IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 59, No. 1, January, 2026, pp. 69-77
Journal homepage: http://iieta.org/journals/jesa

SICA: Generation and Deep Learning-Based Evaluation of a Novel Dataset for Intrusion Detection in SDN-IoT Environments

Mustafa T. Saleh[1*], Ali H. Hamad[2]

[1] Informatics Institute for Postgraduate Studies, University of Information Technology and Communication, Baghdad 10071, Iraq
[2] Biomedical Application Department, College of Artificial Intelligence, University of Baghdad, Baghdad 10071, Iraq

Corresponding Author Email: phd202230706@iips.edu.iq

https://doi.org/10.18280/jesa.590108

Received: 13 November 2025
Revised: 13 January 2026
Accepted: 20 January 2026
Available online: 31 January 2026

Keywords:
Intrusion Detection System, Software-Defined Networking, Internet of Things, deep learning, custom dataset

ABSTRACT

The integration of Software-Defined Networking (SDN) with the Internet of Things (IoT) offers considerable flexibility and programmability in network administration, while also introducing new security concerns. A key obstacle to the development of effective Intrusion Detection System (IDS) in this context is the scarcity of publicly accessible, realistic datasets that accurately reflect contemporary attacks. This study fills this gap by introducing a novel, comprehensive dataset for anomaly-based intrusion detection in SDN-IoT networks. The dataset SDN-IoT Cyber Attack (SICA), was generated in a simulated multi-controller SDN topology using Mininet and Ryu controllers, including both standard traffic and 22 unique attack types across inter- and intra-domain scenarios. Raw packet data was analyzed using CICFlowmeter to derive flow-based characteristics. The dataset was validated through testing, training, and assessing four unique deep learning models: a Deep Neural Network (DNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and a 1D Convolutional Neural Network (1D-CNN). The efficacy of these models on the proposed dataset was evaluated and compared with two established datasets, IoT-SDN and InSDN. The findings indicate that all four deep learning models achieved almost flawless 97% accuracy and 0.07 loss on the proposed dataset, underscoring its dependability and applicability as a novel resource for training and verifying sophisticated deep learning-based security solutions for SDN-IoT systems.

1. INTRODUCTION

The Internet of Things (IoT) refers to the ability of embedded devices to connect to the internet. The concept of IoT is to connect everyday objects to a network, enabling the collection of extensive data from devices with varying functionalities and constrained resources. This may hinder the provision of security and safety. Analyzing network data to identify anomalous activities requires significant time and effort. Numerous lightweight techniques for enhancing IoT security have been developed in recent years. Nevertheless, these methods are unable to address the significant security threats that have emerged recently. It is essential to develop robust Intrusion Detection Systems (IDS) capable of protecting against a diverse array of threats [1].

The advent of SDN signifies a substantial enhancement in network architecture, ushering in a new epoch of efficiency, adaptability, and programmability. The fundamental principle of SDN is the segregation of the control plane from the data plane. This innovative method provides network administrators with significant control over traffic management and network operations. This paradigm shift facilitates rapid network establishment and enhancement; however, it also reveals several inherent security weaknesses within the SDN architecture. The centralized nature of the SDN control plane, coupled with the programmability of network interfaces, renders SDN systems highly vulnerable to cyber threats, including Distributed Denial-of-Service (DDoS) attacks and advanced intrusion attempts.

IDSs are the main component of cybersecurity. They use novel techniques to identify and highlight anomalies [2]. Recent advancements in Machine Learning (ML) and Deep Learning (DL) have significantly enhanced the efficacy of IDS. This suggests they may be used across several domains to combat emerging threats. ML and DL, both subsets of Artificial Intelligence (AI), are often used in this domain and have significantly advanced as automated methods for detecting intrusions. They have demonstrated the ability to identify advanced threats, including malicious URLs, DDoS attacks, and botnet attacks. During training, machine learning and deep learning models use sample network traffic data to differentiate between legitimate and malicious activity. These models autonomously detect and terminate processes immediately. Furthermore, IDS using machine learning and deep learning methodologies may identify novel, unforeseen threats that conventional approaches cannot manage.

IDSs typically use one of two methodologies: signature-based or anomaly-based. Signature-based methodologies are

used in commercial solutions because of their efficacy in detecting many attacks while generating few false positives; nonetheless, they are inadequate for identifying novel or unreported network intrusions that occur daily. Conversely, the anomaly-based detection method has attracted significant academic interest due to its ability to identify novel attacks. Despite prior research on anomaly detection systems for SDN networks, several issues have to be addressed to develop effective IDS systems that adhere to the SDN standard. A significant issue with using IDS is the absence of publicly available datasets derived directly from SDN networks for training and evaluating anomaly detection systems. The majority of studies use intrusion detection datasets created for conventional networks. Due to the virtualization of SDN, the network is susceptible to novel forms of attacks that are infeasible on traditional networks [3].

This study aims to provide a dataset that includes the essential characteristics for detecting attacks and evaluating anomaly-based IDS in SDN environments for the IoT. The project comprises two components: the first module collects traffic data to create the dataset, while the second module uses four deep learning architectures—Deep Neural Network (DNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and 1D Convolutional Neural Network (1D-CNN)—to assess the dataset. The suggested dataset encompasses a variety of attacks, the execution of a multidomain attack scenario, and the use of realistic traffic traces. Inter-domain attacks occur between different domains of the SDN network (with different controllers), while intra-domain attacks occur within the same SDN controller network. These attacks typically involve a perpetrator targeting a system, network, or organization separate from their own.

In contrast, an intra-attack occurs within the same system or entity. These attacks involve nefarious actions carried out by individuals or groups who already have access to the targeted system or organization. The findings of this investigation can be encapsulated as follows:

•The benefits of the existing IDS, SDN, and IoT datasets were examined.

•Two traffic profiles were utilized: the usual profile and the attack profile.

•A total of 22 attack types were analyzed across two scenarios: interdomain and intra-domain attacks.

•The dataset was empirically assessed using deep learning approaches.

The rest of this paper is organized as follows: a survey of related work is presented in Section 2, and Section 3 describes the methodology used in this work. Section 4 introduces the proposed dataset generation, while the results and discussion are presented in Section 5. Finally, a conclusion with suggestions for future work was introduced in Section 6.

2. RELATED WORK

The growing convergence of SDN and the IoT has introduced new complexities and security risks. This has prompted extensive research into advanced IDS. The following publications note significant contributions and their corresponding limitations in this area [4]. Provide a framework for an IDS in an IoT environment that is built on SDN. When choosing features, it uses explainable artificial intelligence (XAI) and DL. An important new feature of XAI is the ability to employ SHAP and LIME to identify domain-specific characteristics. As a result, computing the model becomes much easier. The findings show that memory use may be reduced by 49.4 percent and training time by 6.8 percent while still achieving satisfactory detection accuracy on the InSDN and X-IIOTID datasets. One drawback is that these datasets are well-suited to the XAI feature selection technique. Large, dynamic, and resource-constrained SDN-IoT systems may have unanswered questions about their long-term scalability and real-time inference.

Mahmoodi et al. [5] provided autonomous Federated Learning (FL) with a framework that IDSs can use to safeguard public networks. To address problems with traditional IDSs, which often rely on outdated and inaccurate data, this research introduces a testbed design. The goal of FL is to protect sensitive information while facilitating cooperative model training across different parts of a network. Constraint: Due to the testbed focus, no large-scale public network assessment has been conducted in the real world. Issues such as non-IID data, many customers, the threat of data poisoning attacks, and the challenge of securely and reliably merging global models are part of the real world. To identify security issues in SDN-IoT, Dhirar and Hamad [6] developed and evaluated a new IDS tailored to this environment. The study compares and contrasts the new dataset with current ones, including InSDN, BoT-IoT, and ToN-IoT. They use four DL models—CNN, LSTM, RNN, and DNN—to assess performance using metrics including F1-score, accuracy, and precision. Cons: Even if a new, tailored dataset is created, it could quickly become outdated due to the rapid advancement of zero-day attacks. The models compared do not use a custom-built deep learning architecture that leverages the unique features of their datasets.

A thorough analysis of ML and DL for IDS is presented by Rakine et al. [7], which categorizes contributions by the networks they aim to protect: SDN, IoT, and cloud computing. This investigation highlights the need to understand network-specific vulnerabilities to improve IDS. Drawback: Instead of offering a unique technical solution, this research highlights the constraints of the existing domain. Given the limited resources, the authors highlight how challenging it is to identify the various risks in heterogeneous IoT networks. Furthermore, the paper highlights the need for further study into how IDS can quickly adapt to the ever-changing environments in dynamic SDN systems. Dhirar and Hamad [8] gave an example of an IoT network with several controllers that uses SDN and a Federated Deep Learning Intrusion Detection System (FL-IDS). Using FL to enable local DL model training is the central novel concept. This improves security, streamlines operations, and lessens the risks associated with central data storage. Users of the Interplanetary File System (IPFS) may engage in decentralized model training and sharing. A drawback of decentralized IPFS sharing is the increased difficulty in communicating and the additional latency in model synchronization and aggregation that results. As a result, real-time identification—which is crucial for Network-based Intrusion Detection Systems (NIDS) to function properly—may become more difficult. It may be difficult to scale up in large, real-world installations due to non-IID data and limited resources across different types of IoT devices. Elsayed et al. [9] introduced InSDN, an SDN Intrusion dataset that aims to solve the long-standing problem of incompatible and out-of-date datasets, such as KDD'99, which hinder the development of efficient IDS in SDN. Because it provides a foundation for building and testing

anomaly detection algorithms, the dataset is a valuable asset. Restrictions: Although extensive data collection was undertaken, the collected attack traffic pertains only to distributed denial-of-service (DDoS) flooding attacks. Due to the dataset's narrow focus, it is more difficult to train and evaluate IDS models capable of detecting the full spectrum of contemporary threats, including probing and user-to-root attacks, which are essential for comprehensive network security. Khorseed and Hamad [10] discussed the increasing susceptibility of SDN to DDoS attacks, particularly those using IoT devices. Their major contribution is a method to prevent distributed DDoS attacks spanning several domains, which are controlled and protected by Hyperledger Fabric Blockchain Technology. The intended outcome of implementing smart contracts on the blockchain is a decentralized, immutable method for handling attacks. Cons: Due to its consensus methods and block validation processes, blockchain technology inherently introduces a substantial latency overhead. In high-speed, dynamic SDN systems, where millisecond-level responses are required to prevent service disruptions and controller resource depletion, the proposed solution may not be able to mitigate DDoS attacks in real time. The eleventh resolves a key concern in network security: the lack of publicly accessible, relevant datasets for IDS in IoT environments led by SDN. With an emphasis on both intra- and inter-domain threats, this study presents a new dataset for testing anomaly-based intrusion detection. This dataset is essential for testing algorithms against complex, cross-domain threats; it contains 86 network traffic characteristics. The dataset is static, as most datasets are, and it restricts coverage to a specific network configuration and attack types. This makes it vulnerable to new or zero-day attacks and quickly outdated. Consequently, IDS models that rely solely on this data are less likely to succeed in real-world scenarios. Table 1 compares the previous studies in different evaluation terms.

Table 1. Comparison of previous work

Ref.	Focus Area	Technology / Model Used	Network Environment	Key Contribution
[4]	Feature Selection	DL + XAI (Explainable AI)	SDN-based IoT	Used XAI to select features and domain-constrained techniques to improve transparency.
[5]	Distributed Learning	Autonomous Federated Learning	Public Networks	Proposed an autonomous FL framework to handle distributed IDS without central data sharing.
[6]	Dataset Evaluation	DL Models (CNN, RNN, etc.)	SDN-IoT	Introduced a novel IDS dataset and compared it against InSDN, BoT-IoT, and ToN-IoT.
[7]	Literature Review	ML and DL Algorithms	Multi-network (General)	A comprehensive review of current techniques across various network architectures.
[8]	Privacy & Scaling	Federated Deep Learning	SDN-based IoT	Combined Federated Learning with DL to enhance security and privacy in IoT ecosystems.
[9]	Data Benchmarking	Dataset Creation (InSDN)	SDN	Provided a specialized dataset (InSDN) specifically designed for SDN traffic patterns.
[10]	DDoS Mitigation	Hyperledger Fabric Blockchain	SDN	Used blockchain technology to mitigate inter- and intra-domain DDoS attacks.

3. METHODOLOGY

Attacks should be considered because they can have a significant impact on both the network's infrastructure and the people who use it. So, figuring out which attacks are happening and what they are is very important for keeping the network safe and avoiding threats that could compromise it. So, the suggested model primarily uses DL-based methods to detect various types of attacks on the IoT network.

3.1 Intrusion Detection System

IDSs are security mechanisms that oversee network traffic and system activities for anomalous activity or recognized dangers, subsequently notifying administrators. They are primarily classified according to their detection methodology and monitoring range. There are two types of IDS categorized by detection method: first, Signature Detection (Misuse-Based Intrusion Detection System). This strategy relies on a continually maintained database of signatures, which are established patterns or fingerprints of recognized attacks and suspicious activities. The IDS observes network packets or system events and juxtaposes them with the archived signatures. An alert is activated by a match, signifying a recognized threat [11].

Furthermore, behavioral anomaly detection (IDS): This approach creates a benchmark for typical network or system behavior. Any behavior that markedly diverges from this established normal baseline is identified as an anomaly and a potential intrusion. It employs statistical models, machine learning, or alternative techniques to construct a profile of "normal" activities over time. It subsequently contrasts real-time activity with this baseline.

According to the Monitoring Scope, there are two categories of IDS: Firstly, NIDS monitor traffic within a network segment, analyzing it in real time to identify threats that transit the network. It is positioned at crucial locations throughout the network, including the perimeter and critical junctures such as switches or routers, and frequently uses a network tap or a switch mirror port (SPAN). Additionally, it captures comprehensive network packets, including headers and payloads, across the entire monitored network segment. Secondly, a Host-based Intrusion Detection System (HIDS) operates as a software agent on a specific host (server, laptop, or workstation) to monitor activities pertinent to that machine. It was installed directly on the operating system of each host device [12, 13]. Table 2 shows a brief comparison of different types of IDS systems.

3.2 Software-defined network

The architecture of SDN consists of three essential layers that work together to enable centralized, programmable network management and dynamic traffic control. The SDN

architecture comprises the following layers:

The infrastructure layer consists of SDN switches, including Open Virtual Switch (OVS), which relay traffic to the control plane without examination. The control plane consists of a centralized SDN controller that configures switches by establishing forwarding rules and maintaining an extensive network-wide view. Its essential functions include topological recognition and dynamic flow arrangement. The application layer contains network management software that configures the SDN controller to implement dynamic rules, including firewalling and network address translation (NAT) across switches, thereby abstracting policy logic from the foundational equipment [14].

Table 2. Comparative analysis of Intrusion Detection System primary types [13]

Feature	Signature Detection	Anomaly Detection	Network-based IDS (NIDS)	Host-based IDS (HIDS)
Detection Basis	Matches traffic against known attack patterns	Flags deviations from a baseline	Monitors network traffic	Monitors activity on a single host
Threat Coverage	Known attacks	Unknown attacks	External threats, network-wide attacks	Insider threats, post-exploit activity
Computational Needs	Low	High (ML)	Moderate	Moderate to High
Example	Detecting a specific malware	Flagging a user	Flagging a rapid port scan	Alerting on the modification of the file

3.3 Deep learning algorithms

A DNN is a variant of an Artificial Neural Network (ANN) distinguished by the presence of numerous hidden layers (exceeding one). It is a feedforward network in which data flows unidirectionally from the input layer through the hidden layers to the output layer. Deep neural networks (DNNs) serve as universal function approximators, adept at learning intricate, nonlinear relationships between inputs and outputs, making them appropriate for classification and regression tasks across many fields, such as image recognition and natural language processing. Training entails iterative optimization using backpropagation and an optimizer such as Stochastic Gradient Descent (SGD) [15].

A RNN is a type of neural network specifically engineered to handle sequential input, such as time series or textual information. In contrast to DNNs, RNNs have recurrent connections that allow the network to retain a hidden state, or "memory," of prior inputs in the sequence. This internal memory allows them to capture temporal dependencies. Standard RNNs have the vanishing gradient problem, hindering their ability to learn long-range dependencies, a significant disadvantage in jobs involving extensive sequences [16].

LSTM is an advanced variation of the RNN designed to address the vanishing gradient problem, enabling it to learn and retain long-term relationships in sequential input. The fundamental component of an LSTM unit is the cell state, which functions as a conduit for information transfer. The information flow is governed by three cognitive mechanisms known as gates (the forget, input, and output gates), which selectively determine what information to retain, modify, or release. LSTMs emerged as the preeminent architecture for numerous sequence modeling tasks, including machine translation and speech recognition, prior to the ascendance of the Transformer architecture [17].

A 1D-CNN is a distinct CNN architecture in which convolutional filters (or kernels) traverse the input data along a single dimension. This renders it optimal for processing sequential or time-series data, including sensor data, financial time series, or textual information. The 1D convolution operation extracts local patterns, such as temporal motifs, by applying a filter to a contiguous sequence of data points. These networks excel at feature extraction and are frequently employed for classification or prediction problems where sequence and local interactions are crucial [18].

4. PROPOSED SYSTEM DESIGN

Developing a complete dataset requires deploying multiple application services within the experiment-generated dataset. Consequently, contemporary Internet assaults, which can be executed within existing SDNs, can be accurately modeled. Furthermore, the assault scenarios must include contemporary attack routes across various SDN components. Furthermore, many attack scenarios are being analyzed, originating from multiple sources within and outside the SDN domain. The topology is illustrated by establishing four subdomains using MiniNet on Ubuntu 20.04 LTS. Two Ryu controllers manage the four OpenFlow vSwitch (OVS) switches connected to these subdomains. The Ryu controller is an open-source software framework written in Python. This controller is distinguished by its ability to efficiently and effectively handle demands in modern networking systems while remaining scalable. It is interoperable with numerous SDN protocols and features, making it suitable for varied networking scenarios and interactions with multiple technologies. The first two subdomains pertain to a conventional network encompassing many services, including HTTP and FTP servers. Conversely, the latter two subdomains relate to an IoT network specifically designed for conducting experiments in both indoor and outdoor settings Figure 1. Shows the proposed multi-controller SDN scheme architecture.

The process comprises four basic stages: Data Generation, Data Processing and Feature Extraction, Model Training and Evaluation, and Final Prediction, typically in a sequential order. The methodology for constructing, analyzing, and assessing a network traffic dataset for deep learning-based intrusion detection within an SDN framework is illustrated in Figure 2.

The four principal phases of the proposed methodology are as follows.

•Dataset generation phase: The primary objective of this phase is to generate raw network traffic data within a regulated SDN environment. The RYU controller, an open-source SDN controller, and Mininet, a network emulator, are utilized to establish an SDN network environment. This emulates a real-

world SDN architecture. The simulated SDN produces both standard and various types of attack traffic. Metasploit and hping3 are likely employed to simulate many attack vectors, including exploits and denial-of-service attacks. The ping tool was used to generate normal traffic. A packet analyzer like Wireshark is utilized to capture the unprocessed network traffic data for various types of attacks alongside standard traffic. A singular Pcap (Packet Capture) file, a prevalent format for archiving network packets, is generated by amalgamating all individual traffic captures. Table 3 shows the statistical analysis of the proposed SICA dataset implementation. compared with other datasets (SDN-IoT and InSDN).

•Feature extraction and data processing: This phase prepares the raw data for machine learning. CICFlowMeter is widely used to transform raw network traffic (PCAP files) into structured, bidirectional flows. It extracts 84 statistical features, which can be categorized into four primary types: Basic Flow Features, Time-Based Features (Flow IAT), Packet Length Features, and Flag and Header-Based Features. 10 features were manually removed from network traffic, including 'Flow ID', 'Src IP', 'Dst IP', 'Timestamp', 'Idle Mean', 'Idle Max', 'Idle Min', 'Idle Std', and 'Flow Duration'. This converts raw packets into valuable numerical attributes. It is essential to annotate each flow's characteristics to determine whether it represents standard traffic or a specific type of attack. This process generates the ground truth required for supervised deep learning. The dataset of annotated features is further enhanced as detailed below: One-hot encoding (transforming categorical labels, such as attack names, into a numerical format suitable for deep learning models), normalization (applying the Min-Max algorithm), data cleaning (eliminating inconsistent or missing data), and data splitting (with a ratio of 70% for training, 20% for testing, and 10% for validation).

•Model training and evaluation: During this phase, the deep learning intrusion detection system is developed and assessed. A range of Deep Learning (DL) models is trained on the provided dataset to discern patterns that distinguish legal traffic from malicious behavior. The models discussed include DNN, RNN, LSTM, and 1D-CNN. The performance of the trained models (e.g., accuracy, precision, and recall) in accurately recognizing different attack types is assessed using previously unreported testing data. Table 4 shows the architectural design of the different models used in this work.

•Final prediction: This concluding phase leverages the distinct advantages of each model. The forecasts from many trained models (DNNs, RNNs, etc.) are combined using a voting ensemble methodology. The final prediction is determined by a majority vote or a weighted vote based on a confidence metric, aiming to establish a detection system that is more dependable and accurate than any individual model alone.

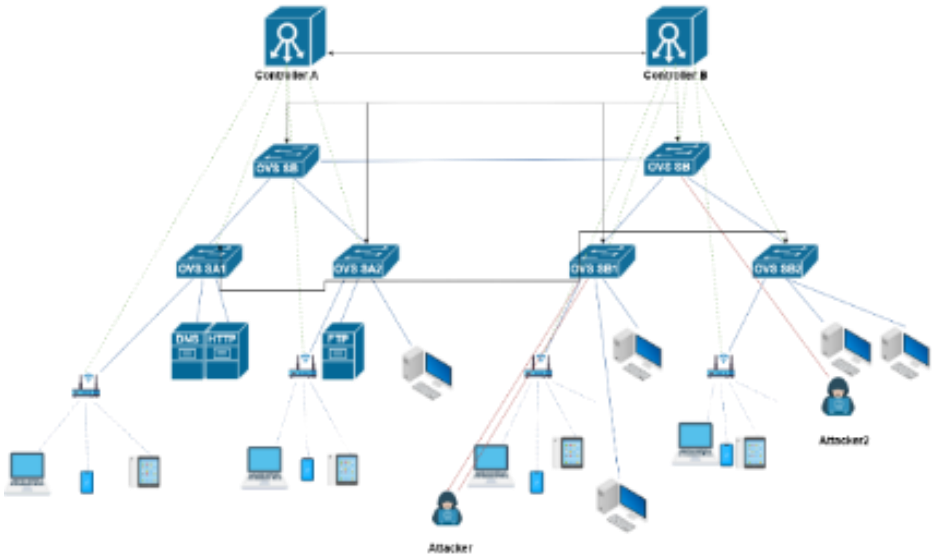


Figure 1. Proposed multi-controller Software-Defined Networking

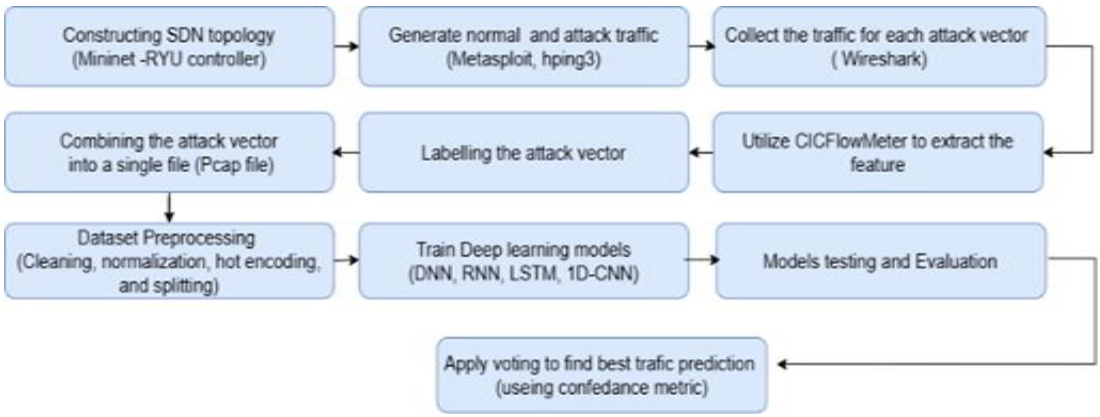


Figure 2. Process flow for dataset generation

Table 3. Statistical analysis of the proposed SICA dataset compared with InSDN and SDN-IoT

Parameters	Description		
Dataset	**InSDN**	**SDN-IoT**	**SICA**
No. of Records	343,889	844,142	2,298,903
No. of Features	83	84	84
No. of attack classes	9	14	22
Dataset size	141.71 MB	470.31 MB	1262 MB
Attack types	Probe, DDoS, Normal, DoS, DDoS, BFA, Web-Attack, BOTNET, U2R	Botnet, Brute-Force, DoS, DDoS (ICMP, SYN, UDP), Exploitation, Malware, MIRAI, Probe, R2L, UR2, Web-based, Spoofing, Recon	['Arp_spoofing' 'apache_range_dos' 'arp_sweep' 'avahi_portzero' 'brute_dirs' 'ddos' 'fragmentation_attack' 'http_flood' 'http_get_uri_long' 'http_get_uri_strings' 'normal' 'ntp_prot' 'ntp_protocol_fuzzer' 'slowloris_attacks' 'smb2_negotiate_corrupt' 'smb_tree_connect_corrupt' 'ssh' 'ssh_version_corrupt' 'synflood' 'teardrop' 'udp_probe' 'udp_reflection']

Table 4. Architectural design of different deep learning models

Model	Layers	Neurons	Hyperparameters
LSTM	Input → LSTM (5x) → Output	256 → 128 → 128 → 64 → 32 → Output	Activations: Tanh (hidden), Softmax (output)
CNN	Input → Conv1D → MaxPool → Conv1D → MaxPool → Conv1D → MaxPool → Flatten → Dense (2x) → Output	256 → 128 →128 → 64 → 32 → Output	Activations: ReLU (hidden), Softmax (output), Kernel size = 2,3
RNN	Input → GRU → Dense (2x) → Output	64 → 64 → 32 → Output	Activations: ReLU (hidden), Softmax (output)
DNN	Input → Dense (5x) → Output	256 → 128 → 128 → 64 → 32 → Output	Activations: ReLU (hidden), Softmax (output)

5. RESULTS AND DISCUSSION

This section discusses the performance outcomes obtained from evaluating four classifiers: DNN, RNN, 1D-CNN, and LSTM. The evaluation metrics include accuracy, loss, precision, recall, and F1-Score. Figures 3, 4, and 5 compare training accuracy and loss for the SICA, SDN-IoT, and InSDN datasets across the four deep learning models.

Figure 3. Accuracy and loss of four deep learning algorithms for the SICA dataset

The SICA training graphs show rapid convergence within the first 10 epochs. All models achieved high performance (>97%). The DNN reached the highest test accuracy of 0.9758. The 1D-CNN training loss (yellow line) shows noticeable "spikes" or fluctuations compared to the smoother DNN and LSTM curves, suggesting it may be more sensitive to specific traffic patterns in this dataset. The SDN-IoT appears to be the most challenging dataset for the models. Accuracy across all models is significantly lower here (ranging from 0.8103 to 0.8485) compared to the other datasets. The training curves for accuracy and loss are much flatter, indicating the models struggled more to find distinct features to separate malicious from benign traffic. The test loss is quite high (0.34–0.42), suggesting the models have less confidence in their predictions than SICA or InSDN. The InSDN Dataset performed exceptionally well on InSDN, reaching near-perfect metrics. Both DNN and LSTM achieved a test accuracy of 0.9961. The training loss dropped to near zero very quickly (before epoch 4). Precision and Recall are nearly identical, indicating the models are balanced and rarely produce false positives or false negatives on this specific data.

Table 5 compares test accuracy and loss across different datasets and deep learning algorithms, and also shows the model size, a very important metric given that the proposed system operates on IoT nodes with limited computational and storage resources. While the DNN is the most accurate, the RNN is roughly 4x smaller. In a resource-constrained IoT environment, the 2-3% drop in accuracy (on SDN-IoT) might be a necessary trade-off for the massive savings in memory and power. The results show that an IDS's effectiveness is highly dependent on the environment. A model tuned for InSDN (99% accuracy) may achieve only ~84% accuracy when deployed in an SDN-IoT environment without retraining.

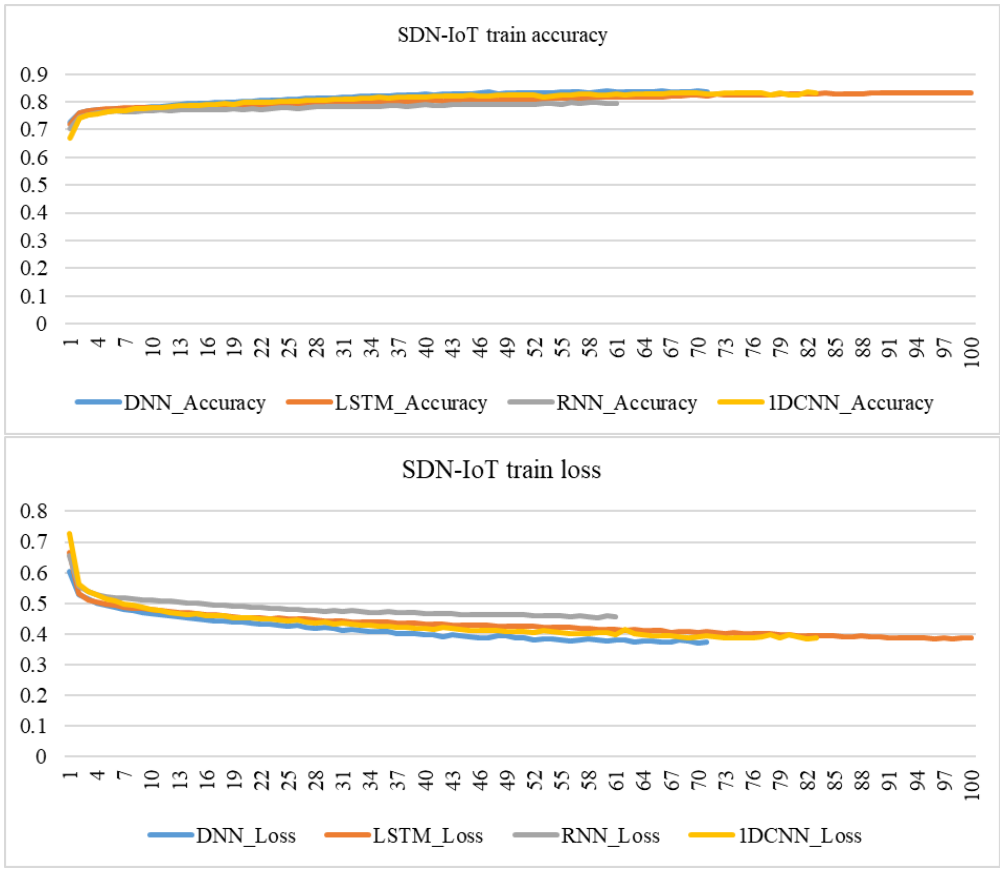
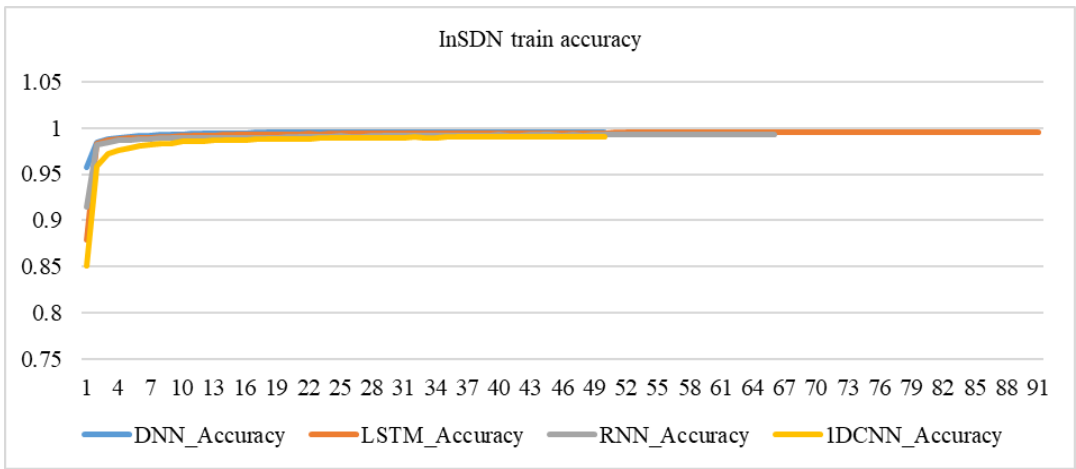


Figure 4. Accuracy and loss of four deep learning algorithms for the IoT-SDN dataset



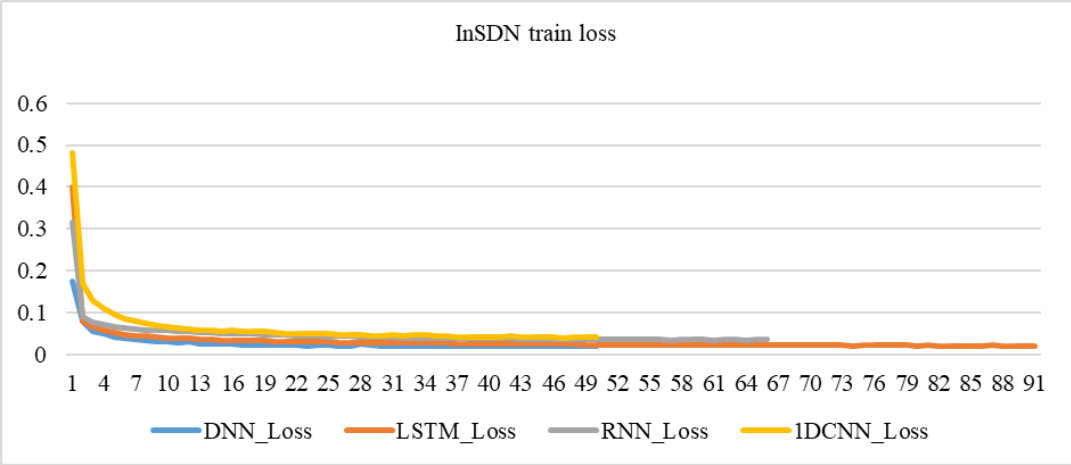


Figure 5. Accuracy and loss of four deep learning algorithms for the InSDN dataset

Table 5. Test evaluation of accuracy and loss of four deep learning algorithms for SICA, SDN-IoT, and InSDN datasets

Algorithm		Test Accuracy	Test Loss	Test Precision	Test Recall	Model Size (KB)
SICA	DNN	0.9758	0.0726	0.9765	0.9758	80.71
	LSTM	0.9756	0.0736	0.9763	0.9756	68.27
	RNN	0.9708	0.0831	0.9734	0.9708	18.96
	1D_CNN	0.9703	0.1128	0.9713	0.9703	22.43
IoT-SDN	DNN	0.8485	0.3498	0.8529	0.8485	78.26
	LSTM	0.8413	0.367	0.8492	0.8413	67.01
	RNN	0.8103	0.4207	0.8139	0.8103	17.7
	1D_CNN	0.8387	0.3703	0.8421	0.8387	21.17
InSDN	DNN	0.9961	0.0194	0.9961	0.9961	79.04
	LSTM	0.9961	0.0208	0.9961	0.9961	67.41
	RNN	0.9943	0.0294	0.9936	0.9943	18.1
	1D_CNN	0.9939	0.0308	0.9932	0.9939	21.57

6. CONCLUSION

This study successfully addressed the critical need for a modern, relevant dataset to develop IDSs in integrated SDN-IoT environments. The primary contribution of this work is twofold: first, the generation of a new, comprehensive dataset (SICA) that models 22 types of contemporary inter- and intra-domain attacks within a realistic, multi-controller SDN-IoT topology; and second, a thorough empirical evaluation of this dataset using four deep learning models (DNN, RNN, LSTM, and 1D-CNN).

Table 6. Overall models comparison

Feature	Best Model	Reasoning
Detection Power	DNN	Highest accuracy and lowest loss across all 3 datasets.
Computational Efficiency	RNN	Smallest model size (~18 KB), making it ideal for edge IoT devices with limited memory.
Balanced Choice	LSTM	Matches DNN accuracy on InSDN but is ~15% smaller in size.
Stability	DNN / LSTM	Smoother training curves with fewer fluctuations than 1D-CNN.

The methodology involved emulating the network using Mininet and Ryu controllers, and employing tools such as CICFlowMeter for high-fidelity feature extraction. The evaluation results were definitive: all four deep learning models achieved high accuracy and near-zero loss when trained and tested on the SICA dataset, indicating that the dataset is highly reliable and provides clear distinctions between normal and attack traffic. This performance was contrasted with the models' evaluation on the IoT-SDN and InSDN datasets, which provided more varied, "realistic" results and established the IoT-SDN dataset as a challenging benchmark for comparison. Table 6 shows an overall comparison between the different models.

REFERENCES

[1] Fatani, A., Abd Elaziz, M., Dahou, A., Al-Qaness, M.A., Lu, S. (2021). IoT intrusion detection system using deep learning and enhanced transient search optimization. IEEE Access, 9: 123448-123464. https://doi.org/10.1109/ACCESS.2021.3109081

[2] Vishwakarma, M., Kesswani, N. (2025). StaEn-IDS: An explainable stacking ensemble deep neural network-based intrusion detection system for IoT. IEEE Access, 13: 109713-109728. https://doi.org/10.1109/ACCESS.2025.3582391

[3] Allafi, R., Alzahrani, I.R. (2024). Enhancing cybersecurity in the internet of things environment using artificial orca algorithm and ensemble learning model. IEEE Access, 12: 63282-63291. https://doi.org/10.1109/ACCESS.2024.3390093

[4] Tserenkhuu, M., Hossain, M.D., Taenaka, Y., Kadobayashi, Y. (2025). Intrusion detection system framework for SDN-based IoT networks using deep

learning approaches with XAI-based feature selection techniques and domain-constrained features. IEEE Access, 13: 136864-136880. https://doi.org/10.1109/ACCESS.2025.3595595

[5] Mahmoodi, A.B.Z., Sheikhi, S., Peltonen, E., Kostakos, P. (2023). Autonomous federated learning for distributed intrusion detection systems in public networks. IEEE Access, 11: 121325-121339. https://doi.org/10.1109/ACCESS.2023.3327922

[6] Dhirar, H., Hamad, A. (2025). Comparative evaluation of a novel IDS dataset for SDN-IoT using deep learning models against InSDN, BoT-IoT, and ToN-IoT. Measurement: Digitalization, 4: 100015. https://doi.org/10.1016/j.meadig.2025.100015

[7] Rakine, I., Oukaira, A., Guemmat, K.E., Atouf, I., Ouahabi, S., Talea, M., Bouragba, T. (2025). Comprehensive review of intrusion detection techniques: ML and DL in different networks. IEEE Access, 13: 104357-104367. https://doi.org/10.1109/ACCESS.2025.3579990

[8] Dhirar, H., Hamad, A.H. (2025). Federated deep learning intrusion detection system on software defined-network based internet of things. IAES International Journal of Artificial Intelligence, 14(4): 3109-3120. https://doi.org/10.11591/ijai.v14.i4.pp3109-3120

[9] Elsayed, M.S., Le-Khac, N.A., Jurcut, A.D. (2020). InSDN: A novel SDN intrusion dataset. IEEE Access, 8: 165263-165284. https://doi.org/10.1109/ACCESS.2020.3022633

[10] Khorseed, W.S., Hamad, A.H. (2024). Inter and intra domain DDoS attack mitigation for software defined network based on hyperledger fabric blockchain technology. Ingenierie des Systemes d'Information, 29(1): 301-311. https://doi.org/10.18280/isi.290130

[11] Dhirar, H., Hamad, A. (2025). Internet of Things software-defined network intrusion detection dataset: inter-and intra-domain. Al-Khwarizmi Engineering Journal, 21(3): 35-47. https://doi.org/10.22153/kej.2025.08.001

[12] Einy, S., Oz, C., Navaei, Y.D. (2021). The anomaly-and signature-based IDS for network security using hybrid inference systems. Mathematical Problems in Engineering, 2021(1): 6639714. https://doi.org/10.1155/2021/6639714

[13] Liu, L., Xu, B., Zhang, X., Wu, X. (2018). An intrusion detection method for internet of things based on suppressed fuzzy clustering. EURASIP Journal on Wireless Communications and Networking, 2018(1): 113. https://doi.org/10.1186/s13638-018-1128-z

[14] Valdivieso Caraguay, Á.L., Benito Peral, A., Barona Lopez, L.I., Garcia Villalba, L.J. (2014). SDN: Evolution and opportunities in the development IoT applications. International Journal of Distributed Sensor Networks, 10(5): 735142. https://doi.org/10.1155/2014/735142

[15] Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., et al. (2019). A state-of-the-art survey on deep learning theory and architectures. Electronics, 8(3): 292. https://doi.org/10.3390/electronics8030292

[16] Mienye, I.D., Swart, T.G., Obaido, G. (2024). Recurrent neural networks: A comprehensive review of architectures, variants, and applications. Information, 15(9): 517. https://doi.org/10.3390/info15090517

[17] Abood, R.H., Hamad, A.H. (2024). Multi-label diabetic retinopathy detection using transfer-learning-based convolutional neural network. Fusion: Practice and Applications, 17(2): 279-293. https://doi.org/10.54216/fpa.170221

[18] Obaid, M.H., Hamad, A.H. (2024). Internet-of-Things-based oil pipeline spill-detection system using deep learning and the LAB color space. Iraqi Journal for Electrical and Electronic Engineering, 20(1): 137-148. https://doi.org/10.37917/ijeee.20.1.14