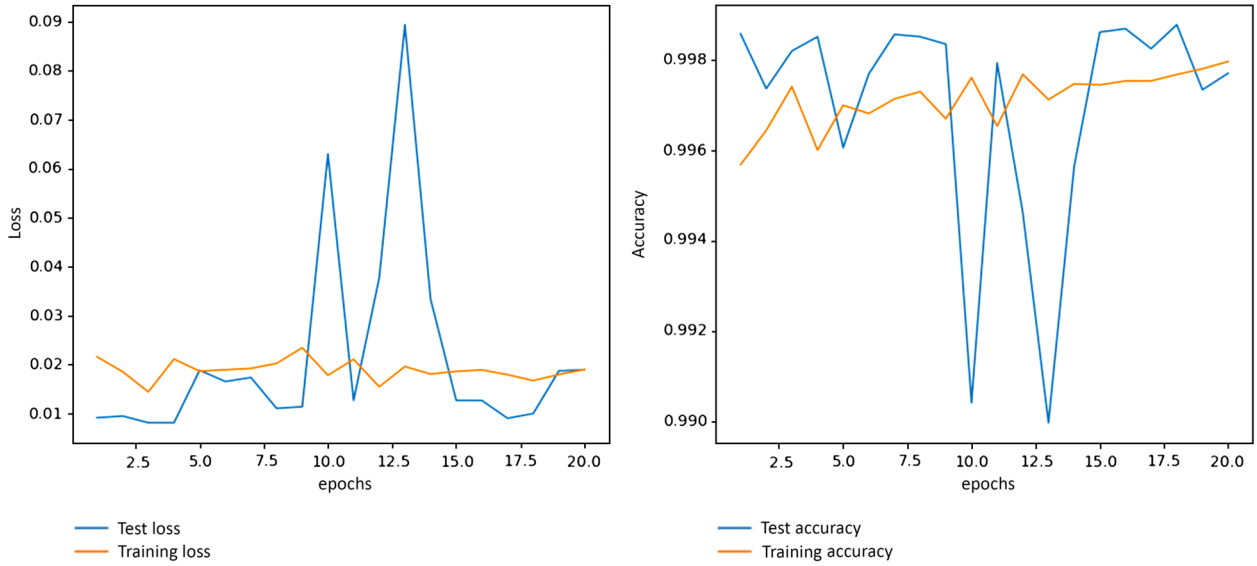
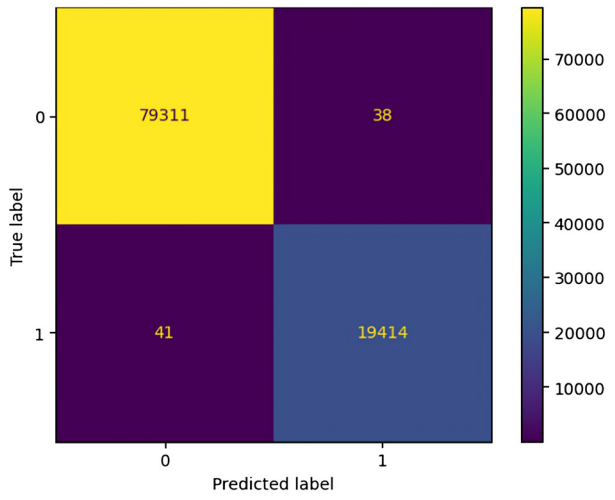


Figure 5
Summary of E-LSTM architecture

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 38, 100)           40800

 lstm_1 (LSTM)               (None, 38, 100)           80400

 flatten (Flatten)           (None, 3800)              0

 dense (Dense)               (None, 2)                 7602

=================================================================
Total params: 128,802
Trainable params: 128,802
Non-trainable params: 0
_________________________________________________________________
```

Figure 6
Confusion matrix for the KDD-Cup'99



data are accurate, consistent, and ready for analysis or modeling. These steps involve identifying and handling missing values, outliers, inconsistencies, and other quality issues. Here is a detailed breakdown of data preprocessing and cleaning.

4.1.1. Missing values handling

Missing values may result from incomplete data collection or data entry errors. Common strategies for handling missing values include:

1) Deletion: Rows or columns with significant missing values were removed if they did not contain critical information.
2) Imputation: Fill missing values with estimated values using techniques such as the mean, median, mode, or interpolation.
3) Advanced Imputation: To fill in missing variables, use more sophisticated techniques like regression imputation, k-nearest neighbors, or predictive modeling.

4.1.2. Outlier detection and handling

Data points that considerably deviate from the rest of the dataset are referred to as outliers, which may distort results from modeling and analysis. The following are some strategies for handling outliers.

1) Identification: Identify outliers utilizing statistical techniques such as z-scores, Interquartile Range (IQR), or domain knowledge.
2) Transformation: Logarithmic or Box-Cox transformations are applied to reduce the impact of outliers.
3) Capping: Set a threshold and cap the extreme values at that threshold to mitigate their effects.
4) Removal: Sometimes, outliers can be removed if deemed erroneous or irrelevant.

4.1.3. Data type conversion

Ensure that data types are consistent and appropriate for analysis or modeling:

1) Categorical to Numerical: Use methods like label encoding to transform categorical information into a numerical representation.

4.1.4. Normalization

Our study uses the StandardScaler technique to normalize the data.

Table 6
Model evaluation metrics - NSL-KDD dataset

	ML models		DL models		
Metric	**Logistic Regression**	**Adaboost**	**DNN**	**RNN**	**E-LSTM**
Accuracy	0.9515	0.8126	0.9803	0.9773	0.9816
Precision	0.9456	0.6796	0.9775	0.9748	0.9756
Recall	0.9515	0.8127	0.9795	0.9764	0.9793
F1-score	0.9472	0.7397	0.9778	0.9748	0.9767

Figure 7
E-LSTM loss & accuracy graph of NSL-KDD

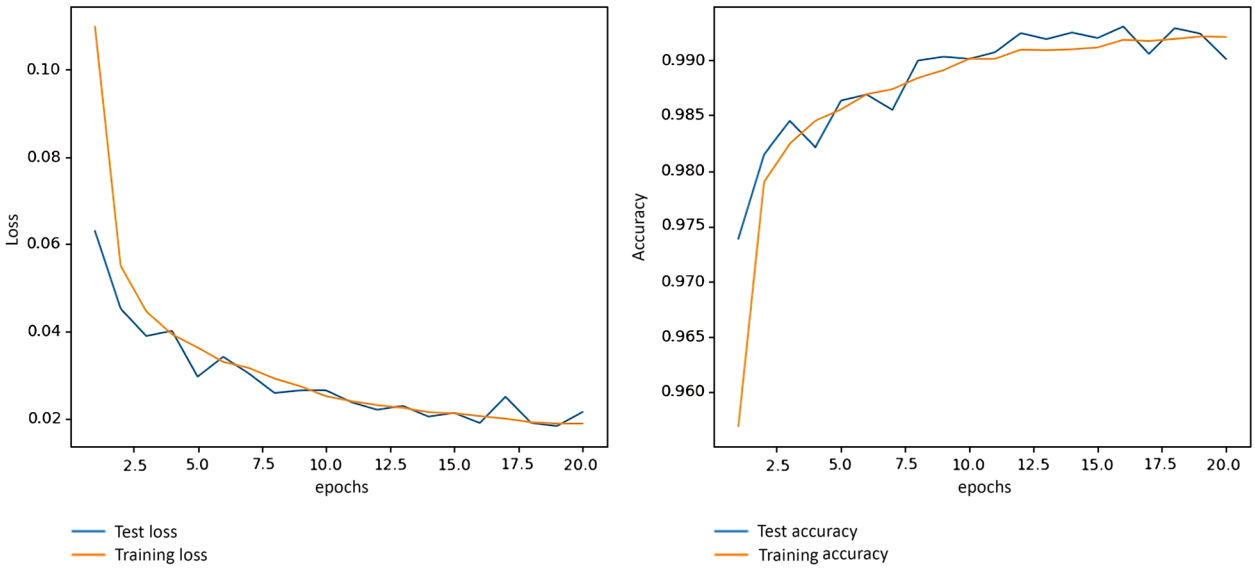


Figure 8
DNN loss & accuracy vs epochs graph of NSL-KDD

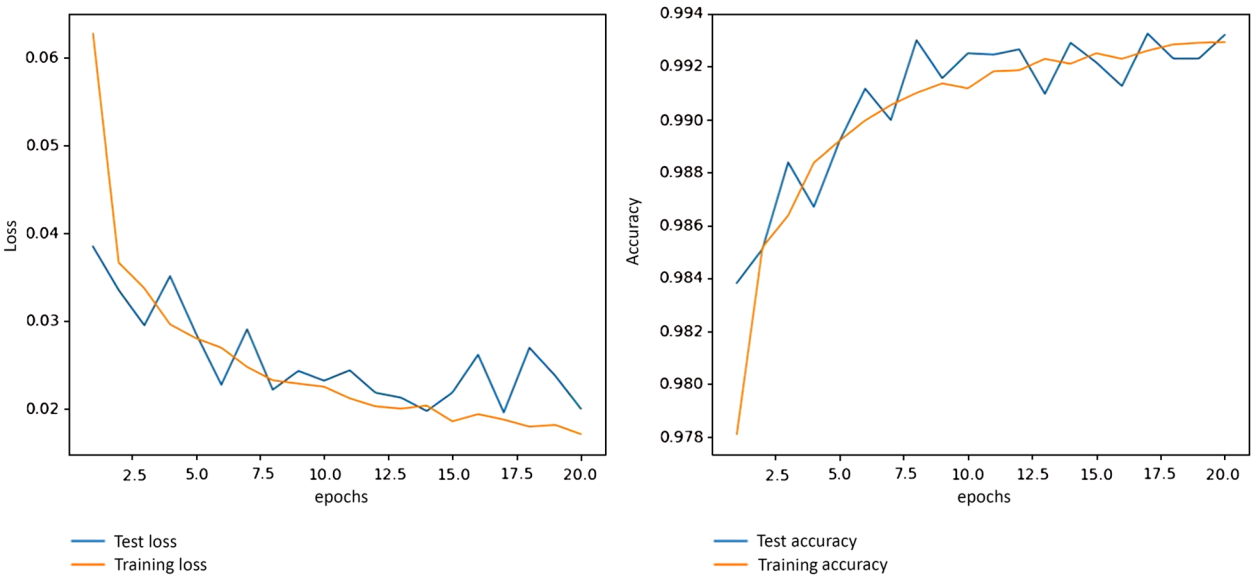
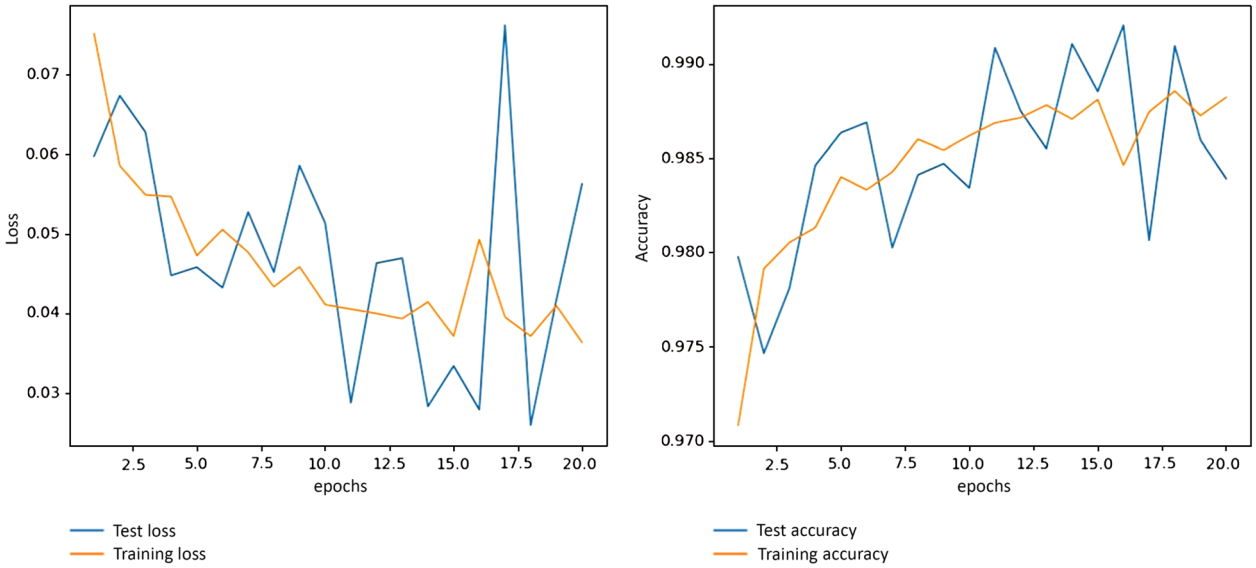


Figure 9
RNN loss & accuracy graph of NSL-KDD



4.2. Performance evaluation

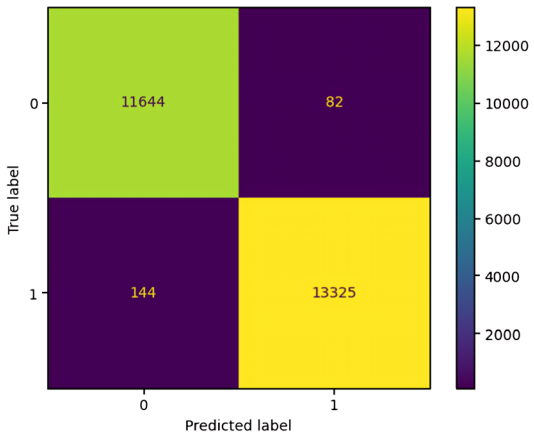
The efficacy of many classifiers was assessed with the major key performance metrics. The confusion matrix measures parameter values are presented in Table 3. The performance evaluation was conducted by considering these estimates. Metrics such as F-measure, area under the curve (AUC), precision, recall, and accuracy are provided in Table 4.

4.3. Experimental setup

Python (version 3.10.7) on Windows 11 was used in this study. The backends for the deep learning models were TensorFlow and the Keras framework (version 2.10.0). The high-performance computing environment included an Intel(R) Xenon(R) W-2255 CPU running at 3.70 GHz, 64 GB of RAM, an NVIDIA GEFORCE RTX A4000 GPU with 16 GB of RAM, and DirectX-12 (version 12.1).

Initially, to develop the model, an LSTM layer, a flatten layer, and a dense layer were taken with Adam optimizer and SoftMax activation function at the classification phase. The model was initially trained up to 10 epochs. Finally, looking at the performance of various candidate models, the model with two LSTM layer, one flatten layer, and one dense layer, with Adam optimizer and SoftMax activation function at the classification phase, was selected as the best model. It was trained up to 20 epochs. Consequently, the proposed LSTM-based model results are studied and assessed compared to the other models.

Figure 10
Confusion matrix for the NSL-KDD dataset



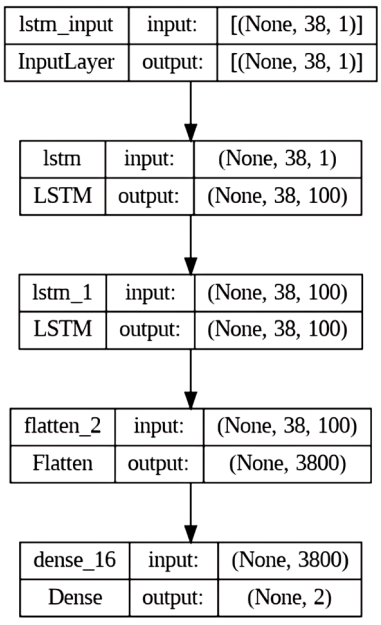
4.4. Results and discussion

Four different datasets were used to test a few ML and DL models: CIC IoT 2023, UNSW-NB15, KDD-Cup'99, and NSL-KDD. Tables give a summary of the Model Evaluation Metrics applied to these datasets.

4.4.1. Experiments on intrusion detection with KDD-CUP'99 dataset

Using a confusion matrix for assessment, the performance of several ML and DL algorithms was fully examined with the KDD-

Figure 11
Summary of proposed enhanced LSTM architecture



Cup'99 dataset. The results, such as F1-score, Precision, and Recall, provide information on how well the models identify and categorize intrusions.

The model's performance in Table 5 shows noteworthy patterns brought to light through confusion matrix analysis. The high recall, precision, and F1-score values demonstrate that the Logistic Regression and deep learning algorithms accurately recognize genuine positives and true negatives. With F1 scores over 0.999, the enhanced LSTM (E-LSTM) model regularly outperforms the other models, demonstrating their resilience in identifying intrusion cases, highlighting the

Table 7
Model evaluation metrics - UNSW-NB15 dataset

	ML models		DL models		
Metric	**Logistic Regression**	**Adaboost**	**DNN**	**RNN**	**E-LSTM**
Accuracy	0.8936	0.9613	0.9758	0.9694	0.9777
Precision	0.8951	0.96136	0.9746	0.9689	0.9771
Recall	0.8936	0.96138	0.9744	0.9686	0.9770
F1-score	0.8916	0.96137	0.9745	0.9687	0.9771

Figure 12
E-LSTM loss & accuracy graph of UNSW-NB15

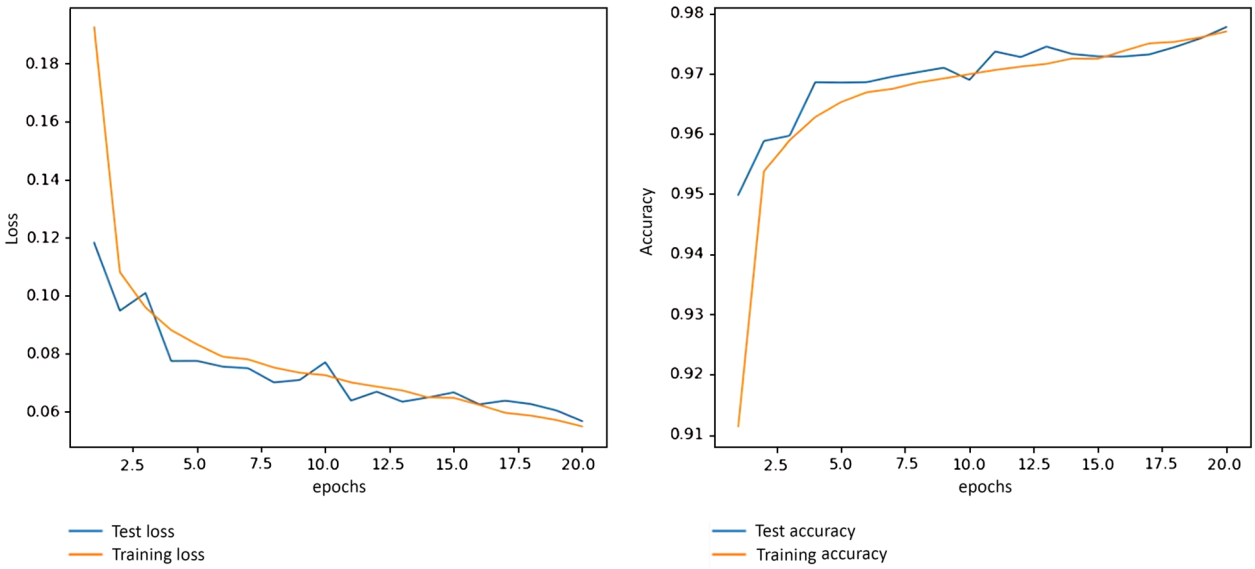


Figure 13
DNN loss & accuracy graph of UNSW-NB15

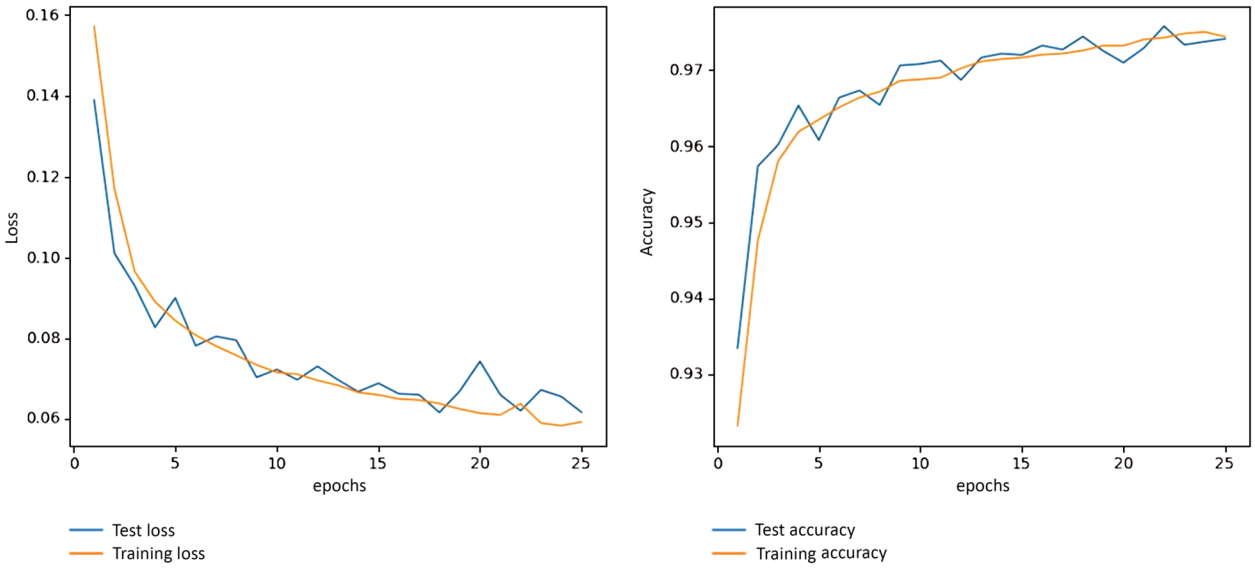
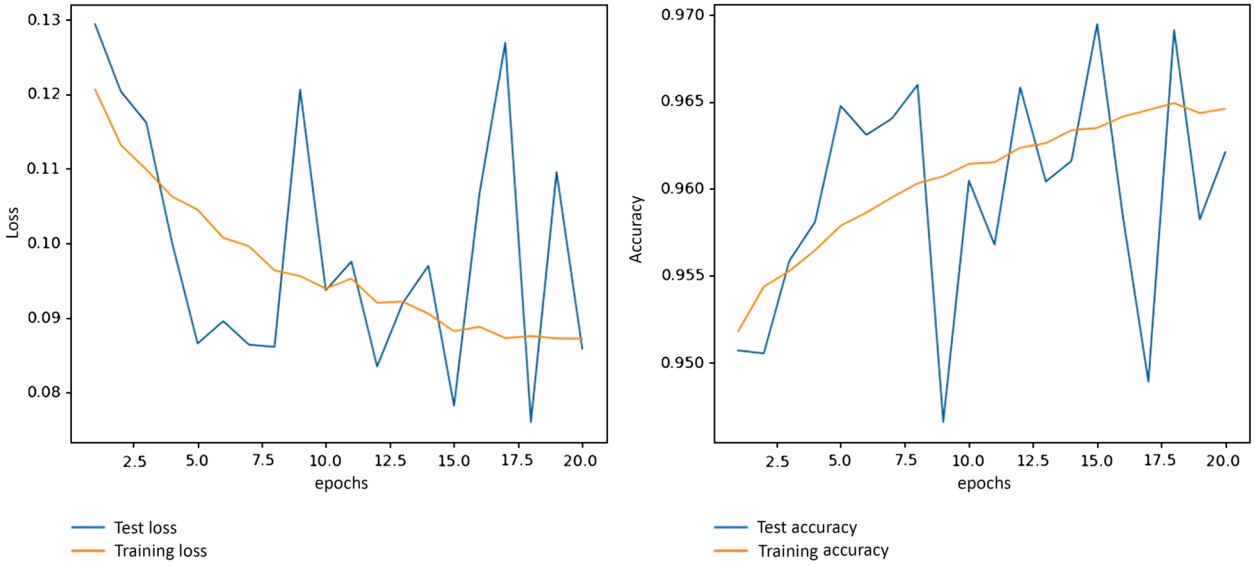


Figure 14
RNN loss & accuracy graph of UNSW-NB15



possibility of adjusting the model parameters to improve performance further. Figures 2, 3, and 4 display the loss and accuracy of the LSTM, DNN, and RNN deep-learning models, respectively. The parameter configuration specification presented in Figure 5 includes the layers and their corresponding positions in the LSTM structural design, output format, parameters or weights for each layer, and the all parameters in the model. In addition, the specifications include the output format. Figure 6 visually represents the predictive outcomes in the confusion matrix.

4.4.2. Experiments on intrusion detection using NSL-KDD dataset

The NSL-KDD dataset was utilized for assessing the performance of several deep and ML models. A performance matrix and various metrics were utilized to assess their efficiency in detecting intrusions.

Logistic regression and DL algorithms consistently exhibited remarkable accuracy, recall, and F1-score values, as presented by the performance matrix in Table 6. With F1 scores over 0.97, the enhanced LSTM model regularly outperforms the other models, demonstrating their resilience in identifying intrusion cases and highlighting their competence in correctly recognizing them. Figures 7, 8, and 10 display the loss and accuracy of the E-LSTM, DNN, and RNN DL models, respectively. Figure 10 visually represents the predictive outcomes in the confusion matrix.

Both loss and accuracy graphs present a detailed outline of the learning processes of the models and their performances on the datasets. These visualizations support earlier findings and conclusions drawn from quantitative metrics, reinforcing the efficacy of the selected models in intrusion detection tasks across different datasets. The parameter configuration specification presented in Figure 11 includes the layers and their corresponding positions in the LSTM architecture, output format, parameters or weights for every layer, and all the parameters in the model. In addition, the specifications include the output format.

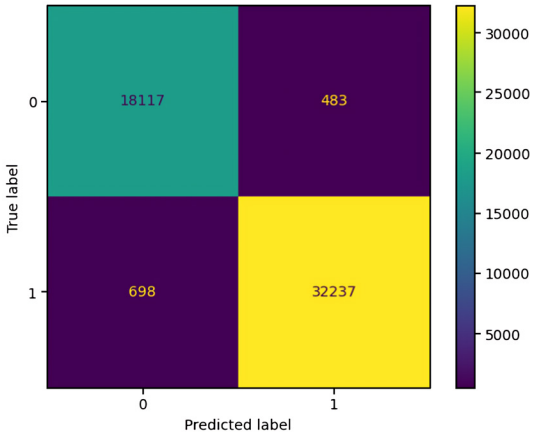
4.4.3. Experiments on intrusion detection using UNSW-NB15

The efficiency of many ML and DL models in intrusion detection was evaluated with UNSW-NB15 dataset. The study included various metric as shown in Table 7, to fully assess model's performance.

Figure 15
Summary of enhanced LSTM architecture

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 40, 100)           40800

 lstm_1 (LSTM)               (None, 40, 100)           80400

 flatten_1 (Flatten)         (None, 4000)              0

 dense_53 (Dense)            (None, 2)                 8002

=================================================================
Total params: 129,202
Trainable params: 129,202
Non-trainable params: 0
_________________________________________________________________
```

Figure 16
Confusion matrix of UNSW-NB15



The performance matrix analysis highlighted significant patterns in the performance of the various models. Across the board, the Logistic Regression and DL models exhibited consistent precision,

Table 8
Model evaluation metrics – CICIoT 2023 dataset

	ML models		DL models		
Metric	**Logistic Regression**	**Adaboost**	**DNN**	**RNN**	**E-LSTM**
Accuracy	0.8013	0.6679	0.9826	0.9825	0.9858
Precision	0.8073	0.6609	0.9798	0.9807	0.9845
Recall	0.8085	0.6679	0.9828	0.9817	0.9854
F1-score	0.7664	0.6070	0.9808	0.9808	0.9845

Figure 17
E-LSTM loss & accuracy graph – CICIoT 2023

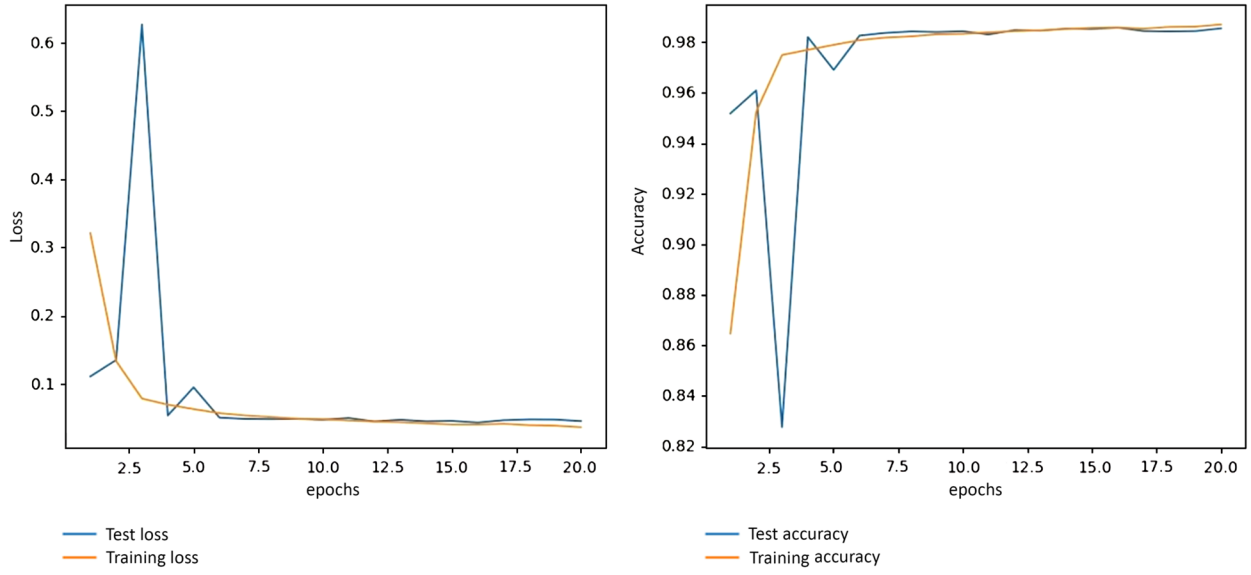


Figure 18
DNN loss & accuracy graph – CICIoT 2023

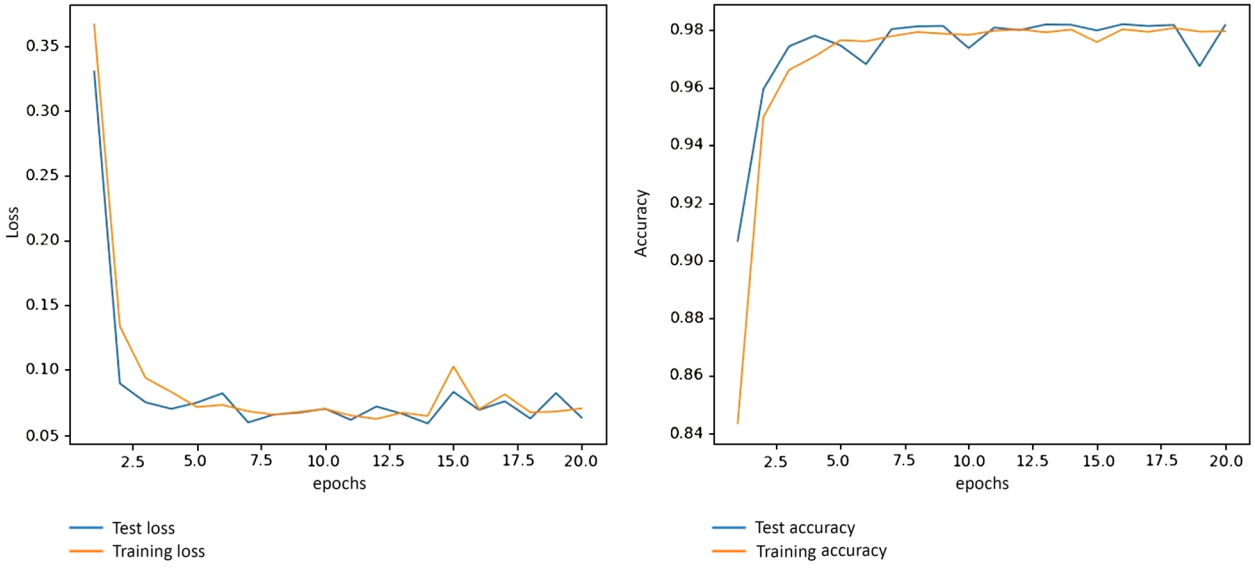
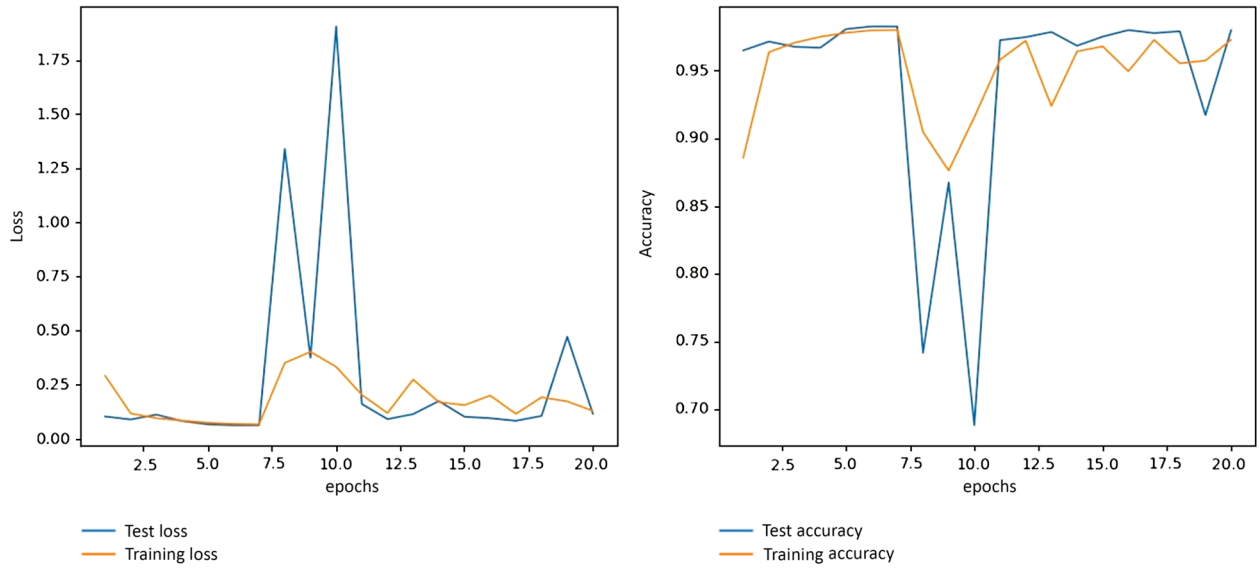


Figure 19
RNN loss & accuracy graph – CICIoT 2023



recall, and F1-score values, suggesting their robustness in accurately identifying intrusion instances. With F1 scores over 0.97, the enhanced LSTM model regularly outperforms the other models, demonstrating their resilience in identifying intrusion cases, highlighting their competence in correctly recognizing them, which indicates their proficiency in differentiating among normal and intrusive network traffic, reinforcing their suitability for real-world intrusion detection scenarios. Figures 12, 13, and 14 present the loss and accuracy of the E-LSTM, DNN, and RNN deep learning models, respectively. The parameter configuration specification presented in Figure 15 includes the layers and their corresponding positions in the E-LSTM structural design, output format, parameters or weights for every layer, and the overall parameters in the model. In addition, the specifications include the output format. Figure 16 visually represents the predictive outcomes in the confusion matrix.

The loss and accuracy graph visualizations provided a comprehensive view of the learning trajectories of the models and their success on the datasets. These graphical depictions support the preceding findings and judgments drawn from the quantitative measurements, thereby supporting the efficacy of the selected models in detecting the intrusion across various datasets.

4.4.4. Experiments on intrusion detection using CICIoT 2023dataset

Table 8 displays various assessment criteria for DL and ML methods functional to diverse datasets. Precision, Recall, and F1-score are among the examined measures, providing a complete understanding of model efficacy across 34 classes. With F1 scores over 0.98, the enhanced LSTM model regularly outperforms the other models, demonstrating their resilience in identifying intrusion cases highlighting their competence in correctly recognizing intrusion cases This indicates their proficiency in differentiating among normal and intrusive network traffic, reinforcing their suitability for real-world intrusion detection scenarios Figures 17, 18, and 19 show the loss and accuracy of deep learning models E-LSTM, DNN and RNN.

The loss and accuracy graphs clearly show the learning procedures and the models' performance on various datasets. These graphics support the efficacy of the chosen models in intrusion detection tasks. The output format, parameters, weights for each layer, the layers

Figure 20
Summary of enhanced LSTM architecture

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_2 (LSTM)               (None, 46, 100)           40800

 lstm_3 (LSTM)               (None, 46, 100)           80400

 flatten_1 (Flatten)         (None, 4600)              0

 dense_1 (Dense)             (None, 34)                156434

=================================================================
Total params: 277,634
Trainable params: 277,634
Non-trainable params: 0
_________________________________________________________________
```

and their respective locations in the LSTM structural design, and the model's overall parameters are specified in Fig. 20. Figure 21 visually represents the predictive outcomes in the confusion matrix.

4.4.5. Performance comparison

The results highlight notable variations in the model performance across the datasets, as shown in Table 9. Logistic regression consistently yields competitive accuracy across all datasets, whereas our proposed enhanced LSTM model demonstrates remarkable accuracy and robustness. Deep-learning models, particularly the proposed E-LSTM, exhibit impressive accuracy on the CICIoT 2023 dataset, emphasizing their capability to capture intricate patterns in IoT network traffic. The findings underscore the significance of choosing models aligned with the dataset features. Dataset intricacies influence the model performance, and certain models are better suited for specific datasets. Figure 22 displays a bar graph to compare the performance of the numerous specified models with their datasets.

Table 8 illustrates how the suggested enhanced LSTM methods fared better on the KDD-Cup'99, NSL-KDD, USNW-NB15, and CICIoT 2023 datasets. Table 10 demonstrates the performance assessment of the proposed model with prevailing work [47–58]. It displays how remarkable the "Proposed Method" is, specifically

Figure 21
Confusion matrix of UNSW-NB15

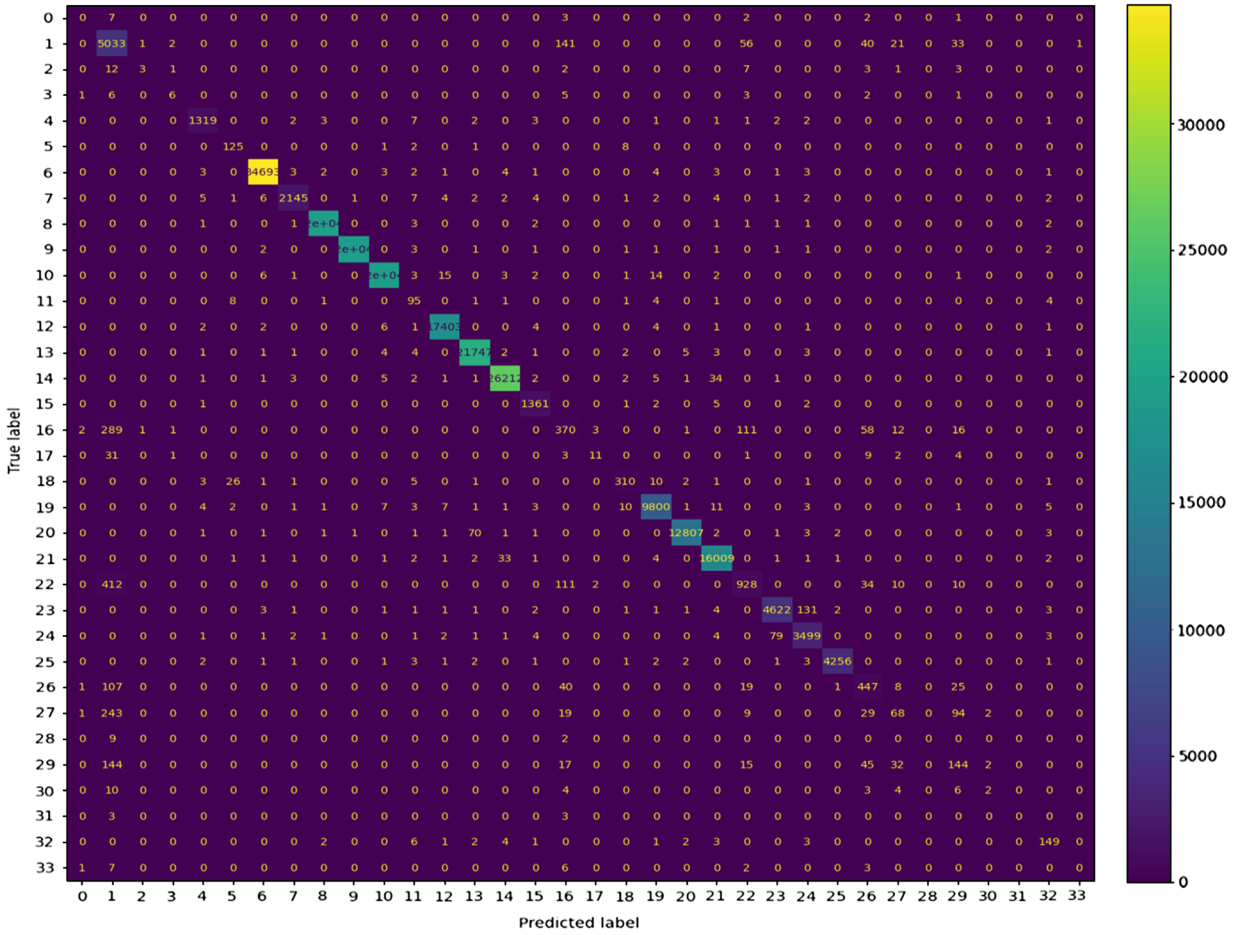


Table 9
Accuracy table of all datasets utilized

	ML models accuracy		DL accuracy		
Datasets	**Logistic Regression**	**Adaboost**	**DNN**	**RNN**	**E-LSTM**
CIC IoT 2023	0.8013	0.6679	0.9826	0.9827	**0.9856**
UNSW-NB15	0.8936	0.9613	0.9758	0.9694	**0.9777**
NSL-KDD	0.9515	0.8127	0.9803	0.9773	**0.9816**
KDD_CUP'99	0.9983	0.7858	0.9991	0.9991	**0.9994**

advantageous when dealing with limited or imbalanced data, common challenges in intrusion detection, and 99.94% accuracy with Enhanced LSTM. It is specifically designed to capture patterns and dependencies in sequences, making it well-suited for analyzing the temporal behavior of network traffic data. Unlike CNNs and DNNs, which may overlook temporal relationships, the proposed E-LSTM can effectively model the context of previous network events, which is essential for identifying complex intrusion patterns. In addition, its recurrent architecture enables real-time analysis of streaming data, making it suitable for network intrusion detection systems that must make rapid decisions as new data arrives, a challenge for methods such as the PSOM or Naive Bayes.

Similarly, the suggested approach obtained an accuracy of 99.16% with E-LSTM on the NSL-KDD dataset. With E-LSTM, the performance for the UNSW-NB15 dataset remained comparatively high at 98.33%. Using enhanced LSTM, the suggested approach obtained 98.56% accuracy even in the large and complex CICIoT 2023 dataset.

5. Conclusion and Future Work

To complete this study, we examined various datasets related to IoT intrusion detection. This group contains several renowned datasets for detecting intrusion, including UNSW-NB15, KDD-Cup'99, NSL-KDD, and CICIoT2023. Our research gathered data to better understand

Figure 22
Accuracy graph of KDD-Cup'99, NSL-KDD, UNSW_NB15 and CICIoT 2023

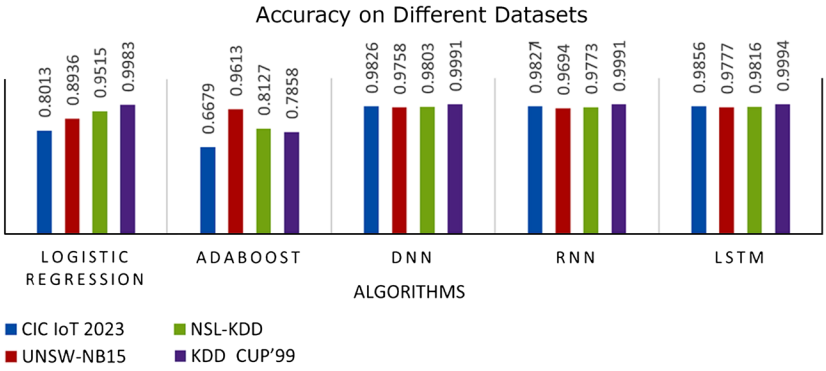


Table 10
Comparison of the proposed model with existing work

Proposed by	Years	Dataset	Feature ex-traction used	Method	Accuracy
S. Mukherjee, et al. [47]	2012	NSL-KDD	Vitality-based	Naive Bayes	97.78
R. M. Elbasiony, et al. [48]	2013	KDD-Cup'99	–	Random Forests + Weighted K-means	98.3
E. D. L. Hoz, et al. [49]	2015	KDD-Cup'99	PCA-FDR	PSOM	88.0%
Farnaaz, N., et al. [50]	2016	NSL-KDD	Symmetrical uncertainty	Random Forest	99.67
Belavagi et al. [51]	2016	NSL-KDD	–	RF, LR, SVM, and Naive Bayes	99% with RF
Thaseen, I. S., et al. [52]	2017	NSL-KDD	Chi-square	SVM	98
Meena, G., and Choudhary [44]	2017	KDD-Cup'99, NSL-KDD	–	Naive Bayes	92.71
S. Ahmad, F. Arif, Z. Zabee-hullah [54]	2020	KDD-Cup'99	relus, SoftMax	DNN	99.91
A. R. Gupta, et al. [55]	2020	NSL-KDD	–	CNN	Not reported
R. Abou Khamis, et al. [56]	2020	UNSW-NB15	PCA	DNN	92% without PCA 93% with PCA
H. Hindy, et al. [57]	2020	NSL-KDD	–	Autoencoder	92.96
Euclides Carlos Pinto Neto, et al. [58]	2023	CICIot 2023	Features ex-tracted from the original Pcap files	Random Forest	99.16
Proposed Method	2024	KDD-Cup'99	relu, SoftMax	Enhanced LSTM	99.94
Proposed Method	2024	NSL-KDD	–	Enhanced LSTM	99.16
Proposed Method	2024	UNSW-NB15	–	Enhanced LSTM	98.33
Proposed Method	2024	CICIoT 2023	Using Pyspark	Enhanced LSTM	99.38

how well different machine learning models can detect and mitigate security risks in IoT environments. The models' performances, including Logistic Regression, AdaBoost, DNNs, E-LSTM, and RNNs—all different forms of neural networks—were evaluated systematically. Comparative analysis of these datasets highlights the need for tailored approaches to address the unique characteristics of each dataset, which highlights the challenges and opportunities associated with IoT intrusion detection.

These results demonstrate how consistent our suggested approach is. Across many datasets, accurate findings were obtained, which indicates that the suggested approach is a viable option for intrusion-detection jobs as it is robust and dependable for precisely identifying incursions in various scenarios.

6. Limitations

In our study, we focused on evaluating the performance of a range of intrusion detection techniques across all four datasets to understand their adaptability and limitations in the context of IoT security. It may be further improved using Ensemble Methods, Feature Engineering, and Explainability Feature to enhance the proposed model's efficacy, effectiveness, and adaptability.

7. Future Directions

Although this study presents valuable insights for IoT based intrusion detection using diverse datasets, several avenues for future research can enhance our understanding and contribute to real-world applications.

1) Ensemble methods: The potential of ensemble approaches to combine the strengths of various techniques and models to enhance the overall intrusion detection performance. Methods, such as stacking and boosting, can lead to enhanced accuracy and robustness.
2) Feature engineering: Explore advanced feature engineering techniques that capture the domain-specific characteristics of IoT network traffic. Leveraging domain knowledge can lead to more informative features, boosting model performance.
3) Explainability: Focus on creating a system that can be understood and models that can be explained to give decision-makers insights and improve the credibility of intrusion detection systems.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in KDD-Cup'99 dataset at http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, NSL-KDD dataset at http://www.unb.ca/cic/datasets/nsl.html, UNSW-NB15 dataset at http://doi.org/10.1109/MilCIS.2015.7348942, and CICIoT2023 dataset at https://doi.org/10.3390/s23135941.

Author Contribution Statement

Gaurav Meena: Conceptualization, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Supervision, Project administration. **Ajay Indian:** Methodology, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization.

References

[1] Andoni, M., Robu, V., Flynn, D., Abram, S., Geach, D., Jenkins, D., ... & Peacock, A. (2019). Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renewable and Sustainable Energy Reviews*, *100*, 143–174. https://doi.org/10.1016/j.rser.2018.10.014

[2] Lone, K., & Sofi, S. A. (2023). A review on offloading in fog-based Internet of Things: Architecture, machine learning approaches, and open issues. *High-Confidence Computing*, *3*(2), 100124. https://doi.org/10.1016/j.hcc.2023.100124

[3] Rahman, S. A., Tout, H., Talhi, C., & Mourad, A. (2020). Internet of things intrusion detection: Centralized, on-device, or federated learning. *IEEE Network*, *34*(6), 310–317. https://doi.org/10.1109/MNET.011.2000286

[4] Aouedi, O., Vu, T. H., Sacco, A., Nguyen, D. C., Piamrat, K., Marchetto, G., & Pham, Q. V. (2024). A survey on intelligent Internet of Things: Applications, security, privacy, and future directions. *IEEE Communications Surveys & Tutorials*. https://doi.org/10.1109/COMST.2024.3430368

[5] Qadir, Z., Le, K. N., Saeed, N., & Munawar, H. S. (2023). Towards 6G Internet of Things: Recent advances, use cases, and open challenges. *ICT Express*, *9*(3), 296–312. https://doi.org/10.1016/j.icte.2022.06.006

[6] Rath, K. C., Khang, A., & Roy, D. (2024). The role of Internet of Things (IoT) technology in Industry 4.0 economy. In A. Khang, V. Abdullayev, V. Hahanov, & V. Shah (Eds.), *Advanced IoT Technologies and applications in the industry 4.0 digital economy* (pp. 1–28). CRC Press.

[7] Venkatraman, S., & Surendiran, B. (2020). Adaptive hybrid intrusion detection system for crowd sourced multimedia internet of things systems. *Multimedia Tools and Applications*, *79*(5), 3993–4010. https://doi.org/10.1007/s11042-019-7495-6

[8] Molaei, R., Rahsepar Fard, K., & Bouyer, A. (2024). An improved influence maximization method for online advertising in social internet of things. *Big Data*, *12*(3), 173–190. https://doi.org/10.1089/big.2023.0042

[9] Dutta, M., & Granjal, J. (2020). Towards a secure Internet of Things: A comprehensive study of second line defense mechanisms. *IEEE Access*, *8*, 127272–127312. https://doi.org/10.1109/ACCESS.2020.3005643

[10] Simoglou, G., Violettas, G., Petridou, S., & Mamatas, L. (2021). Intrusion detection systems for RPL security: A comparative analysis. *Computers & Security*, *104*, 102219. https://doi.org/10.1016/j.cose.2021.102219

[11] Boyanapalli, A., & Shanthini, A. (2020). A comparative study of techniques, datasets and performances for intrusion detection systems in IoT. In *Artificial Intelligence Techniques for Advanced Computing Applications: Proceedings of ICACT 2020*, 225–236. https://doi.org/10.1007/978-981-15-5329-5_22

[12] Ramaiah, M., Chandrasekaran, V., Ravi, V., & Kumar, N. (2021). An intrusion detection system using optimized deep neural network architecture. *Transactions on Emerging Telecommunications Technologies*, *32*(4), e4221. https://doi.org/10.1002/ett.4221

[13] Ghobaei-Arani, M., Souri, A., & Rahmanian, A. A. (2020). Resource management approaches in fog computing: A comprehensive review. *Journal of Grid Computing*, *18*(1), 1–42. https://doi.org/10.1007/s10723-019-09491-1

[14] Souri, A., & Ghobaei-Arani, M. (2022). Cloud manufacturing service composition in IoT applications: A formal verification-based approach. *Multimedia Tools and Applications*, *81*(19), 26759–26778. https://doi.org/10.1007/s11042-021-10645-1

[15] Malhotra, P., Singh, Y., Anand, P., Bangotra, D. K., Singh, P. K., & Hong, W. C. (2021). Internet of things: Evolution, concerns and security challenges. *Sensors*, *21*(5), 1809. https://doi.org/10.3390/s21051809

[16] Affia, A. A. O., Finch, H., Jung, W., Samori, I. A., Potter, L., & Palmer, X. L. (2023). IoT health devices: Exploring security risks in the connected landscape. *IoT*, *4*(2), 150–182. https://doi.org/10.3390/iot4020009

[17] Soliman, S., Oudah, W., & Aljuhani, A. (2023). Deep learning-based intrusion detection approach for securing industrial Internet of Things. *Alexandria Engineering Journal*, *81*, 371–383. https://doi.org/10.1016/j.aej.2023.09.023

[18] Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., & Razaque, A. (2020). Deep recurrent neural network for IoT intrusion detection system. *Simulation Modelling Practice and Theory*, *101*, 102031. https://doi.org/10.1016/j.simpat.2019.102031

[19] Khan, M. A., & Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, *82*, 395–411. https://doi.org/10.1016/j.future.2017.11.022

[20] Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity*, *4*(1), 18. https://doi.org/10.1186/s42400-021-00077-7

[21] Djenna, A., Harous, S., & Saidouni, D. E. (2021). Internet of things meet internet of threats: New concern cyber security issues of critical cyber infrastructure. *Applied Sciences*, *11*(10), 4580. https://doi.org/10.3390/app11104580

[22] Liu, Q., Hagenmeyer, V., & Keller, H. B. (2021). A review of rule learning-based intrusion detection systems and their prospects in smart grids. *IEEE Access*, *9*, 57542–57564. https://doi.org/10.1109/ACCESS.2021.3071263

[23] Heidari, A., Toumaj, S., Navimipour, N. J., & Unal, M. (2022). A privacy-aware method for COVID-19 detection in chest CT images using lightweight deep conventional neural network and blockchain. *Computers in Biology and Medicine*, *145*, 105461. https://doi.org/10.1016/j.compbiomed.2022.105461

[24] Yahyaoui, A., Abdellatif, T., Yangui, S., & Attia, R. (2021). READ-IoT: Reliable event and anomaly detection framework for the Internet of Things. *IEEE Access*, *9*, 24168–24186. https://doi.org/10.1109/ACCESS.2021.3056149

[25] Liu, Z., Xu, B., Cheng, B., Hu, X., & Darbandi, M. (2022). Intrusion detection systems in the cloud computing: A comprehensive and deep literature review. *Concurrency and Computation: Practice and Experience*, *34*(4), e6646. https://doi.org/10.1002/cpe.6646

[26] Meng, W., Li, W., & Zhou, J. (2021). Enhancing the security of blockchain-based software defined networking through trust-based traffic fusion and filtration. *Information Fusion*, *70*, 60–71. https://doi.org/10.1016/j.inffus.2020.12.006

[27] Jabraeil Jamali, M. A., Bahrami, B., Heidari, A., Allahverdizadeh, P., & Norouzi, F. (2019). Some cases of smart use of the IoT. In *Towards the internet of things: Architectures, security, and applications*, 85–129. https://doi.org/10.1007/978-3-030-18468-1_4

[28] Kethineni, K., & Pradeepini, G. (2024). Intrusion detection in internet of things-based smart farming using hybrid deep learning framework. *Cluster Computing*, *27*(2), 1719–1732. https://doi.org/10.1007/s10586-023-04052-4

[29] Choudhary, R. R., Jangid, A., & Meena, G. (2017, September). A novel approach for edge detection for blurry images by using digital image processing. In *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication*, 1029–1034. https://doi.org/10.1109/CTCEEC.2017.8455081

[30] Walling, S., & Lodh, S. (2024). Network intrusion detection system for IoT security using machine learning and statistical based hybrid feature selection. *Security and Privacy*, *7*(6), e429. https://doi.org/10.1002/spy2.429

[31] Ali, A. H., Charfeddine, M., Ammar, B., Hamed, B. B., Albalwy, F., Alqarafi, A., & Hussain, A. (2024). Unveiling machine learning strategies and considerations in intrusion detection systems: A comprehensive survey. *Frontiers in Computer Science*, *6*, 1387354. https://doi.org/10.3389/fcomp.2024.1387354

[32] Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. A. (2020). Performance analysis of machine learning algorithms in intrusion detection system: A review. *Procedia Computer Science*, *171*, 1251–1260. https://doi.org/10.1016/j.procs.2020.04.133

[33] Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences*, *9*(20), 4396. https://doi.org/10.3390/app9204396

[34] Güney, H. (2023). Preprocessing impact analysis for machine learning-based network intrusion detection. *Sakarya University Journal of Computer and Information Sciences*, *6*(1), 67–79.

[35] Kalita, D. J., Singh, V. P., & Kumar, V. (2023). A novel adaptive optimization framework for SVM hyper-parameters tuning in non-stationary environment: A case study on intrusion detection system. *Expert Systems with Applications*, *213*, 119189. https://doi.org/10.1016/j.eswa.2022.119189

[36] Pervez, M. S., & Farid, D. M. (2014, December). Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs. In the *8th International Conference on Software, Knowledge, Information Management and Applications*, 1–6. https://doi.org/10.1109/SKIMA.2014.7083539

[37] Belouch, M., El Hadaj, S., & Idhammad, M. (2018). Performance evaluation of intrusion detection based on machine learning using Apache Spark. *Procedia Computer Science*, *127*, 1–6. https://doi.org/10.1016/j.procs.2018.01.091

[38] Jeffrey, N., Tan, Q., & Villar, J. R. (2023). A review of anomaly detection strategies to detect threats to cyber-physical systems. *Electronics*, *12*(15), 3283. https://doi.org/10.3390/electronics12153283

[39] Dahou, A., Abd Elaziz, M., Chelloug, S. A., Awadallah, M. A., Al-Betar, M. A., Al-Qaness, M. A., & Forestiero, A. (2022). Intrusion detection system for IoT based on deep learning and modified reptile search algorithm. *Computational Intelligence and Neuroscience*, *2022*(1), 6473507. https://doi.org/10.1155/2022/6473507

[40] Logeswari, G., Bose, S., & Anitha, T. J. I. A. (2023). An intrusion detection system for sdn using machine learning. *Intelligent Automation & Soft Computing*, *35*(1), 867–880. https://doi.org/10.32604/iasc.2023.026769

[41] Imran, M., Haider, N., Shoaib, M., & Razzak, I. (2022). An intelligent and efficient network intrusion detection system using deep learning. *Computers and Electrical Engineering*, *99*, 107764. https://doi.org/10.1016/j.compeleceng.2022.107764

[42] Mahmoud, M., Kasem, M., Abdallah, A., & Kang, H. S. (2022). Ae-lstm: Autoencoder with lstm-based intrusion detection in iot. In *2022 International Telecommunications Conference*, 1–6. https://doi.org/10.1109/ITC-Egypt55520.2022.9855688

[43] Zhao, R., Mu, Y., Zou, L., & Wen, X. (2022). A hybrid intrusion detection system based on feature selection and weighted stacking classifier. *IEEE Access*, *10*, 71414–71426. https://doi.org/10.1109/ACCESS.2022.3186975

[44] Hore, S., Ghadermazi, J., Shah, A., & Bastian, N. D. (2024). A sequential deep learning framework for a robust and resilient network intrusion detection system. *Computers & Security*, *144*, 103928. https://doi.org/10.1016/j.cose.2024.103928

[45] Mendonça, R. V., Teodoro, A. A., Rosa, R. L., Saadi, M., Melgarejo, D. C., Nardelli, P. H., & Rodríguez, D. Z. (2021). Intrusion detection system based on fast hierarchical deep convolutional neural network. *IEEE Access*, *9*, 61024–61034. https://doi.org/10.1109/ACCESS.2021.3074664

[46] Kumawat, H., & Meena, G. (2014). Characterization, detection and mitigation of low-rate DoS attack. In *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies*, 1–5. https://doi.org/10.1145/2677855.2677924

[47] Mukherjee, S., & Sharma, N. (2012). Intrusion detection using naive Bayes classifier with feature reduction. *Procedia Technology*, *4*, 119–128. https://doi.org/10.1016/j.protcy.2012.05.017

[48] Elbasiony, R. M., Sallam, E. A., Eltobely, T. E., & Fahmy, M. M. (2013). A hybrid network intrusion detection framework based on random forests and weighted k-means. *Ain Shams Engineering Journal*, *4*(4), 753–762. https://doi.org/10.1016/j.asej.2013.01.003

[49] De la Hoz, E., De La Hoz, E., Ortiz, A., Ortega, J., & Prieto, B. (2015). PCA filtering and probabilistic SOM for network intrusion detection. *Neurocomputing*, *164*, 71–81. https://doi.org/10.1016/j.neucom.2014.09.083

[50] Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. *Procedia Computer Science*, 89, 213–217. https://doi.org/10.1016/j.procs.2016.06.047

[51] Belavagi, M. C., & Muniyal, B. (2016). Performance evaluation of supervised machine learning algorithms for intrusion detection. *Procedia Computer Science*, *89*, 117–123. https://doi.org/10.1016/j.procs.2016.06.016

[52] Thaseen, I. S., & Kumar, C. A. (2017). Intrusion detection model using fusion of chi-square feature selection and multi class SVM. *Journal of King Saud University-Computer and Information Sciences*, *29*(4), 462–472. https://doi.org/10.1016/j.jksuci.2015.12.004

[53] Meena, G., & Choudhary, R. R. (2017, July). A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA. In 2017 International Conference on Computer, Communications and Electronics (Comptelix) (pp. 553–558). *IEEE*. https://doi.org/ 10.1109/COMPTELIX.2017.8004032

[54] Ahmad, S., Arif, F., Zabeehullah, Z., & Iltaf, N. (2020). Novel approach using deep learning for intrusion detection and classification of the network traffic. In *2020 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications*, 1–6. https://doi.org/10.1109/CIVEMSA48639.2020.9132744

[55] bhai Gupta, A. R., & Agrawal, J. (2020). A comprehensive survey on various machine learning methods used for intrusion detection system. In *2020 IEEE 9th International Conference on Communication Systems and Network Technologies*, 282–289. https://doi.org/10.1109/CSNT48778.2020.9115764

[56] Abou Khamis, R., Shafiq, M. O., & Matrawy, A. (2020). Investigating resistance of deep learning-based ids against adversaries using min-max optimization. In *ICC 2020 IEEE International Conference on Communication*, 1–7. https://doi.org/10.1109/ICC40277.2020.9149117

[57] Hindy, H., Atkinson, R., Tachtatzis, C., Colin, J. N., Bayne, E., & Bellekens, X. (2020). Utilising deep learning techniques for effective zero-day attack detection. *Electronics*, *9*(10), 1684. https://doi.org/10.3390/electronics9101684

[58] Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, A. A. (2023). CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors*, *23*(13), 5941. https://doi.org/10.3390/s23135941

How to Cite: Meena, G., & Indian, A. (2025). IDS-IoT: Intrusion Detection System for the Internet of Things Using Enhanced Long-Short Term Memory. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA52025066