Soft Computing (2023) 27:13039–13075
https://doi.org/10.1007/s00500-021-06608-1

FOCUS

Deep learning approaches for detecting DDoS attacks: a systematic review

Meenakshi Mittal[1] · Krishan Kumar[1] · Sunny Behal[2]

Accepted: 23 November 2021 / Published online: 27 January 2022

Abstract

In today's world, technology has become an inevitable part of human life. In fact, during the Covid-19 pandemic, everything from the corporate world to educational institutes has shifted from offline to online. It leads to exponential increase in intrusions and attacks over the Internet-based technologies. One of the lethal threat surfacing is the Distributed Denial of Service (DDoS) attack that can cripple down Internet-based services and applications in no time. The attackers are updating their skill strategies continuously and hence elude the existing detection mechanisms. Since the volume of data generated and stored has increased manifolds, the traditional detection mechanisms are not appropriate for detecting novel DDoS attacks. This paper systematically reviews the prominent literature specifically in deep learning to detect DDoS. The authors have explored four extensively used digital libraries (IEEE, ACM, ScienceDirect, Springer) and one scholarly search engine (Google scholar) for searching the recent literature. We have analyzed the relevant studies and the results of the SLR are categorized into five main research areas: (i) the different types of DDoS attack detection deep learning approaches, (ii) the methodologies, strengths, and weaknesses of existing deep learning approaches for DDoS attacks detection (iii) benchmarked datasets and classes of attacks in datasets used in the existing literature, and (iv) the preprocessing strategies, hyperparameter values, experimental setups, and performance metrics used in the existing literature (v) the research gaps, and future directions.

Keywords Deep learning · Distributed Denial of Service attacks · Datasets · Performance metrics

1 Introduction

In today's fast paced world, one cannot imagine life without Internet, which is required in diverse fields, namely, communication, education, business shopping, and the list is infinite. Despite its many advantages, many crimes have proliferated over the internet, viz. the spreading of misinformation, hacking, attacks, etc. The Denial of Service (DoS) attack occurs when the service (s), machine (s) or network (s) are made unavailable to its legitimate users (https://www.cloudflare.com/en-in/learning/ddos/glossary/denial-of-service/).
The DDoS attack is the subcategory of DoS attack and it occurs when the attacker compromises multiple computing devices to interrupt the regular traffic of a targeted victim (https://www.cloudflare.com/en-in/learning/ddos/what-is-a-ddos-attack/). In February 2021, the Cryptocurrency exchange EXMO was directed with 30 GB of traffic per second and it was unavailable for 2 h (https://portswigger.net/daily-swig/uk-cryptocurrency-exchange-exmo-knocked-offline-by-massive-ddos-attack; Han et al. 2012). In December 2020, the popular website tracker Down Detector had claimed many outages because of DDoS attacks (https://www.livemint.com/technology/apps/google-services-youtube-gmail-google-drive-face-outage-11607947475759.html). The other DDoS attacks that happened in 2018–2020 are detailed in (https://www.livemint.com/technology/apps/google-services-youtube-gmail-google-drive-face-outage-11607947475759.html; https://www.vxchnge.com/blog/recent-ddos-attacks-on-companies; https://www.thesslstore.com/blog/largest-ddos-attack-in-history/; https://securelist.com/ddos-report-q4-2019/96154/).

According to NETSCOUT's ATLAS Security Engineering & Response Team (ASERT), in the first quarter of 2021, approximately 2.9 million DDoS attacks were launched by

Communicated by Oscar Castillo.

✉ Meenakshi Mittal
meenakshi.cup@gmail.com

[1] UIET: University Institute of Engineering and Technology, Chandigarh, India

[2] Shaheed Bhagat Singh State University, Ferozepur, Punjab, India

the threat actors, and it is a 31% increase from the same time in 2020 (https://www.netscout.com/blog/asert/beat-goes). It hence proves that it is essential to detect DDoS attacks.

The above-cited incidents necessitate the need for an effective method to detect DDoS attacks. There are many techniques, viz. Statistical, Shallow Machine Learning, the Deep Learning, etc., to detect DDoS attacks. Of these techniques, Deep learning technique is suitable to detect DDoS attacks. The rest of these methods have limitations that have been explored and are explained as below:

- Statistical Methods Limitations: The statistical-based detection methods work on the basis of the previous knowledge of network flow (Catak and Mustacoglu 2019). But in today's world, malicious network flows are becoming a changing target. Hence, it is a challenging task to characterize the network traffic correctly. Most of the statistical DDoS detection methods are highly dependent on various user-defined thresholds. Hoque et al. (2017). Therefore, those thresholds need to be modified dynamically to be up to date with changes in a network. Hoque et al. (2017). The entropy measure of statistical methods requires extensive network awareness and experimentations to choose suitable statistical characteristics (Li and Lu 2019). To detect DDoS attacks, an entropy method, the Shannon entropy is used and this entropy detection uses only one feature like source IP address to create the detection model. Henceforth, attackers can easily manipulate source IP address using tools like scapy, hping, etc. Thus, the diversity of this feature to detect DDoS attacks is not a reliable source (Catak and Mustacoglu 2019). Most of the statistical approaches like entropy, correlation, etc., take excessive computational time throughout DDoS attack detection. Therefore, they cannot be carried out in real time (Hoque et al. 2017).
- Shallow Machine Learning (SML) Limitations: It works well by using the rules over a small amount of data. The SML identifies the attacks based on statistical features (Yuan et al. 2017) and then determines the class or value. It also requires regular updating of the model (Yuan et al. 2017) corresponding to the changes in attacks. The SML approaches solve the problem by breaking it into small subproblems and solves subproblems, and gives the final result (Xin et al. 2018). In SML some algorithms take less time in training and a long time in testing (Xin et al. 2018).

The DL methods are suitable to detect DDoS attack as: The DL methods can do feature extraction as well as classify the data. In today's world, there is a requirement for a detection system that can deal with the unavailability of data. Although the label for legitimate traffic is generally available, the availability of labelled malicious traffic is less. The DL approaches can extract the information from incomplete data (Van et al. 2017). The DL approaches are suitable to identify the low-rate attacks. Historical information is required to identify low-rate attacks (Yuan et al. 2017) and the DL approaches can learn long-term dependencies of temporal patterns (Vinayakumar et al. 2017). Thus, the DL approaches are useful in such a situation. The DL approaches have complex mathematical operations that are executed through multiple hidden layers using many parameters during the training phase (Aldweesh et al. 2020). The DL approaches use many matrix operations as compared to traditional machine learning approaches. GPU is efficient in doing well with matrix operations, and the availability of GPU machines makes it computationally efficient and fast.

Also, quantum computing has been very promising in many areas viz: artificial intelligence (AI), cybersecurity, medical research, etc. The possibilities of applying quantum computing in AI is to create quantum algorithms that perform better than classical algorithms and can be used for learning, decision problems, quantum search, and quantum game theory (https://research.aimultiple.com/quantum-ai/). In AI, to tackle more complex problems, quantum computing can provide a computation boost. It can be used for fast training or other improvements in SML and DL models (https://research.aimultiple.com/quantum-ai/). Thus, quantum computing extends the capabilities of deep learning by solving complex problems that involves large datasets and high computational requirements.

The abbreviations used in this article are summarized in Table 1. This article has been compared with other review articles, and a detailed comparison is provided in Table 2. It has been observed from Table 2 that most of the existing review articles do not discuss the preprocessing strategies, strengths, and types of attack used from the datasets in the existing literature. Our systematic review differs from the existing reviews described in Table 2 as we present the various types of DDoS attack detection DL approaches. Moreover, as per the research undertaken, there is no systematic literature review that covers DDoS attacks detection using the DL approaches.

In this paper, we have used the SLR protocol to review the DDoS attacks detection system based on DL approaches and have contributed the following findings:

- The state-of-the-art DDoS attack detection Deep learning approaches have been identified and categorized based on common parameters.
- The methodologies, strengths, and weaknesses of existing deep learning approaches for DDoS attacks detection have been summarized.
- The available DDoS benchmarked datasets and classes of attacks in datasets used in the existing literature have been summarized.

Table 1 List of abbreviations

Acronym	Meaning	Acronym	Meaning
AE	Auto Encoder	MLP	Multilayer Perceptron
ANN	Artificial Neural Network	MSDA	Marginalized Stacked De-noising Auto-encoder
BOW	Bag of Word	MSE	Mean Squared Error
CH	Cluster Head	NB	Naïve Bayes
CIC	Canadian Institute of Cyber security	NID	Network Intrusion Detection
CL	Convolutional Layer	NIDS	Network Intrusion Detection System
CNN	Convolutional Neural Network	NN	Neural Network
DDoS	Distributed Denial of Service	NS2	Network Simulator 2
DL	Deep Learning	OBS	Optical Burst Switching
DNN	Deep Neural Network	PCA	Principal Component Analysis
DoS	Denial of Service	PCC	Pearson's Correlation Coefficient
DT	Decision Tree	PDR	Packet Delivery Ratio
FC	Fully Connected	PL	Pooling Layer
FCNN	Fully Connected Neural Network	RBF	Radial Basis Function
FFBP	Feed-Forward Back-Propagation	RE	Reconstruction Error
FNR	False Negative Rate	ReLu	Rectified Linear Unit
FPR	False Positive Rate	ResNet	Residual Network
GD	Gradient Descent	RF	Random Forest
GPU	Graphics Processing Unit	RMS	Root Mean Square
GRU	Gated Recurrent Unit	RNN	Recurrent Neural Network
GT	Game Theory	RQ	Research Question
IDS	Intrusion Detection System	SCC	Sparse Categorical Cross-entropy
IoT	Internet of Things	SDN	Software Defined Network
IP	Internet Protocol	SGD	Stochastic Gradient Descent
KNN	k-Nearest Neighbours	SLR	Systematic Literature Review
LR	Logistic Regression	SML	Shallow Machine Learning
LSTM	Long short-term memory	SVM	Support Vector Machine
LUCID	Lightweight Usable CNN in DDoS Detection	TCP	Transmission Control Protocol
MANETs	Mobile Ad-hoc Networks	TL	Transfer Learning
MCC	Mission Control Center	TNR	True Negative Rate
MKL	Multiple Kernel Learning	TPR	True Positive Rate
MKLDR	Multiple Kernel Learning for Dimensionality Reduction	UDP	User Datagram Protocol
ML	Machine Learning	WSN	Wireless Sensor Network

- Focus has been on the preprocessing strategies, hyperparameter values, experimental setups, and performance metrics that the existing deep learning approaches have used for DDoS attacks detection.
- The paper aims at highlighting the research gaps, and points at the future directions in this area.

The rest of the paper is organized as follows: Sect. 2 explains the SLR protocol; Sect. 3 talks about the state-of-the-art DDoS attacks detection DL approaches used in the existing literature; Sect. 4 analyses the methodologies, strengths, and weaknesses of the existing literature; Sect. 5 describes the details about the available DDoS benchmarked datasets and classes of attacks in datasets that are used in the existing literature; Sect. 6 provides the details about the preprocessing strategies, hyperparameter values, experimental setups, and performance metrics; Sect. 7 illustrates the research gaps in the existing literature; and Sect. 8 explicates the conclusion and future directions of this review article.

Table 2 A detailed comparison with other review articles: (✓: Yes, : No)

Review article	Ferrag et al. (2020)	Aleesa et al. (2020)	Ahmad et al. (2021)	Gamage and Samarabandu (2020)	Ahmad and Alsmadi (2021)	This article
Focused security domain	Cyber security intrusion detection	IDS	IDS	NID	IoT security	DDoS
ML/DL	DL	DL	ML/DL	DL	ML/DL	DL
Systematic study		✓	✓		✓	✓
Taxonomy		✓	✓	✓		✓
Preprocessing strategy						✓
Types of attack used in existing literature from the datasets		✓				✓
Strengths			✓			✓
Weaknesses			✓	✓	✓	✓
Research gaps						✓

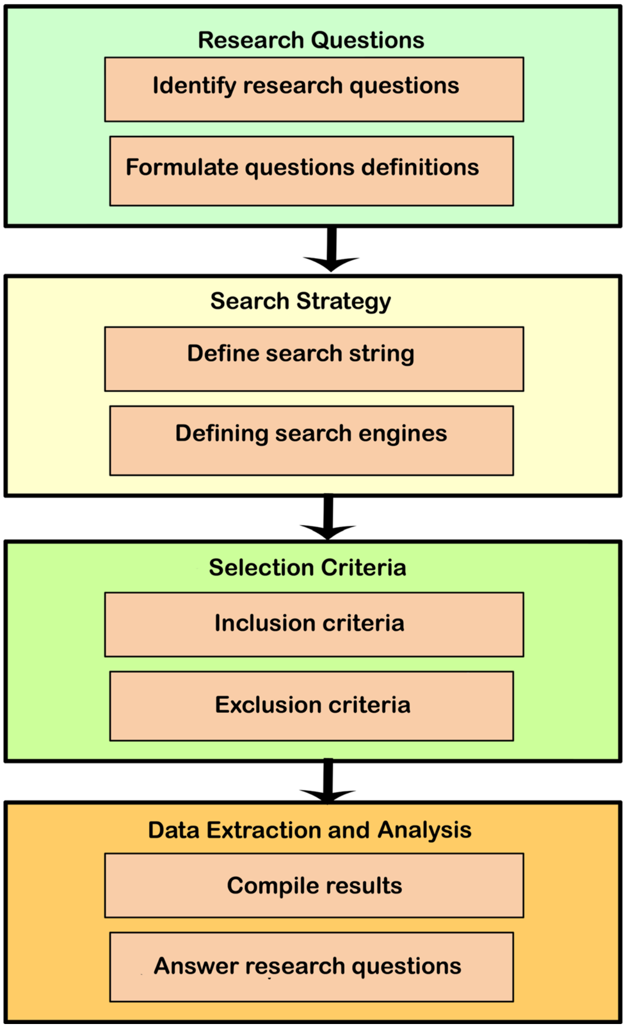


Fig. 1 Survey protocol overview

2 Systematic literature review protocol

SLR provides a comprehensive approach towards understanding the problem and is considered an effective method in evaluating the literature related to the problem. A step-by-step methodology is adopted for conducting the research in systematic surveying. The SLR survey in this research work follows the guidelines of Keele et al. (2007). This work focuses on DDoS attacks detection using deep learning-based solutions, published from 2018 to 2021. The result of SLR provides a set of research articles that are categorized based on the taxonomy of DL approaches used. The purpose of SLR is to figure out various research gaps in the existing literature that provide promising future research directions. Figure 1 shows the overview of the survey protocol, and it is explained step-by-step as below.

2.1 Research questions

The main objective of the systematic review is to outline the research questions and to answer them after evaluating the data taken out from the list of final selected research papers. Research questions that have been addressed in this work are given as below:

RQ1: What are the state-of-the-art DDoS attacks detection DL approaches and how can these approaches be categorized?
RQ2: What are the methodologies, strengths, and weaknesses of existing deep learning approaches for DDoS attacks detection?
RQ3: What are the available DDoS benchmarked datasets and classes of attacks in datasets that have been used in the existing literature?
RQ4: What are the preprocessing strategies, hyperparameter values, experimental setups, and performance metrics that the existing DL approaches have used for DDoS attack detection?
RQ5: What are the research gaps in the existing literature?

2.2 Search strategy

A systematic survey is initialized by forming a suitable search strategy. A proper search strategy is the pre-requisite to any research. Therefore, a suitable set of databases has been selected to mine out the appropriate literature. In the present research work, search was carried out in two phases from 2018 to 2021. Phase 1 of the search consisted of four digital libraries: ACM digital library, IEEE Explore, Springer, Science Direct, and Phase 2 included Google Scholar academic search engine. The addition of Google Scholar has helped in preventing the omission of any relevant literature. In addition, a pilot study was also carried out to refine the search string. Ten most cited and suitable articles have been selected from a set of pre-collected articles kept in the database during the pilot study. One common search query that was performed with little modification in different digital libraries is:

(Detection of DDoS attacks using deep learning **OR** DDoS attack detection using deep learning approaches)

The results obtained from the chosen digital libraries were refined by " filtering options." Figure 2 depicts the flow of various steps of the survey protocol.

2.3 Study selection criteria

The main objective of study selection was to exclude any irrelevant literature concerned with the defined RQs. This was done with the help of addition and elimination criteria. Besides, the research articles which extended the previous related work were included. The search phase 1 produced 3039 entries, and from search phase 2, we have taken only the first 1000 entries, making 4039 entries in stage 1. Out of 4039, 178 duplicate entries were removed in stage 2. Then stage 2 is followed by removal of articles according to the titles (3130), abstract (581), and full texts (118), respectively, in subsequent stages. Finally, 32 research articles were selected after stage 5. The inclusion and exclusion criteria were specified to eliminate the research studies that are not related to the defined research questions. The inclusion/exclusion criteria used are defined as below:

Inclusion criteria:

- All articles that provide a new approach for DDoS attacks detection using deep learning.
- All studies that focus on only deep learning approaches.
- Studies that are closely associated but vary in essential parameters were included as distinct primary studies.
- Studies that fulfil the research questions.
- Studies that extend the previous related work.
- The articles were published between 2018 and 2021.

Exclusion criteria:

- Articles not in the English language.
- Articles not related to the research topic.
- Review articles, Editorials, Discussion, Data articles, Short communications, Software publications, Encyclopedia, Poster, Abstract, Tutorial, Work in progress, Keynote, Invited talk.
- Articles did not demonstrate an adequate amount of information.
- Duplicate research studies.

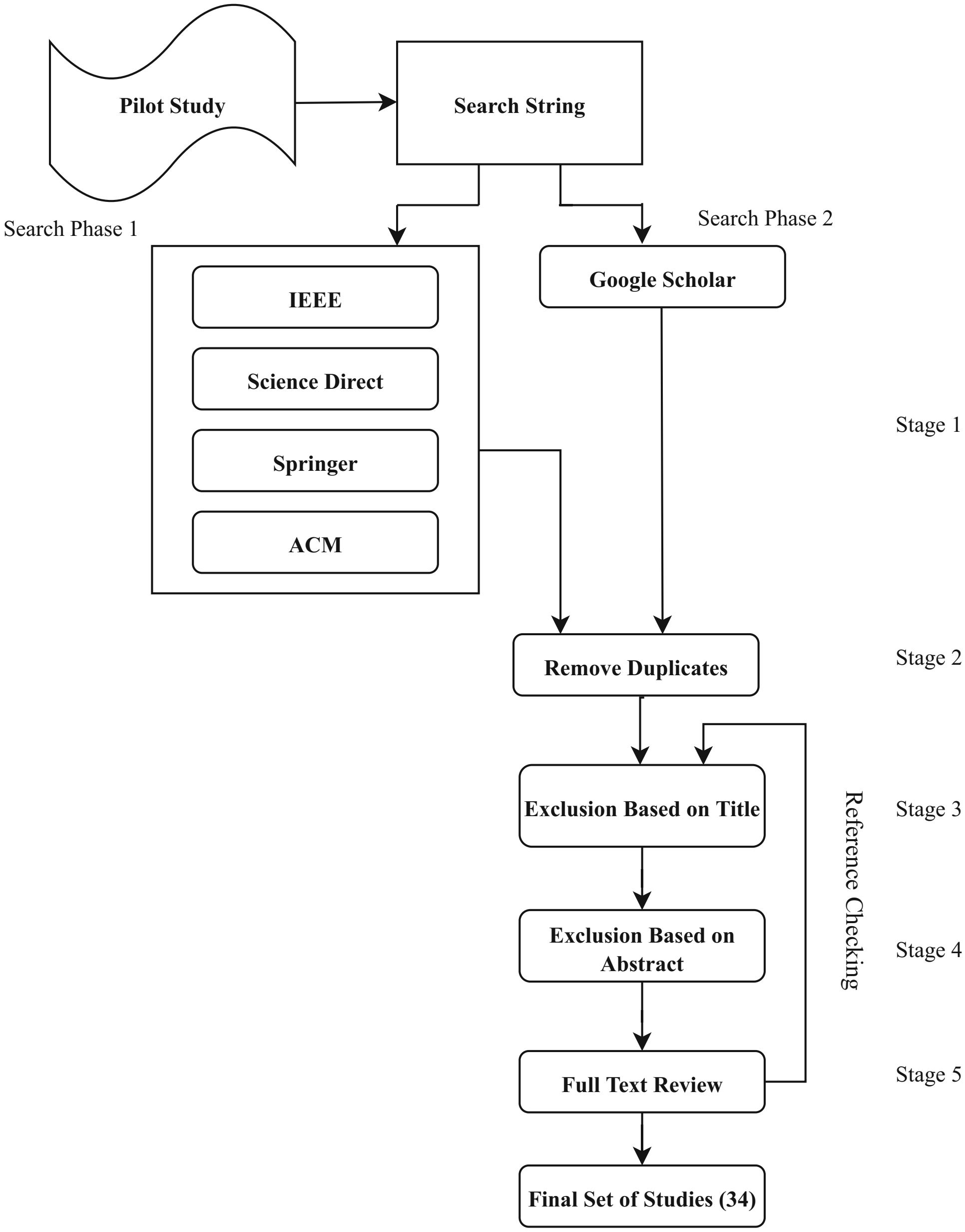
2.4 Reference checking

The references of 32 articles obtained after reviewing the full texts were evaluated to prevent the omission of any relevant work. The results (76 articles) obtained were then moved to inclusion and exclusion criteria for further assessment based on title, abstract, and full text. Then articles based on titles (11), abstract (51), and full text (12) were removed in subsequent stages. In the end, two articles were finalised after removing 74 articles through reference checking.

2.5 Data extraction

The required data were extracted after studying the text of the complete article based on the research questions. The data extracted from each study is used to fill a predesigned form. This form consists of various field entries, including title, the approach used, datasets used, number of features, attack and legitimate classes identified, preprocess-

Fig. 2 Systematic literature review process



ing strategy, experiment setup/performance optimization of the model, performance metrics, strength, weakness, and the summary which is used to critically analyze the final set of articles to simplify the responses to the research questions. The details of data extraction fields are given in Table 3.

3 State-of-the-art DDoS attack detection Deep learning approaches

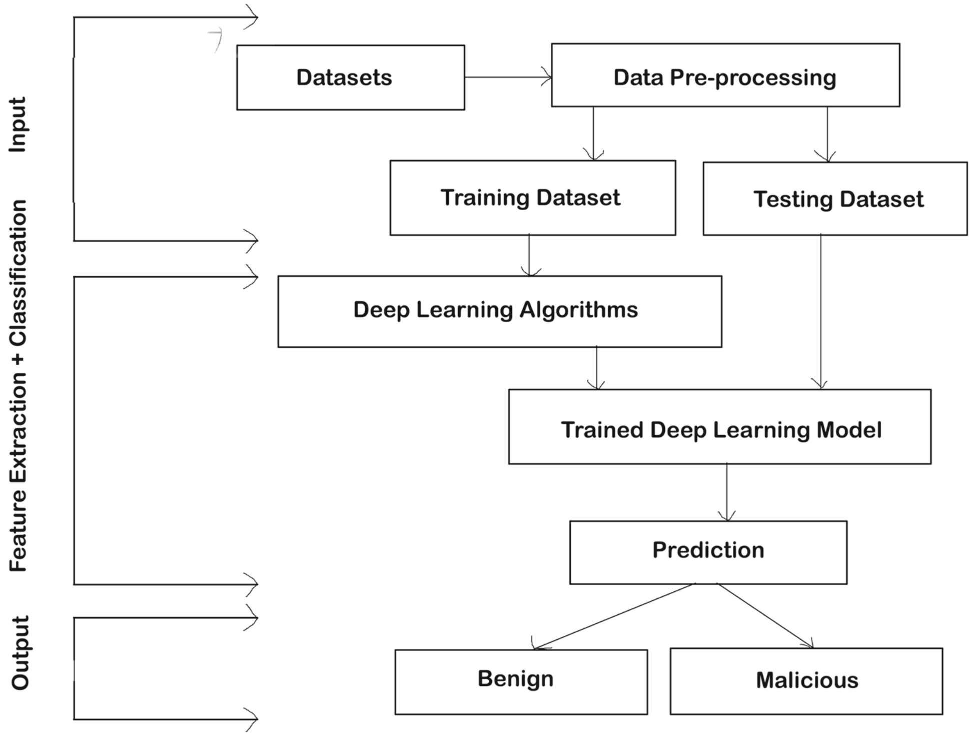
Deep learning is defined as the subset of ML in artificial intelligence (https://www.investopedia.com/terms/d/deep-learning.asp) with the capabilities of learning from supervised or unsupervised data. Deep learning uses multilayer networks; therefore, it is also called as deep neural network or deep neural learning (Aldweesh et al. 2020). The layers are linked through neurons, representing the mathematical calculation of the learning processes (Goodfellow et al. 2016).

As shown in Fig. 3 DL algorithms take the preprocessed data as input and do both feature extraction as well as classification and predict the samples as benign or malicious as output. The taxonomy contains the five categories of DL models for DDoS attacks detection based on common parameters of the DL approaches. The taxonomy of DL is shown in Fig. 4. The DL methods have been classified into five categories that are supervised instance learning, supervised sequence learning, semi-supervised learning, hybrid learn-

Table 3 Data extraction fields

Field	Objective
Title	Provides the title of research paper
Approach used	List the different approaches related to the DL used in the paper
Datasets	List the different datasets used in the paper for the evaluation purpose
Number of features	List the selected features from the datasets
Attack and legitimate classes identified	Provides the name of attacks used in the paper
Preprocessing strategy	Describes the preprocessing processes used before training the model
Experiment setup/performance optimization of the model	Explains about how the experiment is done/list different parameter values of the model at which it gives best performance
Performance metrics	Provides the results and through these metrics we can compare one model with another model
Strength	List the good points about the model
Weakness	List the weak points of the model
Summary	A concise explanation about the above fields

Fig. 3 A deep learning process



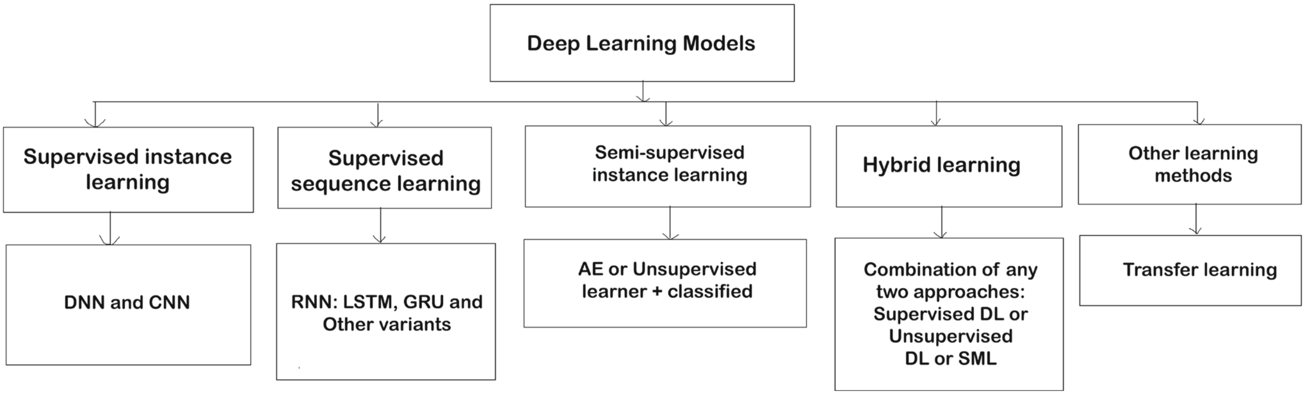

Deep Learning Models
Supervised instance learning
Supervised sequence learning
Semi-supervised instance learning
Hybrid learning
Other learning methods
DNN and CNN
RNN: LSTM, GRU and Other variants
AE or Unsupervised learner + classified
Combination of any two approaches: Supervised DL or Unsupervised DL or SML
Transfer learning

Fig. 4 Taxonomy of deep learning models

ing, and other learning methods. The following is the brief description of each category:

(1) *Supervised instance learning* Supervised instance learning uses the flow of instances (Gamage and Samarabandu 2020). It uses the labelled instances for training purposes. The following is the description of the most commonly used methods in this category:

- *Deep neural networks* A DNN is an artificial neural network with more than two hidden layers between the input layer and the output layer (Yuvaraj et al. 2020). The conventional neural networks have two or less than two hidden layers compared to the deep neural networks (Subasi 2020).
- *Convolutional neural network* The CNN consists of convolutional, pooling, flattening, and FC layers (https://www.ibm.com/cloud/learn/convolutional-neural-networks). The convolutional layer is the main constructing block of CNN (Gopika et al. 2020). The convolution layer performs the mathematical operation (https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/) by applying the filters to the input to produce a convoluted feature or feature map. The filters are applied in a moving window manner over the height, width and depth of the input. The pooling layer followed the convolution layer (Gopika et al. 2020). It is used to reduce the dimensionality of feature maps (Zhu et al. Jan 2018; Ke et al. 2018) by taking a maximum or minimum value from a given area. The flattening layer is used to change the multidimensional data in pooling layer, to 1-D vector to input into a FC layer. The FC layer determines the probability of each class label to classify the samples (Yamashita et al. 2018).

(2) *Supervised sequence learning* The supervised sequence learning uses a sequence of flows (Gamage and Samarabandu 2020). In this type, the models learn from the series of data by keeping the previous input states in the memory. The most commonly used models in this type are described as below:

- *Recurrent neural networks (RNN)* The feed-forward neural network comprises of the input, hidden, and output layers. In feed-forward neural networks all inputs and outputs are independent of each other (Nisha et al. 2021), and thus, it cannot use the previous information. Therefore, it is not suitable in case of next word prediction of a sentence (https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-9e5eb8049c9). In RNN the output from the previous step is given to the current step in addition to the current input, and thus, it can predict the next word of a sentence by retaining the previous information. But the RNN has disadvantages of gradient vanishing problems, exploding problems (Nisha et al. 2021) and to process the long sequential data using RNN (https://www.mygreatlearning.com/blog/types-of-neural-networks/).
- *Long short-term memory* (https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/) The problem of RNN has been solved by the LSTM. The LSTM network comprises different memory blocks or cells. The two states, i.e. hidden state and the cell state, are given to the next cell. The memory blocks can select which information to remember or to forget through the three mechanisms called gates, i.e. forget, input, and output gates (https://purnasaigudikandula.medium.com/recurrent-neural-networks-and-lstm-explained-7f51c7f6bbb9). A forget gate eliminates the information from the cell state which is no longer necessary for the LSTM. The input gate adds the information to the cell state and the output gate is responsible for extracting valuable information from the present cell state and treated it as an output.

- *Gated recurrent unit* (Alom et al. 2018) In the GRU the forget and input gates are combined into an update gate and merged the cell state and hidden state along with a few different changes.

(3) *Semi-supervised learning* Semi-supervised methods use the pre-training stage using unlabelled data (Gamage and Samarabandu 2020). It uses both labelled and unlabelled records for training a model. In this, autoencoder has been used for extraction of features, and other deep or shallow machine learning models are used for the classification.

- *Autoencoders* (Aldweesh et al. 2020) An AE is a deep neural network used for dimensionality reduction and feature extraction. An AE comprises of input (for encoding) and output (for decoding) layers along with the hidden layer. AE trains the encoder and decoder collectively using back-propagation. The encoder extracts the raw features and converts the input into low-dimensional abstraction. The decoder then reconstructs the original features from the low-dimensional notion.

(4) *Hybrid learning* It uses the combination of any two approaches, i.e. supervised DL or unsupervised DL or shallow machine learning. In the existing literature, many researchers have used CNN-LSTM (Roopak et al. 2019, 2020; Nugraha and Murthy 2020), LSTM-Bayes (Li and Lu 2019), RNN-AE (Elsayed et al. 2020), etc.

(5) *Other learning methods* Under this category comes transfer learning. A transfer learning method uses the already pre-trained model from a repository (Gamage and Samarabandu 2020). The researchers have used the deep learning approaches to train them on one attack domain and later used that trained model on another domain.

4 Methodologies, strengths, and weaknesses

In this section, the methodologies, strengths, and weaknesses in the existing paper have been briefed according to the proposed taxonomy:

(1) *Supervised instance learning*

- *Deep neural networks:*
  Sabeel et al. (2019) proposed two ML models, DNN and LSTM, for the prediction of unknown DoS/DDoS attacks. In this paper, authors first trained their models on the preprocessed DoS/DDoS samples in the CICIDS2017 dataset and then evaluated the results on the synthesized ANTS2019 dataset to measure the accuracy. In the second part, the authors have merged the synthesized dataset with the CICIDS2017 dataset. The models are then retrained and the detection performance to newly synthesized unknown attacks is evaluated. The performance of these models have showed great enhancements on the second part of the experiment, i.e. DNN and LSTM achieving an accuracy of 98.72% and 96.15%, respectively. The DNN and LSTM have AUC values of 0.987 and 0.989, respectively. The dataset ANTS2019 has been created synthetically to mimic real-life attacks. The binary class classification has been done and the real-time detection setup has not been used.

  In the private cloud, DDoS is one of the causes to degrade the services. The focus of Virupakshar et al. (2020) is on bandwidth and connection flooding types of DDoS attacks. Authors have used DT, KNN, NB, and DNN algorithms for the detection of DDoS attacks in the OpenStack-based cloud. The authors have also compared several classifiers and selected the model with the best precision and accuracy. DNN model has been chosen as it has higher accuracy and precision value when the dynamically generated dataset is being used. DNN classifier achieved 96% precision and higher accuracy for cloud datasets than DT, KNN, NB. Authors have used an old dataset, i.e. KDDCUP99, and also, there is no detail given about the LAN and cloud dataset. The precision value of the DNN algorithm is less for the KDDCUP99 dataset compared to other algorithms.

  Asad et al. (2020) introduced DNN architecture (i.e. DeepDetect). It is based on feed-forward back-propagation architecture. The authors proposed this model to protect the services from the application layer DDoS attacks. The proposed approach is evaluated using the CICIDS2017 dataset for DDoS detection. The authors have compared their method with RF and DeepGFL. The DeepDetect yielded F1-score value of 0.99 and outperformed the other approach. Also, the AUC value is so close to 1, that it shows the high accuracy achieved by the proposed model. In this article researchers have done multiclass classification and this approach has been deployed on the cloud as a web service to provide security from application-layer DDoS attacks. This approach has been evaluated only on the Application layer DDoS attacks.

  Muraleedharan and Janet (2020) proposed a flow data-based deep neural classification model to detect slow DoS attacks on HTTP. The classification model used a FC feed-forward deep network. The model is evaluated on the CICIDS2017 dataset in which only the DoS samples have been selected for the model. The classifier can detect the type of DoS attacks. The results obtained illustrate that the model can classify the attacks with an overall accuracy of 99.61%. This approach has evaluated only HTTP slow DoS attacks (Slowloris, SlowHTTP, Hulk, GoldenEye) over the CICIDS2017 dataset.

Sbai and El Boukhari (2020) proposed a DL model DNN (with two hidden layers and 6- epochs) to detect data flooding or UDP flooding attack in MANETs, by using the dataset CICDDoS2019. The authors trained and evaluated the model with the CICDDoS2019 dataset. The proposed model obtained results that are: Recall: 1, precision: 0.99, F1-score: 0.99, Accuracy: 0.99, which are very promising. In this article, the authors have worked only on the data flooding or UDP flooding attack of the CICDDoS2019 dataset.

Amaizu et al. (2021) proposed an efficient DL-based DDoS attack detection framework in 5G and B5G environments. The proposed framework is developed by concatenating two differently designed DNN models, coupled with a feature extraction algorithm, i.e. PCC. It is built to detect the DDoS attacks and the type of DDoS attacks encountered. The authors evaluated the proposed framework using four different scenarios over an industry-recognized dataset (i.e. CICDDoS2019 dataset). Results illustrated that the framework could detect DDoS attacks with an accuracy of 99.66% and a loss of 0.011. Furthermore, the proposed detection framework results were compared with the existing approaches, i.e. KNN, SVM, DeepDefense, and CNN ensemble. The proposed framework outperformed all except the CNN ensemble. The CNN ensemble has better precision and recall than the proposed framework. The proposed model has a complex structure so it can take more detection time and thus can affect the model's performance in a real-time scenario.

Cil et al. (2021), proposed the DL model that contains both feature extraction as well as classification processes in its structure. The DNN model consists of an input layer with 69 units, three hidden layers consist of an equal amount of 50 units and two units are used in the output layer. The authors have divided the dataset CICDDoS2019 into two datasets, i.e. Dataset1 and Dataset2. Dataset1 is categorized as two types of traffic: normal and attack traffic. Dataset2 is created to define the types of DDoS attacks. DNN model has nearly 100% accuracy for DDoS attack detection on Dataset1 and thus the DNN model has achieved the reliable result for early action, suitable for real time scenarios. Also, it successfully classifies DDoS attacks with approximately 95% of accuracy on the Dataset2. The proposed model gives less accuracy in the case of multiclass classification.

- *Convolutional neural network*

The Optical Burst Switching (OBS) network is usually victimized by DDoS attacks, known as Burst Header Packet (BHP) flooding attacks. According to Hasan et al. (2018) because of a minimal number of records of the datasets, conventional machine learning techniques such as NB, KNN, and SVM cannot examine the data efficiently. Therefore, the authors have proposed a Deep CNN model. The results showed that the proposed method outperformed the three ML methods for a given dataset with fewer features. In this multiclass Classification has been done and the model has been evaluated over 11 performance metrics and obtained good results. The dataset used to evaluate the proposed model has a small number of instances and does not contain all traffic types.

In the paper, Amma and Subramanian (2019) a Vector Convolutional Deep Feature Learning (VCDeepFL) approach to identify DoS attacks has been introduced. The VCDeepFL approach is a combination of Vector VCNN and FCNN. The proposed method has two phases, i.e. training and testing. The training phase consists of pre-training using unsupervised learning, i.e. VCNN, and training using supervised learning, i.e. FCNN. VCNN uses the vector form and the FCNN has been trained using the features from the pre-training module. FCNN is a multiclass classifier. The testing is done using the weights which are learned during the training phase in VCDeepFL. The proposed approach has been tested over the NSL KDD dataset and compared with the base classifiers (MLP, SVM) and state-of-the-art attack detection systems. It has been observed from the results that the proposed approach achieved high accuracy, low false alarm, and improved detection rate compared to base classifiers and the state-of-the-art attack detection system. In this study, the old dataset has been used and the authors have not shown the experiments for detecting unknown attacks.

Chen et al. (2019) proposed a DAD-MCNN (i.e. multichannel CNN) framework to detect DDoS attacks. The number of feature groups decides the number of channels. The authors have split the features into different levels, like packet level, host level, and traffic level. The authors have used the incremental training approach to train MC-CNN. The authors have conducted a sequence of tests over KDDCUP99, CICIDS2017 datasets for binary classification in both datasets and multiclass category in KDDCUP99 only. They also compared MC-CNN with CNN, LSTM (3 layers), and other shallow ML methods (RF, SVM, C4.5, and KNN). The results showed that MC-CNN outperformed the state-of-art methods for all binary and multiclass classification. Further, the authors have also changed the training dataset size and evaluated the CNN and MC-CNN. The results showed that MC-CNN is better in the restricted dataset and helpful in building DDoS detection systems when the training data are relatively insufficient. There is no much difference in the results of multichannel and single-channel models. Also, the multichannel models will increase the complexity and thus might not be suitable when validated over real-time scenarios.

In Shaaban et al. (2019), the CNN model has been proposed to detect DDoS attacks. Authors have compared their proposed model with the classification algorithms like DT, SVM, KNN, and NN over two datasets, i.e. dataset 1 (simulated network traffic) and dataset2 (NSL-KDD). It has been observed that the proposed model performed well compared to the other four classification algorithms such as like DT, SVM, KNN, and NN and gives an accuracy of 99% on both datasets. In this approach one-column padding has been used to convert the data into matrix form. Thus it can affect the learning of the model.

Haider et al. (2020) proposed a deep CNN framework for the detection of DDoS attacks in Software Defined Networks, and this proposed ensemble mechanism has been evaluated over the CICIDS2017 dataset. This solution is compared with the state-of-the-art DL-based ensembles and hybrid approaches (i.e. RNN, LSTM, RL). The ensemble CNN performed better than other three proposed DL-approaches, but there is a trade-off between their training and testing time. The authors have also compared the proposed ensemble CNN approach with existing competing approaches. The results showed that the ensemble CNN approach outperformed the existing competing approaches. The ensemble CNN has achieved an accuracy of 99.45%. This approach has training and testing times higher than other approaches. Thus, it can affect the mitigation mechanism. Therefore, attacks can cause more damage.

Wang and Liu (2020) proposed an information entropy and DL method to detect DDoS attacks in SDN environment. Thus, the technique uses two-level detection for the identification of the attacks. Firstly, the controller will inspect the suspicious traffic through information entropy detection. A CNN model will then execute the detection based on the fine-grained packet to distinguish among normal traffic and attack traffic. The authors have compared their method with the DNN, SVM, and DT. The CNN achieved higher precision, accuracy, F1-score, and recall among them. The accuracy of it is 98.98%. 1. The ROC curve of CNN is steeper than DNNs, SVM, and DT. The AUC of CNN is 0.949. There is a need to set the threshold value for the detection method based on information entropy.

Kim et al. (2020) developed a CNN-based model to detect DoS attacks using the records of DoS attacks in CSE-CIC-IDS 2018 and KDD datasets. Authors have designed their CNN model considering the number of CLs and kernel size. They evaluated their model by creating 18 scenarios considering hyperparameters, the type of image, i.e. greyscale or RGB, the number of CLs, and the kernel size. The authors have evaluated each scenario for both binary and multiclass classifications. They then suggested optimal scenarios with higher performance. The CNN model is also compared with RNN. The CNN model can identify specific DoS attacks with alike characteristics compared to the RNN model. It has also been found that kernel size in CNN has not significantly impacted both binary and multiclass classification. The preprocessing time of conversion of features to RGB and greyscale images has not been considered, as it matters in real-time validation.

LUCID technique (Doriguzzi-Corin et al. 2020) has been used to detect DDoS attacks, which helps in, lightweight execution with low processing overhead and detection time. Their unique traffic preprocessing mechanism is designed to feed the CNN model with network traffic for online DDoS attack detection. The authors compared LUCID with DeepDefense 3LSTM over ISCX2012, CIC2017, CSECIC2018, UNB201X and got comparable results. However, the LUCID outperforms 3LSTM in detection time. The performance of LUCID has been compared against state-of-the-art works (DeepDefense, TR-IDS, E3ML) and validated on ISCX2012. Also, compared the LUCID with state-of-the-art works (DeepGFL, MLP, LSTM, 1D-CNN, 1D-CNN + LSTM) and validated on CIC2017 Dataset. The evaluation results show that the LUCID matches the existing state-of-the-art performance. It has also been demonstrated the suitability of the model in resource-constrained environments. Their work has also proved that LUCID is learning the correct domain information by calculating each feature's kernel activations. The LUCID training time on the GPU development board is 40 times faster than the authors' implementation of DeepDefense 3LSTM. The feasibility test has also been done for the proposed approach. The padding has been used for making the size of each flow equal to n. By using padding, the CNN may get affected in learning the patterns. Also, there are trade-offs between accuracy and memory requirements. The pre-processing time has not been calculated as it is important for real-time scenarios.

In de Assis et al. (2020), the authors have proposed an SDN defence system. The defence system detects and mitigates DDoS attacks over the external targeted server and on the controller. The detection module detects attacks. In this module, the authors have used DL-based CNN method to detect DDoS attacks by inspecting the SDN traffic behaviour. The proposed method works in near real-time, as in this study, IP flow data have been extracted and analyzed in one-second intervals to reduce the DDoS effect over genuine users. The proposed CNN approach within the detection module has been compared with the other three anomaly detection approaches, i.e. the LR, the MLP network, and the Dense MLP. The authors have tested the above detection methods over two test scenarios, i.e. the first one uses simulated SDN data, and the second one uses CICDDoS 2019 dataset. The overall results showed that CNN is efficient in detecting DDoS attacks for all these test scenarios. A GT-based technique has been applied in the SDN controller to mitigate the attack in the mitigation module. The outcomes showed that the mitigation method efficiently restores the SDN's regular operation. The proposed system operates autonomously to allow the speed of the detection and mitigation processes. The model shows less accuracy for CICDDoS 2019 dataset.

Authors Hussain et al. (2020) have proposed a method to transform the non-image network traffic into three-channel image forms. It has been evaluated on the existing ResNet-18 model, a state-of-the-art CNN model, for detecting the recent DoS and DDoS attacks. The proposed method used the cleaned and normalized features to transform the data into images without using any encoding or transformation techniques. The authors also compared the proposed methodology using ResNet-18 with a state of art solution and outperformed it on the same dataset. The proposed methodology using ResNet-18 achieved 99.99% accuracy in binary class classification. It has also achieved an accuracy of 87.06% for the 11 types of DoS and DDoS attacks on the CICDDoS2019 dataset. The preprocessing time is not calculated for converting non-image data to image data as this is the important metric for real-time validation. Also, the transformation of the original 60*60*3 dimensions into 224*224*3 dimensions has not been described for the input to the ResNet model.

(2) *Supervised sequence learning*

- *Long short-term memory*

Li et al. (2018) proposed a deep learning model to detect DDoS attacks in SDN environment. The model comprises an input, forward recursive, reverse recursive, FC hidden layer, and output layers. RNN, LSTM, and CNN are also used in the model. Thus, the authors have formed four different models that are: LSTM, CNN/LSTM, GRU, 3LSTM. The accuracy of the DDoS attack by the use of the ISCX dataset is 98%. The DDoS attack detection and defence system are built using the ubuntu14.04 operating system, and the DDoS defence system is verified through real-time DDoS attacks. But tested on only limited types of real-time DDoS attacks that are the Ping Of Death attack, ARP flood inundation attack, SYN flood inundation attack, Smurf attack, and UDP flood inundation attack.

Priyadarshini and Barik (2019) have designed a DL-based model to protect from DDoS attacks in a fog network. The LSTM has been used to detect Network/Transport level DDoS attacks. The LSTM model's parameters are also varied and were implemented using two scenarios. The authors have produced the results by implementing the DL model over the CTU-13 Botnet and the ISCX2012 IDS datasets in the first scenario. In the second scenario, the DL model is trained with the Hogzilla dataset and is examined on 10% of it and a few real-time DDoS attacks. The authors compared the model with other approaches also. It has been observed that the LSTM model showed 98.88% of accuracy for all the test scenarios. DDoS defender module can block the infected packet from being transmitted to the cloud server through the OpenFlow switch present in SDN. In this article, no real-time feasibility analysis of the proposed has been done and only Network/transport-level DDoS attacks have been detected.

Liang and Znati (2019) have proposed the four-layered architecture model consisting of two LSTM layers, a dropout layer, and a FC layer. In this approach, the handcrafted feature engineering has been obviated, and network traffic behaviour has been learned directly from a small sequence of packets. This paper has carried out three experiments with three other algorithms (DT, ANN, SVM) over CICIDS 2017 Wednesday and Friday datasets. According to the results observed, Experiment 1 showed that the LSTM-based scheme successfully learned the complex flow-level feature descriptions embedded in raw input and performed well than other approaches. Experiment 2's result showed that the proposed scheme can capture the dynamic behaviours of unknown network traffic accurately. Experiment

3 concluded that permitting the model to test more packets for every flow, with increasing n values, no longer always enhances the performance. The proposed scheme outperforms traditional machine learning methods over unknown traffic. The proposed model uses a subsequence of n packets, i.e. $S \subset F$. If a flow does not have enough packets, S is padded with fake packets. These padding values can affect the learning of the proposed model and can cause performance degradation.

Shurman et al. (2020) proposed two methodologies the first method is a hybrid-based IDS, and the second method is a DL model based on LSTM to detect DoS/DDoS attacks. The first method, the IDS framework, defined as an application, can detect malicious network traffic from any network device with running datasets of IPs against it. It is capable of blocking unwelcome IPs. The second method used the LSTM and this model is trained on the CICDDoS2019 dataset with several types of DrDoS attacks. The second model is compared with other existing models. The results show that the model outperformed the other models. The LSTM-based model shows an accuracy of 99.19% on the reflection-based CICDDoS2019 dataset but only reflection-based CICDDoS2019 dataset has been used. Also, the hybrid IDS and LSTM methods are independent of each other.

- *Gated recurrent unit*

Assis et al. (2021) proposed a defence system against DDoS and intrusion attacks in SDN environment. The proposed system is consists of two essential modules, i.e. the detection and mitigation modules. The detection module detects attacks. In this module, the authors have used the DL-based GRU method to detect DDoS and intrusion attacks by analyzing single IP flow records. The mitigation module takes effective actions against the detected attacks. Authors have tested their proposed model against seven different ML approaches on two datasets, i.e. CICDDoS 2019 and the CICIDS 2018. These different ML approaches are DNN, CNN, LSTM, SVM, LR, KNN, and GD. The authors have taken two test scenarios, i.e. first for CICDDoS 2019 dataset and second for the CICIDS2018. In both scenarios, authors have tested their proposed model with other ML methods for accuracy, precision, recall, f-measure, the effectiveness of the methods' classification concerning normal and attack flows separately. The results showed that the GRU could detect DDoS and intrusion attacks for all these test scenarios. Furthermore, a feasibility test is also performed by calculating the average number of flows per second the detection methods can analyze and classify. This test is done using collected actual IP flow data from the State University of Londrina. The results pointed out that GRU is a viable proposed approach. The average results of the proposed approach including the accuracy, recall, precision, and f-measure for CICDDoS2019 and CICIDS2018 datasets are 99.94% and 97.09%, respectively. In this article, the detection and training times are not calculated and also the offline analysis of datasets has been done.

(3) *Semi-supervised learning*

Catak and Mustacoglu (2019) proposed a combination of two different models, i.e. AE and a deep ANN. The AE layer of the model learns the representation of the network flows. The DNN model tries to determine the exact malicious activity class. The authors have evaluated their model on the UNSWNB15 dataset and KDDCUP99 with different activation functions. The results obtained the best F1 results with ReLu activation function, i.e. 0.8985. The overall accuracy and precision for KDDCUP'99 are approximately 99% for activation functions softplus, softsign, ReLu, tanh. In this article, the focus is only on the activation functions.

Ali and Li (2019) have proposed the deep AE for feature learning and MKL framework for detection model learning and classification. The authors first trained the multiple deep AEs to learn features in an unsupervised manner from training data. Then, the features are automatically combined using the MKL algorithm called the MKLDR algorithm. It is then used to form a DDoS detection model in a supervised fashion. The proposed method has been evaluated on two datasets, i.e. ISCXIDS2012 and UNSW-NB15 and their subsets. Also, the proposed method is compared with NB, DT, KN, LSVM, RF, and LSTM. It has been observed that the accuracy of the proposed method is higher compared to other methods. The detection time of the proposed model is not calculated as the model is very complex and thus can take time to respond and thus, attacks can cause significant damage to the system.

Yang et al. (2020) have designed a five-layered AE model for an effective and unsupervised DDoS detection. It requires only normal data for building the detection model. Then this model classifies the traffics into the attack and normal. Authors have demonstrated through experiments over different datasets (i.e. public datasets synthetic dataset) that the knowledge learned from one network environment cannot be applied to another. Also showed that one of the supervised ML approaches, i.e. DT, cannot effectively detect new attacks which have not appeared in its train set. Still, the AE performed well on unknown and new attacks. The authors also demonstrated that the results of AE-based DDoS attacks Detection

Framework (AE-D3F) with 27 features and the sixteen selected features with PCC on the datasets achieved a comparable performance while using fewer features. This approach used only normal traffic to train the model and is helpful for the unavailability of labelled attack data. It is used for both feature learning as well as classification of traffic. The classification is done using the RE threshold value. AE-D3F can achieve on both known and unknown attacks test sets, nearly 100% DR with less than 0.5% FPR, but there is a need to set the RE threshold value.

In the paper (Kasim 2020), the author has proposed the AE-SVM approach. Authors evaluated their proposed model on the following test scenarios: (1) The model trained over 16,902 data (2) Tested over randomly selected 15,000 data from CICIDS dataset (3) Tested over the 6957 dataset of DDoS attacks created with Kali Linux environment (4) Trained using NSL-KDD train dataset with ten-fold cross-validation. (5) Tested over NSLKDD. The AE-SVM method outperformed other methods in terms of low false-positive rate and rapid anomaly discovery. The accuracy of the proposed model over the NSL-KDD dataset is less compared to the other two datasets.

Bhardwaj et al. (2020) proposed an approach that combines a stacked sparse AE to learn features with a DNN for network traffic classification. First of all, Naive AE and DNN have been considered a baseline model in which authors have taken the random hyperparameters values for both AE and DNN. Then naive AE and DNN have been optimized for further improvements in AE and DNN model. The ten state-of-the-art approaches have been compared with the proposed approach. The approaches taken to compare over the NSL-KDD dataset are SAEC-SMR, AECGaussian NB, RNN, MLP, AECSVM, and SAVAERCDNN. The approaches taken to compare over the CICIDS2017 dataset are DT, ANN, SVM, SAVAER-CDNN, and LSTM. Results showed that the proposed approach outperformed the existing approaches over the NSL-KDD dataset with 98.43% accuracy and produced competitive results over the CICIDS2017 dataset by giving the accuracy of 98.92%. The proposed method is adequate to deal with feature learning and overfitting problem. The feature learning is achieved by training the AE with random samples of training data and the overfitting problem has been prevented by using the sparsity parameter. This article has not evaluated the recent dataset and has done offline analysis. Also, the detection time is not calculated for the proposed model.

Premkumar and Sundararajan (2020) proposed a DLDM frame structure to detect DoS attacks in WSN. The authors have used the DLDM framework, which uses RBF-based neural DL to classify the data. The authors took the simulation parameters, simulated the experiments in NS2, and presented the detection performance over a single CH. Authors showed that by taking a single CH, and the number of attackers taken from 5 to 15%, the detection ratio is between 86% to 99%, and the average false alarm rate is 15%. The DLDM showed a higher detection rate and a low false alarm rate than the MAS for the entire data forwarding phase. The nodes' lifetime is enhanced due to the reduction in the energy utilization of the nodes. The feasibility analysis of the proposed model has been done on simulator NS2 by calculating PDR, energy consumption, and throughput. The DLDM framework is valid for nodes with little mobility or without mobility, but in the WSN, nodes are highly dynamic and move frequently. Also, only generated dataset has been used for model evaluation.

(4) *Hybrid learning*

Roopak et al. (2019) have proposed four DL models, i.e. MLP, CNN, LSTM, and hybrid CNN-LSTM model, and compared with ML algorithms (SVM, Bayes, and RF ML algorithms). The authors evaluated them on the CICIDS2017 dataset, and this dataset is unbalanced. It is made balanced by duplicating the data. It has been observed that the hybrid CNN-LSTM model performed well compared to the rest of the DL and ML models. It gives an accuracy of 97.16%, and recall of 99.1%. The method by which the dataset has been made balanced is missing and offline analysis of the proposed model has been done for IoT networks.

Li and Lu (2019) proposed a model which is the combination of the LSTM and Bayes method, referred to as LSTM-BA. In this approach, LSTM first learns the DDoS attack mode using network traffic, which gives a probability of prediction for a DDoS attack. In this, the authors have determined the DDoS attacks with high prediction value (value greater than 0.5) and the normal traffic with a low prediction value (value less than 0.5) for DDoS attacks. Those prediction values from 0.2 to 0.8 authors re-detect it for high accuracy by using the Bayes method for identifying the DDoS. Authors have evaluated their LSTM-BA approach and LSTM module without Bayes over intrusion detection ISCX2012 dataset. From the results, it has been shown that the LSTM-BA performed well compared to LSTM in terms of F1-score. Then, the authors have compared their model with other existing methods, i.e. DeepDefense and Random Forest. LSTM-BA outperformed them with the highest F1-score and accuracy. In addition to the above experiments, authors have also verified the generalization of LSTM-BA. They examine the performance of LSTM-BA on data of the 5th day of the ISCX2012 dataset. Results showed that performance indicators have declined a little in the new data and the results are still good. Hence, it proves the generalization of the LSTM-BA approach. The LSTM-BA can take more time to detect the attack that is unsuitable for real-time scenarios. The proposed model increases the accuracy only by 0.16% compared to the existing DeepDefense method. The preprocessing time has not been calculated as the BOW, and feature hashing is used to convert IP addresses to a real vector.

Roopak et al. (2020) used the multi-objective optimization, i.e. the Non-dominated sorting algorithm (NSGA) method for feature selection on the preprocessed dataset. In this study, the combination of CNN and LSTM has been used to classify the attack. The CICIDS2017 dataset has been used for experimentations using GPU. The proposed method achieved a high accuracy of 99.03% and a F1-score value of 99.36%. Authors have also compared their method with MLP, SVM, RF, Bayes, and other state-of-the-art techniques. The results showed that the proposed model outperforms other work. The training time is reduced 11 times lower compared to other DL methods. In this article, most of the state-of-the-art techniques are not using the CICIDS2017 dataset. So the comparison seems not suitable.

Elsayed et al. (2020) proposed DDoSNet to detect DDoS attacks in SDNs. DDoSNet is a DL-based technique, which combines the RNN with AE. The model has been evaluated using the new dataset CICDDoS2019. Authors have also compared the DDoSNet with six classical ML techniques, i.e. DT, NB, RF, SVM, Booster, and LR. The evaluation of the DDoSNet model showed that it outperformed the existing six classical ML techniques in terms of accuracy, recall, precision, and F-score. The approach achieved 99% accuracy and AUC of 98.8 on CICDDoS2019 dataset. The offline analysis of the dataset has been done, and no multiclass classification has been done.

In Nugraha and Murthy (2020) a DL-based approach has been proposed to detect slow DDoS attacks in SDNs. This approach uses a hybrid CNN-LSTM. Firstly, authors have created synthetic datasets for slow DDoS attacks and benign flows because these attack traffic datasets are not available publicly. The synthetic traffic flow dataset having UDP and HTTP flows as benign traffic and HTTP flows as slow DDoS attack traffic are generated. Secondly, the proposed CNN-LSTM model is trained, validated, and tested over the generated datasets. The authors have compared the performance of the hybrid CNN-LSTM model with the DL model (MLP) and the ML technique (1-Class SVM). The proposed model outperformed other methods by achieving more than 99% in all performance metrics. The model is used only for the detection of slow DDoS attacks.

(5) *Other learning methods*

In the paper (He et al. 2020) He et al. have proposed a method based on deep transfer learning to detect small sample DDoS attack. Firstly, several neural networks are trained using DL techniques. The authors then compare the transfer performance of different networks using transferability metric. Then by comparing the transferability metric, the model with the best transfer performance has been selected out of the four networks. The authors then fine-tuned the parameters of the layers of the transferred network and trained it on the target domain. Authors showed a 20.8% improvement in detection of the 8LANN network in the target domain compared to the network where the parameters of all layers are initialized randomly, in which the final detection performance drops from 99.28 to 67%. Thus, the deep transfer network method combined with fine tuning technology improves the deterioration of detection performance caused by small sample of DDoS attacks. Only one attack is taken in the source domain and the target domain for model evaluation.

5 Available DDoS benchmarked datasets and classes of attacks in datasets

Table 4 lists the datasets and types of attack classes used by the papers that were reviewed for DDoS attack detection. It has been observed that most of the papers used seven datasets, namely, CICIDS2017 dataset, CICDDoS2019 dataset, ISCX2012 dataset, KDDCUP 1999 dataset, NSL-KDD dataset, CSECICIDS2018 dataset, and UNSWNB15 dataset. The description of these datasets is given as below.

KDDCUP 1999 The KDDCUP99 dataset (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html) is an intrusion detection standard dataset and was provided by the Massachusetts Institute of Technology laboratory (MIT). It is based on DARPA'98 data set. The total number of normal and attacks records are 1,033,372 and 4,176,086, respectively (Tavallaee et al. 2009). It contains total training and testing records of 4,898,431 and 311,027, respectively (Tavallaee et al. 2009). Each record has 41 features. It has three types of features, i.e. basic, traffic, and content (Tavallaee et al. 2009). This dataset contains emulated records. It is labelled and imbalanced dataset (Ring et al. 2019). This dataset has four types of attacks, i.e. Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R) and Probe attacks.The details are given as below (Gamage and Samarabandu 2020):

(1) Probe: ipsweep, nmap, satan, portsweep.
(2) DoS: back, land, smurf, neptune, pod, teardrop.
(3) U2R: buffer overflow, perl, loadmodule, rootkit.
(4) R2L: ftp write, guesspasswd, imap, multihop, phf, spy, warezmaster, warezlient.

Table 4 The recent DL-based DDoS attacks detection studies, their methods, datasets, and classes of attacks used

Taxonomy	References	Date of publication	Approach used	Dataset used	Classes of attacks in the studies
Supervised Instance Learning	Hasan et al. (2018)	2018	Deep CNN	Optical Burst Switching (OBS) Network dataset	–
	Amma and Subramanian (2019)	2019	CNN	NSL KDD	DoS
	Chen et al. (2019)	2019	Multichannel CNN	KDDCUP99 and CICIDS2017	KDDCUP99: Normal, DoS, R2L, U2R, Probe. CICIDS2017: DoS/DDoS: Hulk, Heartbleed, slowloris, Slowhttptest, GoldenEye
	Shaaban et al. (2019)	2019	CNN model	1. Captured from simulated MCC network by Wireshark 2. NSL-KDD	Dataset 1: TCP and HTTP Flood DDoS Attack. NSL-KDD: DoS, Probe, R2L, U2R
	Sabeel et al. (2019)	2019	DNN, LSTM	CICIDS2017 and ANTS2019	CICIDS2017: Benign, DoS GoldenEye, DoS Slowloris, DoS Hulk, DoS Slowhttptest, DDoS. ANTS2019: DDoS attack and Benign
	Virupakshar et al. (2020)	2020	DT, KNN, NB, and DNN	KDDCUP, LAN, and Cloud	KDDCUP99: Normal, DoS, R2L, U2R, Probe. Cloud: ICMP flooding, TCP flooding, and HTTP flooding
	Haider et al. (2020)	2020	Ensemble RNN, LSTM, CNN, and Hybrid RL	CICIDS2017	Slowloris, Slowhttptest, Hulk, GoldenEye, Heartbleed, and DDoS
	Wang and Liu (2020)	2020	Information entropy and CNN	CICIDS2017	Benign, BForce, SFTP and SSH, slowloris, Slowhttptest, Heartbleed, Web BForce, Hulk, GoldenEye, XSS and SQL Inject, Infiltration Dropbox Download, Botnet ARES, Cool disk, DDoS LOIT, PortScans
	Kim et al. (2020)	2020	CNN	KDDCUP99 and CSE-CIC-IDS 2018	KDDCUP99: Benign, Neptune and Smurf Attack. CSE-CIC-IDS 2018: Benign, DoS-SlowHTTPTest, DoS-Hulk Attack, DoS-GoldenEye, DoS-Slowloris, DDoS-HOIC, DDoS-LOIC-HTTP
	Doriguzzi-Corin et al. (2020)	2020	CNN	ISCX2012, CIC2017, and CSECIC2018	ISCX2012: DDoS attack based on an IRC botnet. CIC2017: HTTP DDoS generated with LOIC. CSECIC2018: HTTP DDoS generated with HOIC
	Asad et al. (2020)	2020	DNN	CICIDS2017	Benign, DoS Slowloris, DoS Hulk, DoS SlowHTTPTest and DoS GoldenEye
	Muraleedharan and Janet (2020)	2020	DNN	CICIDS2017	Benign, Slowloris, SlowHTTP, Hulk, GoldenEye

Table 4 continued

Taxonomy	References	Date of publication	Approach used	Dataset used	Classes of attacks in the studies
	Sbai and El Boukhari (2020)	2020	DNN	CICDDoS2019	Data flooding or UDP flooding attack
	de Assis et al. (2020)	2020	CNN	Simulated SDN data and CICDDoS 2019	SDN dataset: DDoS attack. CICDDoS2019: Twelve DDoS attacks on the training day and seven attacks during the testing day
	Hussain et al. (2020)	2020	CNN model i.e., ResNet	CICDDoS2019	Syn, TFTP, DNS, LDAP, UDP Lag, MSSQL, NetBIOS, SNMP, SSDP, NTP, UDP, and Normal traffic
	Amaizu et al. (2021)	2021	DNN	CICDDoS2019	UDP LAG, SYN, DNS, MSSQL, NTP, SSDP, TFTP, NetBIOS, LDAP, UDP and Benign
	Cil et al. (2021)	2021	DNN	CICDDoS2019	Twelve DDoS attacks on the training day and seven attacks during the testing day
Supervised Sequence Learning	Li et al. (2018)	2018	LSTM, CNN/LSTM, GRU, 3LSTM	ISCX2012 dataset and Generated DDoS attacks	Generated DDoS attacks : ARP flood inundation attack, Smurf attack, SYN flood inundation attack, Ping of Death attack, and UDP flood inundation attack. ISCX2012: HTTP Denial of Service and Distributed Denial of Service using an IRC Botnet
	Priyadarshini and Barik (2019)	2019	LSTM	CTU-13 Botnet, ISCX 2012 and, some real DDoS attacks	ISCX2012: Infiltrating the network from the inside, DDoS using an IRC botnet, HTTP DoS, SSH brute force. CTU-13: IRC, Port Scan, FastFlux, spam, ClickFraud, US. Some real DDoS attacks are: TCP, UDP and ICMP
	Liang and Znati (2019)	2019	LSTM	CICIDS2017	Slowloris, Hulk, Slowhttptest, GoldenEye and LOIC
	Shurman et al. (2020)	2020	Hybrid IDS and LSTM	Reflection-based CICDDoS2019	MSSQL, SSDP, CharGen, LDAP, NTP, TFTP, DNS, SNMP, NETBIOS, and PORTMAP
	Assis et al. (2021)	2021	GRU	CICDDoS2019 and CICIDS2018	CICDDoS2019: Twelve DDoS attacks on the training day and seven attacks during the testing day. CICIDS2018: Infiltration of the network from inside, HTTP denial of service, Collection of web application attacks, Brute force attacks, Last updated attacks
Semi-supervised instance learning	Catak and Mustacoglu (2019)	2019	AE and a deep ANN	UNSWNB15 and KDDCUP99	UNSWNB15 dataset: Normal, Analysis, Fuzzers, Backdoors, Exploits, DoS, Reconnaissance, Shellcode and Worm. KDDCUP99: neptune, Smurf, Teardrop
	Ali and Li (2019)	2019	Deep AE and MKL	ISCXIDS2012 and UNSWNB15	ISCXIDS2012: Normal Activity. UNSWNB15: Fuzzers, Backdoors, Analysis, DoS, Exploits, Generic, Shellcode, Reconnaissance and Worms

Table 4 continued

Taxonomy	References	Date of publication	Approach used	Dataset used	Classes of attacks in the studies
	Yang et al. (2020)	2020	AE	Synthetic Dataset, UNB2017 and MAWI	Synthetic dataset: Excessive get post-attack, Recursive get attack, SlowLoris attack, and Slow post-attack. UNB2017: Slow HTTP attack, Hulk attack, Slowloris attack, and Golden eye. MAWI: Normal samples
	Kasim (2020)	2020	AE-SVM	CICIDS2017, NSL-KDD and 6957 data set of DDoS attacks	CIC-IDS2017: Slowloris, Slowhttptest, Hulk, GoldenEye, DDoS LOIT. NSL-KDD : Back, Land, Pod, Smurf, Neptune, Teardrop, Processtable, Udpstorm, Apache2, Mailbomb, Worm. 6957 data set of DDoS attacks
	Bhardwaj et al. (2020)	2020	AE with DNN	NSL-KDD and CICIDS2017	NSL-KDD: Back, Land, Teardrop, Mailbomb, Processtable, Udpstorm, Neptune, Pod, Smurf, Apache2, and Worm. CICIDS2017: Slowloris, Hulk, Slowhttptest, GoldenEye, DDoS LOIT
	Premkumar and Sundararajan (2020)	2020	RBF	Generated dataset	Data Flooding, Jamming, Exhaustion, Sinkhole, Eavesdropping and Packet dropping attack
Hybrid Learning	Roopak et al. (2019)	2019	MLP, CNN, LSTM, and hybrid CNN + LSTM	CICIDS2017	Slowloris, Slowhttptest, Hulk, GoldenEye, DDoS LOIT
	Li and Lu (2019)	2019	LSTM and Bayes	ISCX2012	HTTP Denial of Service and Normal Activity
	Roopak et al. (2020)	2020	CNN with LSTM	CICIDS2017	DDoS
	Elsayed et al. (2020)	2020	RNN-AE	CICDDoS2019	Twelve DDoS attacks on the training day and seven attacks during the testing day
	Nugraha and Murthy (2020)	2020	CNN-LSTM	Synthetically generated	Slow DDoS attack: HTTP flows. Benign traffic: UDP and HTTP flows
Transfer learning	He et al. (2020)	2020	6LANN, 7LANN, 8LANN, 9LANN	–	SYN-type, and LDAP-type DDoS attacks

NSL-KDD dataset (https://www.unb.ca/cic/datasets/nsl.html; Protić 2018) This dataset is an extension of the KDDCUP99 dataset to eliminate some problems of KDDCUP99 dataset. KDDCUP99 dataset contains many redundant and duplicate records, and to fix these problems, the NSL-KDD dataset was proposed. The number of records in the train and test sets is reasonable in the NSL-KDD dataset. It contains approximately 150,000 data points, and this dataset also contains emulated records (Ring et al. 2019). The dataset is labelled and imbalanced (Ring et al. 2019) and contains training records of 125,973 and testing records of 22,544 (Gamage and Samarabandu 2020). It also includes four types of attacks (Protić 2018):

(1) DoS: Back, Land, Pod, Smurf, Apache2, Neptune, Teardrop, Mailbomb, Processtable, Udp storm, Worm.
(2) Probe: IPsweep, Satan, Nmap, Mscan, Portsweep, Saint.
(3) R2L: Ftp write, Imap, Guess password, Phf, Multihop, Warezmaster, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Named, Sendmail.
(4) U2R: Buffer overflow, Perl, Loadmodule, Rootkit, Sqlattack, Ps, Xterm.

UNSWNB15 dataset (Moustafa and Slay 2015) It was generated in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). Four tools were used to create this dataset, i.e. IXIA PerfectStorm, argus, bro-IDS, and tcpdump tools. The IXIA PerfectStorm tool is utilised to generate a hybrid of the normal and abnormal network traffic. The IXIA tool generates nine types of attacks that are fuzzers, reconnaissance attacks, exploits, backdoors, generic attacks, shellcode, DoS attacks, worms, and analysis attacks (Gümüşbaş et al. 2020). The tcpdump tool captured the network traffic in the form of packets. The simulation period of the dataset was a total of 31 h for capturing 100 GBs, i.e. 16 h on 22-01-2015 and 15 h on 17-02-2015. Argus and bro-IDS tools extracted the reliable features from the pcap files. It has 49 features. In addition to it, twelve algorithms using a C# language were also developed to analyse the flow of the connection packets. It contains two million and 540,044 number of records having 2,218,761 benign records and 321,283 malicious records.

ISCX2012 (https://www.unb.ca/cic/datasets/ids.html) The ISCX2012 dataset was created in 2012 by Ali Shiravi et al. (Shiravi et al. 2012), consisting of the 7 days from 11-06-2010 to 17-06-2010) of network activity having normal and malicious traffic and includes full-packet network data. The malicious traffic includes Infiltrating the network from inside, Distributed Denial of Service, HTTP Denial of Service, and Brute Force SSH. This dataset was created in an emulated network environment. It has imbalanced and labelled dataset (Ring et al. 2019). In the ISCX dataset two general profiles are used, i.e. α profiles, which characterize attack behaviour and β profiles, which characterize normal user scenarios (Ring et al. 2019). It has a total of 2,381,532 benign and 68,792 malicious records (Ahmad and Alsmadi 2021).

CICIDS2017 (https://www.unb.ca/cic/datasets/ids-2017.html)
The CICDS2017 dataset was generated in an emulated environment from 03-07-2017 to 07-07-2017 (Ring et al. 2019). This dataset comprises packet-based and bidirectional flow-based format of network traffic. The CICIDS2017 dataset is created by Sharafaldin et al. It implements normal activity and attacks like DoS, Heartbleed, Brute Force SSH, Web Attack, Botnet, Infiltration, and DDoS, and Brute Force FTP (Gümüşbaş et al. 2020; Panigrahi et al. 2018). More than 80 features have been extracted for each flow by the CICFlowMeter tool from the generated network traffic. The dataset made the abstract behaviour of 25 users based on some protocols like FTP, SSH, HTTP, HTTPS, and email protocols. It has 2,273,097 benign records and 557,646 malicious records (Ahmad and Alsmadi 2021).

CSE-CIC-IDS2018 dataset (https://www.unb.ca/cic/datasets/ids-2018.html) It has been created by the Communications Security Establishment (CSE) & the Canadian Institute for Cybersecurity (CIC) collected over 10 days, from Wednesday (14-02-2018) to Friday (02-03-2018). This dataset has been generated on the large network and includes seven types of attack scenarios: Heartbleed, Botnet, Brute-force, DoS, Web attacks, DDoS, and infiltration of the network from inside. The CICFlowMeter tool has extracted 80 features from the created network traffic.

CICDDoS2019 (https://www.unb.ca/cic/datasets/ddos-2019.html) The CICDDoS2019 dataset was generated by Sharafaldin et al. (2019). The features were extracted from the raw data, by using the CICFlowMeter-V3 tool and extracted more than 80 traffic features. The CICDDoS2019 comprises benign and up-to-date common DDoS attacks. This dataset was generated using real traffic and comprises a large amount of different DDoS attacks generated through protocols using TCP/UDP. The taxonomy of attacks include exploitation-based and reflection-based attacks. The reflection-based attacks contain Microsoft SQL Server (MSSQL), Network Time Protocol (NTP), Simple Service Discovery Protocol (SSDP), CharGen, Trivial File Transfer Protocol (TFTP), Lightweight Directory Access Protocol (LDAP), Domain Name Server (DNS), Simple Network Management Protocol (SNMP), Network Basic Input/Output System (NETBIOS), and PortMap. The Exploitation-based attacks include UDP flood, UDPLag and SYN flood. This dataset was gathered over 2 days in both PCAP file and flow format based for training and testing evaluation. On the training day, twelve types of DDoS attacks included DNS, LDAP, NTP, MSSQL, UDP, UDP-Lag, NetBIOS, SNMP, SSDP, WebDDoS, TFTP, and SYN which were captured on January 12th, 2019 and seven attacks on the testing day include NetBIOS, PortScan, LDAP, UDP, UDP-Lag, MSSQL and SYN, which were captured on March 11th, 2019.

6 Preprocessing strategies, hyperparameter values, experimental setups, and performance metrics

Table 5 shows the preprocessing strategies, hyperparameter values, experimental setups, and performance metrics that the existing DL approaches have used for DDoS attack detection.

Preprocessing strategies The preprocessing of the data is done before training and testing the model (Holzinger 2019). The preprocessing of data is vital because it extracts valuable information from raw data and converts that information into a suitable format that rises the learning capability of the model (Deshmukh et al. 2015; Kim 2019). In this paper, a summary of preprocessing strategies used in the existing literature is given in Table 5.

Hyperparameter values Wu et al. (2019): Hyperparameters are important as they directly control the behaviours of training ML algorithms. The selection of particular hyperparameter values is done before training the model and requires expert knowledge and experience. The process of finding the hyperparameter values which gives the best performance on the data for ML algorithms is called the hyperparame-

Table 5 The recent DL-based DDoS attack detection studies with their preprocessing strategies, hyperparameter values, experimental setups, and performance metrics

Taxonomy	References	Preprocessing strategies	Hyperparameter values	Experimental setups	Performance metrics
Supervised instance learning	Hasan et al. (2018)	–	The architecture consist of two convolutional layer followed by maxpooling layer, ReLu function, fully connected layer (250 neurons), ReLu function, dropout layer FC layer (four neurons). The stride size used is 1 X 1 for CL and 1 X 2 used for PL. Used back-propagation and SoftMax loss function	–	Accuracy = 99%, Sensitivity = 99%, Specificity = 99%, Precision = 99%, F1-score = 99%, False positive rate = 1%, and False negative rate = 1%
	Amma and Subramanian (2019)	Min–max normalization	The pre-training module has two stages of training and each stage comprises of CL and PL. The filter size is 3 and the max pooling has size 2. Training Module: The FCNN comprises of input layer, two hidden layers, the output layer with 11-9-7-6 number of nodes, respectively, and the activation function is ReLu	–	Accuracy: Normal = 99.3%, Back = 97.8%, Neptune = 99.1%, Smurf = 99.2%, Teardrop = 83.3%, Others = 87.1%. Precision: Normal = 99.6%, Back = 95.9%, Neptune = 97.9%, Smurf = 91.1%, Teardrop = 19.6%, Others = 97.8%. Recall: Normal = 99.3%, Back = 97.8%, Neptune = 99.1%, Smurf = 92.2%, Teardrop = 83.3%, Others = 87.1%. F1-score: Normal = 99.4%, Back = 96.8%, Neptune = 98.5%, Smurf = 95.0%, Teardrop = 31.7%, Others = 92.1%. False Alarm: Normal = 0.7, Back = 2.2, Neptune = 0.9, Smurf = 0.8, Teardrop = 16.7, Others = 12.9. AUC: Normal = 0.993, Back = 0.978, Neptune = 0.991, Smurf = 0.992, Teardrop = 0.833, Others = 0.871

Table 5 continued

Taxonomy	References	Preprocessing strategies	Hyperparameter values	Experimental setups	Performance metrics
	Chen et al. (2019)	–	Used incremental training method to train MC-CNN	–	Accuracy: KDDCUP99 (2 class) = 99.18%, KDDCUP99 (5 class) = 98.54%, CICIDS2017 = 98.87%
	Shaaban et al. (2019)	Features are converted into matrix form i.e., 8 and 41 features into 3*3 and 6*7 matrix using padding	The CNN model contains three stages and the first stage comprised of the input layer, two CLs, and the output from these layers is fed to PL. Second stage has two CLs and a PL. The third stage consists of a FC network and output layer. ReLu function has been used in all layers except the output layer that uses softmax function	Keras library and Tenser-flow library	Dataset1: Accuracy = 0.9933, Loss = 0.0067. Dataset2 (NSL-KDD): Accuracy = 0.9924, Loss = 0.0076
	Sabeel et al. (2019)	Z-score normalization	The input layer of the DNN/LSTM model has size of 25 followed by a dense/recurrent layer having 60 neurons and dropout of 0.2, FC dense layer having 60 neurons, a dropout rate is 0.2, another dense layer having 60 neurons. All layers have ReLu activation function. Then, a dense FC output layer used the sigmoid activation function. The learning rate is 0.0001 and batch size is set to 0.0001	GPU NIVDIA Quadro K2200, Intel (R) Xenon (R) with CPU E5-2630 v3@ 2.40GHz, 240GB SSD, 64-bit Windows 10 Pro 1809. Software: Python 3.7.3, TensorFlow1.13.0, keras 1.1.0, and NVIDIA Cuda Toolkit 10.0.130 with cuDNN 7.6.0	Accuracy = 98.72%, TPR = 0.998, Precision = 0.949, F1-score = 0.974, AUC = 0.987
	Virupakshar et al. (2020)	–	–	Controller: 1 GB RAM, 50 GB Memory, 2 (No. of core) Processor. Neutron: 1 GB RAM, 20 GB Memory, 2 (No. of core) Processor. Computer-1: 1 GB RAM, 20 GB Memory, 2 (No. of core) Processor. Computer-2: 1 GB RAM, 20 GB Memory, 2 (No. of core) Processor	KDDCUP99: Recall = 0.99, F1-score = 0.98, Support = 2190. LAN Dataset: Recall = 0.91, F1-score = 0.91, Support = 2140. Cloud Dataset: Recall = 0.91, F1-Score = 0.91, Support = 2138, Precision = 96%

Table 5 continued

Taxonomy	References	Preprocessing strategies	Hyperparameter values	Experimental setups	Performance metrics
	Haider et al. (2020)	Z-score normalization	The ensemble CNN model (M1 and M2) contains three 2-d CLs (with 128, 64, & 32 filters, respectively), 2 max PLs, 1 layer to flatten, and 2 dense FC layers. ReLu as activation function in hidden layers and Sigmoid at the output layer	System Manufacturers : Lenovo, Processor: Intel Core i7-6700 CPU with 3.4 GHz Processor, Memory 8GB, Operating System: Microsoft Windows 10. Software: Keras Library with TensorFlow	Accuracy = 99.45%, Precision = 99.57%, Recall = 99.64%, F1-score = 99.61%, Testing time = 0.061 (minutes), Training time = 39.52 (minutes), CPU Usage% = 6.025
	Wang and Liu (2020)	Each byte of a packet is converted into a pixel and gathered as a picture	The model includes two CLs, two PLs, and two FC layers. For Information entropy the threshold is selected as 100 Packets/s	Mininet emulator, POX controller and a PC with Inter Core i5- 7300HQ CPU, 8GB RAM, and Ubuntu 5.4.0-6 system. The experimental topology comprises of six switches, one server and a controller. Software: Hping3, TensorFlow framework	Accuracy =98.98%, Precision =98.99%, Recall =98.96%, F1-score =98.97%, and Training time =72.81s, AUC = 0.949
	Kim et al. (2020)	One-hot encoding and 117 features are converted into images with 13 9 pixels and 78 features to 13 6	The model comprises of 1, 2, or 3 CLs, and the number of kernels is set to 32, 64 and 128, respectively. In addition, the kernel size is set to 2 2, 3 3, and 4 4. The stride value is set to 1	Python with TensorFlow	Accuracy: KDDCUP99 = 99%. CSE-CIC-IDS2018 = 91.5%
	Doriguzzi-Corin et al. (2020)	Min–max normalization and converts the traffic flows into array-like data structures and splits them into sub-flows based on time windows	Used $n = 100$, $t = 100$, $k = 64$, $h = 3$, $m = 98$ (n is the maximum number of packets, the time window of length t seconds, a single CL with k filters of size h f, h is the length of each filter, and f is the no. of features, m is pool size). The model has batch size $s = 2048$ with the Adam optimizer and learning rate = 0.01. The output layer uses the sigmoid activation function	Used a server-class computer equipped with two 16-core Intel Xeon Silver 4110 @2.1 GHz CPUs and 64 GB of RAM. Software: Python v3.6 using the Keras API v2.2.4 on top of TensorFlow 1.13.1	ISCX2012: Accuracy = 0.9888, FPR = 0.0179, Precision = 0.9827, Recall = 0.9952, F1-score = 0.9889. CICIDS2017: Accuracy = 0.9967, FPR = 0.0059, Precision = 0.9939, Recall = 0.9994, F1-score = 0.9966. CSECIC2018: Accuracy = 0.9987, FPR = 0.0016, Precision = 0.9984, Recall = 0.9989, F1-score = 0.9987. UNB201X: Accuracy = 0.9946, FPR = 0.0087, Precision = 0.9914, Recall = 0.9979, F1-score = 0.9946

Table 5 continued

Taxonomy	References	Preprocessing strategies	Hyperparameter values	Experimental setups	Performance metrics
	Asad et al. (2020)	Min–max scaling and Cost sensitive learning technique	The model with the input (66 neurons), output layer (5 neurons) and the seven hidden layers (with 128, 256, 128, 64, 32, 16, 8 number of neurons, respectively). The batch normalization with the batch size of 1024, ReLu function with dropout rate 0.2 used in each hidden layer. The learning rate is 0.001 and number of epochs are 300	CPU Platform: 2.5 GHz Intel Xeon E5 v2, GPU: NVIDIA Tesla K80, CPU Cores: 4, GPU Memory: 24 GB of GDDR5, RAM: 26 GB	Accuracy = 98%, F1-score = 0.99 and AUC ≈ 1
	Muraleedharan and Janet (2020)	-	The input and output layer has 80 and 5 number of neurons. The output layer and four hidden layers used Softmax and ReLu activation function, respectively. Also used the adam optimization algorithm and categorical cross-entropy function	Keras API and SciKit	Accuracy = 99.61%, Precision: Benign = 0.99, Slowloris = 1.00, Slowhttptest = 0.99, Hulk = 1.00, GoldenEye = 1.00. Recall: Benign = 1.00, Slowloris = 0.99, Slowhttptest = 0.98, Hulk = 1.00, GoldenEye = 1.00. F1-score: Benign = 1.00, Slowloris = 0.99, Slowhttptest = 0.99, Hulk = 1.00, GoldenEye = 1.00
	Sbai and El Boukhari (2020)	–	–	–	Precision = 0.99, Recall = 1, F1-score = 0.99, Accuracy = 0.99997
	de Assis et al. (2020)	The qualitative dimensions are converted to quantitative using Shannon Entropy	The CNN model is composed of a stack of two Conv1D (with 16 and 8 filters) and MaxPooling1D (with a pool size of 2) layers followed by a Flatten layer, a Dropout layer (Dropout rate of 0.5), and a FC layer (with 10 neurons). The output has a neuron with sigmoid activation function. The model used 1000 epochs	A computer with Windows 10 64 bit, Intel Core i7 2.8GHz, and 8GB of RAM. Software: Python and Keras	Simulated SDN data: Accuracy = 99.9% (On average), Precision = 99.9% (On average), Recall = 99.9% (On average) and F-measure = 99.9% (On average). CICDDoS 2019: Accuracy = 95.4%, Precision =93.3%, Recall =92.4% and F-measure =92.8%

Table 5 continued

Taxonomy	References	Preprocessing strategies	Hyperparameter values	Experimental setups	Performance metrics
	Hussain et al. (2020)	Used min–max normalization and converted the samples matrices to images using the OpenCV library. The images of dimension 60 x 60 x 3 is converted into 224 x 224 x 3	The ResNet18 model consists of 10 CLs and 8 PLs. The outputs set for binary and multiclass classifications are as 1 and 12, respectively. The leaning rate = 0.0001, momentum = 0.9, epochs for binary classification = 10 , epochs for multi-class classification = 50 and SGD optimizer have been used	–	Multiclass: Precision = 87%, Recall = 86%, Accuracy = 87.06% and F1-measure = 86. Binary: Accuracy = 99.99%
	Amaizu et al. (2021)	Min–max scaling function	No. of hidden layers = 7, Activation function = ReLu, Dropout Layers = 2, Learning rate = 0.001, Loss function = SCC and epochs = 50	One server, one firewall, two switches, and four PC. Software: Keras Sequential API and the Keras Functional API, Keras-tuner Library	Recall = 99.30%, Precision = 99.52%, F1-score = 99.99%, Accuracy = 99.66%
	Cil et al. (2021)	Min–max normalization	The DNN model comprises of three hidden layers having 50 units of neurons and sigmoid activation function. The output layer has two neurons with softmax activation function	The computer with Windows 10 OS, Intel Core i7-7700, CPU 4.2 GHz processor, 32 GB RAM, 2X512GB SSD and NVIDIA GTX 1080 Ti Graphics Coprocessor. Software: Python 3.7 and deep learning libraries	Dataset1: Accuracy = 0.9997, Precision = 0.9999, Recall = 0.9998, F1-Score = 0.9998. Dataset2: Accuracy = 0.9457, Precision = 0.8049, Recall = 0.9515, F1-Score = 0.8721
Supervised sequence learning	Li et al. (2018)	BOW and the 2-D feature matrix is transformed to a 3-D matrix	A DL model comprises of an input, forward recursive, reverse recursive, FC hidden, and output layers	Two NVIDIA K80 GPU and 128 GB memory. Software: Ubuntu 14.04, Keras, and Spirent contracting tools	Accuracy = 98%

Table 5 continued

Taxonomy	References	Preprocessing strategies	Hyperparameter values	Experimental setups	Performance metrics
	Priyadarshini and Barik (2019)	One-hot encoding	LSTM uses two hidden layers with 128 neurons, and Sigmoid function, the output layer uses a tanh function and the loss function = binary cross-entropy, Adams' optimizer with a dropout rate = 0.2, the mini-batch size = 512 iterations	PHP, MySQL as pre-requisite, Cent OS7, MariaDB, Apache server, Linux, windows OS, HPing-3, Mininet emulator, FloodLight controller, Python library Keras and TensorFlow	Accuracy = 98.88%
	Liang and Znati (2019)	For each network flow, F, a subsequence of n packets, $S \subset F$, is inspected. If there is not enough packets in a flow then it is padded with fake packets	The model has two LSTM layers, a dropout layer, and a FC layer. A sequence of 10 packets from each flow has been taken	–	CICIDS2017 (Wednesday): Precision = 0.9995, Recall = 0.9997, F1-score = 0.9991. CICIDS2017 (Friday): Precision = 0.9998, Recall = 1, F1-score = 0.9999
	Shurman et al. (2020)	One hot encoder and RF for feature selection	The model comprises of three LSTM layers with 128 neurons and sigmoid function, three dropout layers, and a dense layer with tanh function. The model used categorical cross-entropy loss function and RMS propagation as an optimizer	–	Accuracy = 99.19%
	Assis et al. (2021)	Used MD5 hashing process to convert qualitative dimensions into quantitative values	GRU layer (C = 32), followed by a dropout layer having dropout rate 0.5, and a FC layer with ten neurons. The output layer has a neuron with sigmoid function	A computer using Windows 10 64 bit, Intel Core i7 2.8 GHz, and 8 GB of RAM. Software: Python, Keras and Sklearn	CICDDoS2019: Average metrics (Accuracy, Precision, Recall, and F-measure) = 99.94%, legitimate flow classification rate = 99.6%. CICIDS2018 dataset: Accuracy = 97.1%, Precision = 99.4%, Recall = 94.7%, and F-measure = 97%, legitimate flow classification rate = 99.7%. Mitigation Evaluation Outcomes: i. the absolute number of normal flows dropped, ii) the absolute number of attack flows not dropped. CICDDoS2019: i. 188 ii. 48. CICIDS2018 dataset: i. 2660 ii. 48636

Table 5 continued

Taxonomy	References	Preprocessing strategies	Hyperparameter values	Experimental setups	Performance metrics
Semi-supervised instance learning	Catak and Mustacoglu (2019)	Normalization model	AE model consisted of the input layer, three hidden layers, and output layer with unit numbers 28, 19, 9, 19 and 28 with sigmoid as an activation function. DNN comprises of the input layer, five hidden layers with unit numbers 28, 500, 800, 1000, 800, and 500, respectively. Mini-batch SGD optimization, and binary cross-entropy loss function have been used	A CPU and a NVIDIA Quadro 1000M GPU. Software: Python 3.5 with 64 bits, Keras, TensorFlow and SciKit-learn libraries of Python, Windows 7 with 64 bits	UNSWNB15: F1-score = 0.8985, Accuracy = 0.9744, Precision = 0.8924, Recall = 0.9053. KDDCUP99: Overall Accuracy and Precision≈ 99%
	Ali and Li (2019)	Features that are not numbers are discretized	The number of MSDA used are 9 in the experiments with the number of layers selected as $L = [1, 3, 5, 7, 9, 11]$	A computer with 32.5 GB memory and NVIDIA Tesla V100 GPUs Software: MATLAB	Average Accuracy for Dataset D1 to D16 = 93% and on D2 = 97%
	Yang et al. (2020)	The flow is divided into many subflows according to threshold value of 10 ms	AE model has one input layer, three hidden layers, and one output layer. The neurons number in each layer is 27-24-16-24-27, respectively. The leaky ReLu activation function, Adam optimizer, MSE are used and Batch size is set to 32	–	Exp1: SYNT: DR (Detection Rate) = 98.32%, FPR = 0.38%. UNB2017: DR = 94.10%, FPR = 1.88%. Exp2: SYNT: DR = 100%, FPR = 100%. UNB2017: DR = 94.14%, FPR = 1.91%. Exp2: Testset1: DR = 100%, FPR = 0.49%. Testset2: DR = 99.99%, FPR = 0.49%
	Kasim (2020)	Label encoding and Min–max normalization	AE parameters: Input neurons: 82, Output neurons: 82, Hidden neurons: 25, Learning rate: 0.3, Momentum: 0.2, SVM Parameters: It has 25 input and 2 output nodes. The learning rate is 0.01 and the number of iterations are 1000	A computer with Intel (R) Core (TM) i7–2760 QM CPU with a frequency of 2.40 GHZ and 8 GB of RAM. Software: Rest API and Python with keras, scapy, TensorFlow and SciKit libraries	1. Training Time = 2.03s, Testing Time = 21 ms, Accuracy on CICIDS2017 = 99.90%. 2. Created DDoS attacks: Accuracy = 99.1%. AUC = 0.9988. 3. NSL-KDD test: Accuracy = 96.36%

Table 5 continued

Taxonomy	References	Preprocessing strategies	Hyperparameter values	Experimental setups	Performance metrics
	Bhardwaj et al. (2020)	One hot encoding and min–max normalization	AE model: Two encoding layers = 70 and 50 neurons, coding layer = 25 neurons, and two decoding layers = 25 neurons and all layers with ReLu activation. The output layer has sigmoid activation, and the optimizer used is Adadelta. DNN: It has 20, 12 neurons in two hidden layers, and the optimizer is Adabound	PC with Windows 10-64 bits and 16 GB RAM and CPU Intel(R) Core-i7, and VMware workstation	NSL-KDD: Accuracy = 98.43%, Precision = 99.22%, Recall = 97.12%, F1-score = 98.57% CICIDS2017: Accuracy = 98.92%, Precision = 97.45%, Recall: 98.97%, F1-score = 98.35%
	Premkumar and Sundararajan (2020)	–	–	No. of nodes: 200, Simulation time: 500s, the number of attacking nodes = 5 to 20% of the normal nodes and Constant Bit Rate (CBR) application is used	Attackers between 5 and 15%, detection ratio is 86–99%, the false alarm rate = 15%
Hybrid Learning	Roopak et al. (2019)	–	CNN + LSTM model has 1d CNN layer with ReLu function, followed by a LSTM layer with adam activation function, a dropout layer having rate of 0.5, a FC layer and a dense layer with a sigmoid function	A PC with 64-bit Intel Core-i7 CPU with 16 GB RAM in Windows 7. Software: Keras on TensorFlow package for DL and MATLAB 2017a for ML algorithm	Accuracy =97.16%, Recall =99.1% and Precision =97.41%
	Li and Lu (2019)	BOW and feature hashing. The 2-d matrix converted into a three-dimensional matrix	LSTM module consists of two hidden, a FC layers of 256 neurons with ReLU activation function, and a FC layer of 1 neuron with Sigmoid activation function	NVIDIA GTX 1050 GPU	Accuracy = 98.15%, Precision = 98.42%, Recall = 97.6%, TNR = 98.4%, FPR = 1.6%, F1-Score =98.05%

Table 5 continued

Taxonomy	References	Preprocessing strategies	Hyperparameter values	Experimental setups	Performance metrics
	Roopak et al. (2020)	Min–max normalization	The 1d-CNN is followed by maxpooling, LSTM, and dropout layers with Relu activation function. The output layer classifies using a sigmoid function with binary cross-entropy. The dropout rate is 0.2, learning rate is 0.001, batch size is 256 and epochs are 100.	GPU NVIDIA Tesla VIOO GPUs, having 16 GB VRAM with 256 GB on 10 number of nodes in HPC. Software: Keras on TensorFlow	Precision = 99.26%, Recall = 99.35%, Accuracy = 99.03%, F-measure = 99.36%, Training time = 15313.10 s
	Elsayed et al. (2020)	Min–max normalization	The RNN-AE consists of four RNN hidden layers. The encoder phase has the number of channels equal to 64, 32, 16, and 8 and the decoder phase has in the reverse order of it. The last layer has two channels with softmax function. Other parameters are categorical cross-entropy as loss function with adam optimizer, ReLu function in all layers, with 50 number of epochs and a batch size is 32, learning rate is 0.0001	–	Precision: Attack = 0.99, benign = 1.00. Recall: Attack = 0.99, benign = 0.99. F1-score: Attack = 0.99, benign = 0.99. Accuracy: 99%. AUC =98.8
	Nugraha and Murthy (2020)	Min–Max Scaler	Three layers between the CNN and the LSTM layer i.e. dropout, maxpool, and flatten layers. Then LSTM layer is followed by a FC dense layer having ReLu function, dropout layer, and the last dense layer having sigmoid function. The learning rate = 0.0005, dropout rate = 0.3, and CNN filter = 64 and kernel size = 5 and the number of epochs is set to 50	Python	Accuracy = 99.998%, Precision = 99.989%, Specificity = 99.997%, Recall = 100%, F1 score = 99.994%

Table 5 continued

Taxonomy	References	Preprocessing strategies	Hyperparameter values	Experimental setups	Performance metrics
Transfer learning	He et al. (2020)	–	8LANN consists of eight FC layer. Each layer except the eighth layer are followed by batch normalization and ReLu function. The batch size is 500, cross-entropy loss function, SGD optimizer and the learning rate = 0.001 have been used	Ubuntu 16.04 64 bit OS with 64 GB of the memory. The GPU accelerator is NVIDIA RTX 2080Ti. Software: PyTorch	Detection performance = 87.8%, Transferability value = 19.65

ter tuning. The hyperparameter tuning can be done in two ways, like manual search and automatic search methods. In the manual search hyperparameter values are selected by hand. The automatic search method is like Grid search. But the grid search method is expensive. Therefore, to solve the problem of grid search, another method, i.e. Random search, has come into the picture. Hyperparameters include the number of epochs, batch size, learning rate, activation functions, number of layers, number of neurons in each layer, etc. (https://towardsdatascience.com/understanding-hyperparameters-and-its-optimisation-techniques-f0debba07568; https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a).

Experimental setup It involves the hardware configuration, software, dataset used, etc., and describes the procedure of experiments conducted. The hardware configuration is important because the training and testing times depend upon it. As the DL algorithms are complex so they require good hardware configurations.

Performance metrics In this section the most commonly used performance metrics are defined. The performance metrics are accuracy, recall, precision, f1-score, AUC, etc., for the binary class classification.

Confusion matrix It is defined as the summary of results predicted by the classification model (Amanullah et al. 2020). It includes the following (Amanullah et al. 2020; https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234):

- True Positive (TP): Classification model predicted + ve and its true.
- True Negative (TN): Classification model predicted − ve and its true.
- False positive (FP): Classification model predicted + ve and its false.
- False Negative (FN): Classification model predicted − ve and its false.

True positive rate (TPR) It is also called Sensitivity or Recall (Amanullah et al. 2020). Its formula is defined as below: TP/(TP + FN) It should be high as possible.

Precision It is defined as out of all the positive classes the model has predicted correctly, how many are actually positive (https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62). Its formula is as: TP/(TP + FP).

Accuracy It is defined as out of the all the classes, how much the model has predicted correctly. It should be high as possible (https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62). Its formula is defined as below: TP + TN/Total.

False Positive Rate (FPR) (Amanullah et al. 2020) It is also called Fall-Out. It is defined as the portion of negative instances wrongly predicted positive by the model. Its formula is defined as: FP/(TN + FP).

False Negative Rate (FNR): It is defined as the portion of positive instances wrongly predicted negative. Its formula is defined as (Amanullah et al. 2020): FN/(TP + FN).

True Negative Rate (TNR): It is also called Specificity. It is defined as the portion of negative instances correctly predicted negative. Its formula is given as (Amanullah et al. 2020): TN/(TN + FP).

F-measure (https://www.analyticsvidhya.com/blog/2020/12/accuracy-and-its-shortcomings-precision-recall-to-the-rescue/) If the two models have low precision and high recall or vice versa then it is difficult to compare them. So, to make them comparable, F-score is used. It is used to measure recall and precision at the same time. It is calculated using the following formula: 2*Recall*Precision/(Recall + Precision).

AUC-ROC curve: It is defined as the performance measurement at various threshold settings for classification problem (https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5). Its formula is given as below (Han et al. 2011; Amma and Subramanian 2019):

$$\text{AUC} = ((\text{Recall} - \text{False Alarm}) + 100)/200$$

If the value of AUC is close to 1, then better is the model at prediction (https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5).

7 Research gaps in the existing literature

After the extensive review of literature as summarized in the previous section IV, the following Research gaps have been identified and also it is shown in Fig. 5.

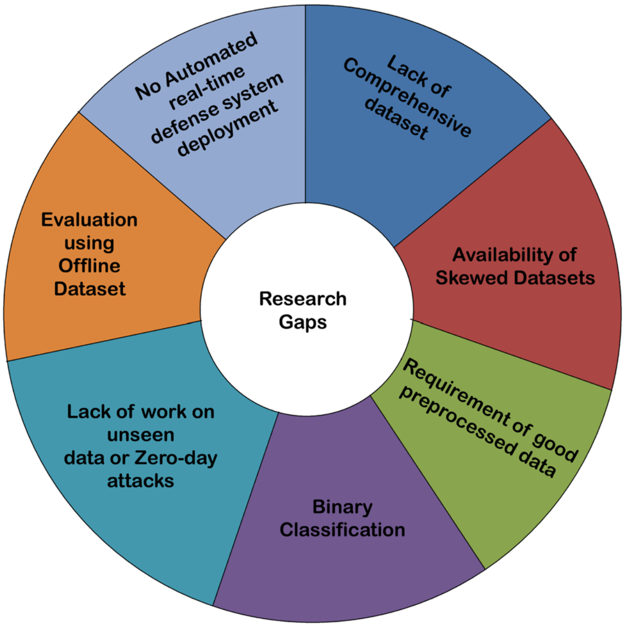

Research Gaps
Lack of Comprehensive dataset
Availability of Skewed Datasets
Requirement of good preprocessed data
Binary Classification
Lack of work on unseen data or Zero-day attacks
Evaluation using Offline Dataset
No Automated real-time defense system deployment

Fig. 5 Research gaps in existing studies

(1) *Lack of comprehensive dataset* Most of the victim organizations resist disclosing the information about attacks launched against them due to risk of reputation or revenue loss. Moreover, comprehensive datasets with all traffic types (like legitimate, low rate, high rate, and flash traffic) are missing in public domain (Amma and Subramanian 2019; Li and Lu 2019; Catak and Mustacoglu 2019; Amaizu et al. 2021; Cil et al. 2021; de Assis et al. 2020; Muraleedharan and Janet 2020; Virupakshar et al. 2020; Doriguzzi-Corin et al. 2020; Hussain et al. 2020; Wang and Liu 2020; Sbai and El Boukhari 2020; Kim et al. 2020; Haider et al. 2020; Asad et al. 2020; Chen et al. 2019; Shaaban et al. 2019; Hasan et al. 2018; Shurman et al. 2020; Assis et al. 2021; Priyadarshini and Barik 2019; Liang and Znati 2019; Li et al. 2018; Kasim 2020; Premkumar and Sundararajan 2020; Bhardwaj et al. 2020; Yang et al. 2020; Elsayed et al. 2020; Nugraha and Murthy 2020; Roopak et al. 2020; Sabeel et al. 2019). Thus, experimental setups are required to generate these inclusive datasets for comprehensive validation of DDoS detection approaches.

(2) *Availability of skewed datasets* In the existing datasets, instances of DDoS attacks are normally skewed as compared to legitimate events (Amma and Subramanian 2019; Li and Lu 2019; Catak and Mustacoglu 2019; Muraleedharan and Janet 2020; Virupakshar et al. 2020; Doriguzzi-Corin et al. 2020; Wang and Liu 2020; Kim et al. 2020; Haider et al. 2020; Asad et al. 2020; Chen et al. 2019; Shaaban et al. 2019; Hasan et al. 2018; Assis et al. 2021; Priyadarshini and Barik 2019; Liang and Znati 2019; Li et al. 2018; Bhardwaj et al. 2020; Yang et al. 2020; Nugraha and Murthy 2020; Roopak et al. 2020; Sabeel et al. 2019). However, for effective implementation of deep learning approaches, we need lot of instances of all classes. Therefore, good augmentation techniques to generate a sufficient number of instances of all types of traffic (legitimate, low rate, high rate, and flash traffic) are required for efficient research in this field.

(3) *Requirement of good preprocessed data* The accuracy of the deep learning model depends on the quality of preprocessed data. Therefore, suitable preprocessing techniques are required for efficient training of the DL model (Kim et al. 2020; Liang and Znati 2019; Chen et al. 2019; Shaaban et al. 2019; Li and Lu 2019; Amma and Subramanian 2019; Li et al. 2018; de Assis et al. 2020; Doriguzzi-Corin et al. 2020; Hussain et al. 2020; Yang et al. 2020; Wang and Liu 2020).

(4) *Binary classification* Most of the existing literature (Li and Lu 2019; de Assis et al. 2020; Virupakshar et al.

2020; Doriguzzi-Corin et al. 2020; Wang and Liu 2020; Sbai and El Boukhari 2020; Haider et al. 2020; Shaaban et al. 2019; Shurman et al. 2020; Assis et al. 2021; Priyadarshini and Barik 2019; Liang and Znati 2019; Li et al. 2018; Kasim 2020; Premkumar and Sundararajan 2020; Bhardwaj et al. 2020; Yang et al. 2020; He et al. 2020; Elsayed et al. 2020; Nugraha and Murthy 2020; Roopak et al. 2020; Sabeel et al. 2019) has focused on the binary classification rather than the multi-class classification of DDoS attacks.

(5) *Lack of work on unseen data or Zero-day attacks* Machine learning models show a high-performance rate when training and evaluation datasets have the same characteristics or patterns. But in the real-life, the attacks are launched using new patterns, due to which these machine learning-based models are not able to detect unseen attacks with accuracy. Therefore, these models must be updated at regular intervals for the new and unknown attacks (Sabeel et al. 2019).

(6) *Evaluation using offline dataset* In most of the literature deep learning models have been evaluated using offline datasets (Chen et al. 2019; Amaizu et al. 2021; Cil et al. 2021; Muraleedharan and Janet 2020; Assis et al. 2021; Shurman et al. 2020; Liang and Znati 2019; Li et al. 2018; Premkumar and Sundararajan 2020; Bhardwaj et al. 2020; Catak and Mustacoglu 2019; Yang et al. 2020; Asad et al. 2020),(Haider et al. 2020; Elsayed et al. 2020; Roopak et al. 2020; Li and Lu 2019; He et al. 2020; Doriguzzi-Corin et al. 2020; Hussain et al. 2020; Wang and Liu 2020; Sbai and El Boukhari 2020; Kim et al. 2020; Shaaban et al. 2019; Amma and Subramanian 2019). However, the deployment of these models in real networks is still a pending issue. Therefore, it would be helpful to evaluate the models in real-time for proper validation.

(7) *No automated real-time defence system deployment* Most of the DDoS attacks overwhelm the target site in a very short span of time, and network administrators cannot detect and defend these attacks in an automated manner. The major reason behind it is that the defence solutions themselves become vulnerable to flood-based DDoS attacks. Thus, there is a need of high-speed and computationally efficient DDoS solutions so that these attacks could be defended in an automated manner.

8 Conclusion and future directions

Discriminating the DDoS attacks with different rates and patterns from benign traffic is a very challenging issue. Many efficient DL approaches have been proposed by fellow researchers for DDoS attack detection over the years. But unfortunately, the scope of these methods is very limited as the attackers are continuously updating their attack strategies and skills very rapidly to launch unknown or zero-day DDoS attacks with unique traffic patterns every time. In this paper, we have used the SLR protocol to review the DDoS attacks detection system based on DL approaches and results of the SLR protocol are analyzed and concluded as below:

(1) In Sect. 3, we have categorized the DDoS attack detection DL approaches into five categories, viz: supervised instance learning, supervised sequence learning, semi-supervised learning, hybrid learning, and other learning methods.
Figure 6 shows the percentage of papers covered under each category. The present paper has reviewed 34 of such prominent research articles. It has been concluded that out of the total of 34 articles, around 50% of researchers have used supervised instance learning, 14.7% have used supervised sequence learning, 17.64% have used semi-supervised learning, 14.7% have used hybrid learning, and other learning methods have been used by 2.94%.

(2) In Sect. 4, the literature has been briefed according to the proposed taxonomy of DDoS attack detection using DL approaches. In this strengths and weaknesses of each study have been summarized. In most of the literature, the accuracy is above 99%. Most of them have been evaluated using offline analysis of the benchmarked datasets and thus, their performance metric values could change in the production or real environment. It has been observed that the articles have not used the same datasets or approaches for the evaluation; thus, comparison among them seems useless.

(3) In Sect. 5, the available DDoS benchmarked datasets and classes of attacks in datasets have been described, that have been used in the existing literature. As shown in Fig. 7 out of the total 34 articles, 29% of the existing prominent research has used the CICIDS2017 dataset, 20% has used the CICDDoS2019 dataset, 12% has used the ISCX2012 dataset, and 10%-10% for NSL KDD and KDDCUP99.
Figure 8 shows the accuracy of the studied DL-based DDoS attacks detection approaches on the CICIDS2017 dataset. It has been observed that the approaches CNN (Haider et al. 2020; Doriguzzi-Corin et al. 2020), DNN (Muraleedharan and Janet 2020), AE-SVM (Kasim 2020), and CNN-LSTM (Roopak et al. 2020) showed accuracy greater than 99%.
Figure 9 illustrates the accuracy of the studied DL-based DDoS attacks detection approaches employed on the CICDDoS2019 dataset for evaluating their approaches. It has been observed that the approaches CNN-based ResNet (Hussain et al. 2020), LSTM (Shurman et al. 2020), DNN (Sbai and El Boukhari 2020; Amaizu et al. 2021; Cil et al. 2021), and GRU (Assis et al. 2021) showed

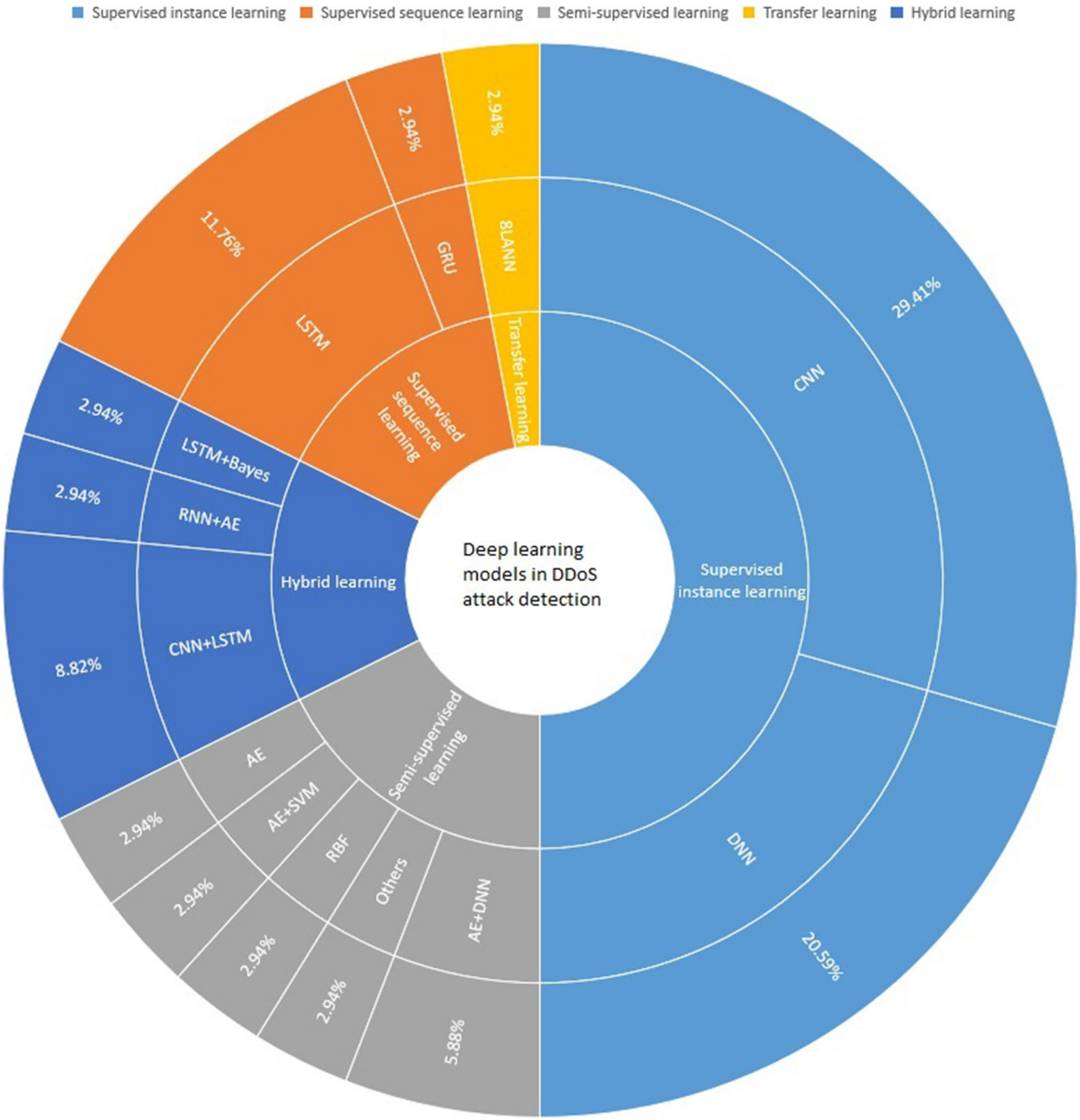


Fig. 6 The percentage of papers covered in each category

accuracy greater than 99%.

Figure 10 projects the accuracy of the studied DL-based DDoS attacks detection approaches employed on the ISCX2012 dataset. It has been observed that the approaches LSTM (Li et al. 2018; Priyadarshini and Barik 2019), CNN (Doriguzzi-Corin et al. 2020), and LSTM-Bayes (Li and Lu 2019) showed accuracy less than 99%.

Figure 11 displays the accuracy of the DDoS attack detection deep learning-based solutions on the NSL-KDD dataset. In this only CNN (Shaaban et al. 2019) approach showed an accuracy above 99%.

(4) In Sect. 6, the preprocessing strategies, hyperparameter values, experimental setups, and performance metrics have been dwelt upon that the existing DL approaches have used for DDoS attacks detection. The most common preprocessing strategies used are min–max normalization, one-hot encoding, Z-score normalization, BOW, etc.

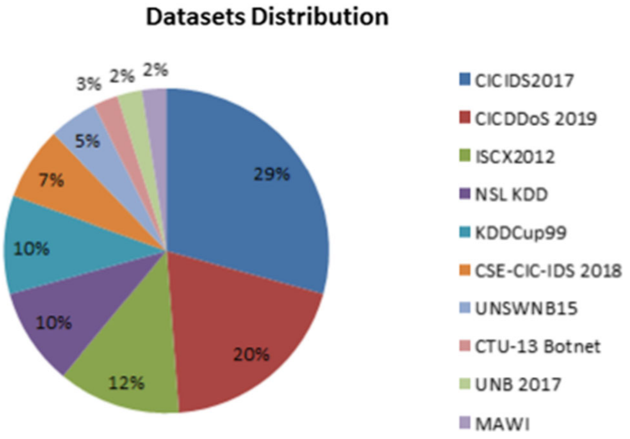


Fig. 7 Datasets distribution

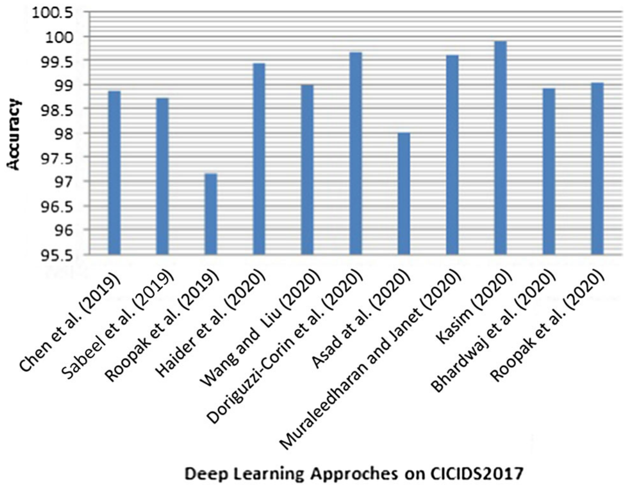


Fig. 8 Accuracy of the studied DL approaches on the CICIDS2017

Hyperparameter values of the existing models show that all the studies have used different parameter values for their models. Experimental setups configurations are also not the same for the existing studies. Thus, the conclusion drawn is that it would not be suitable to compare these techniques among themselves.

Figure 12 exhibits the number of studies that have applied each performance metric. As shown in this figure, 29 studies used accuracy metrics for evaluation of their approaches, 22 studies used precision, recall, and F1-score metrics, 6 studies used FPR and AUC metrics. In addition to it, fewer studies used other performance metrics as shown in Fig. 12. From Fig. 12, it has been observed that most of the studies have not examined the testing and training time for their approaches as such metrics are important for the deployment of the model in a real-time or production environment.

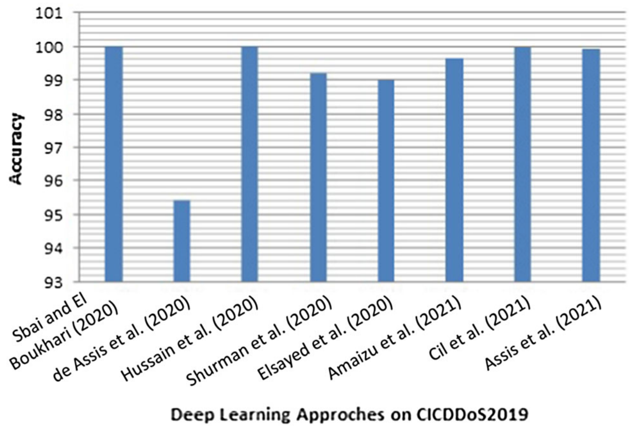


Fig. 9 Accuracy of the studied DL approaches on the CICDDoS2019

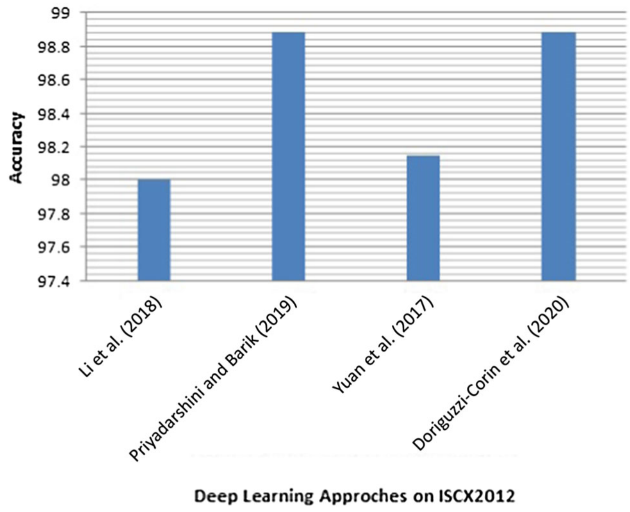


Fig. 10 Accuracy of the studied DL approaches on the ISCX2012

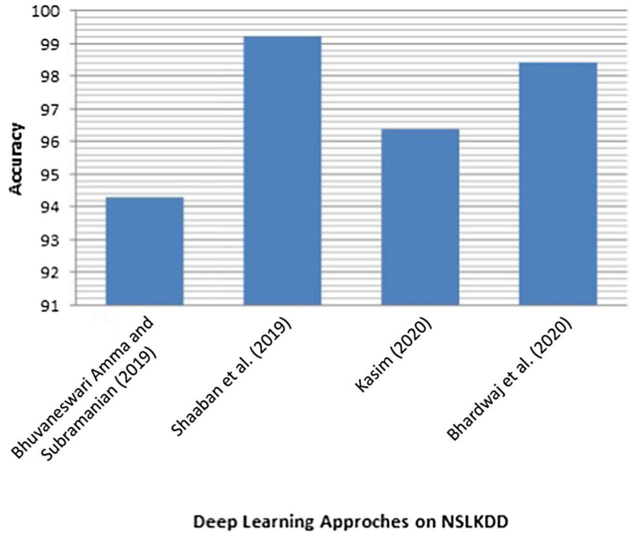


Fig. 11 Accuracy of the studied DL approaches on the NSL-KDD

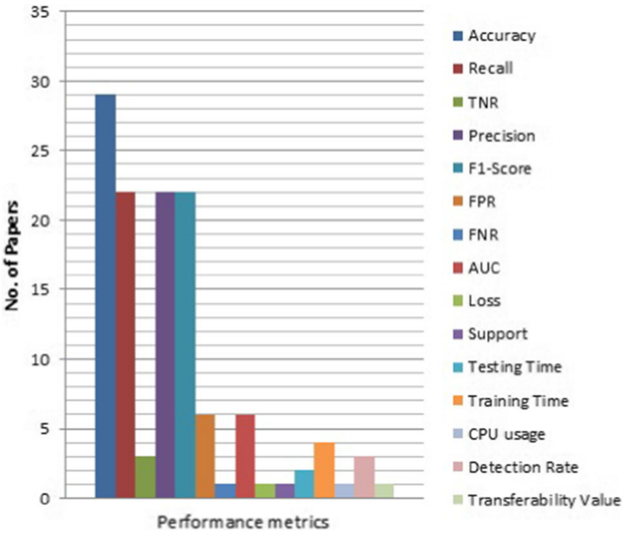


Fig. 12 The percentage of the DL approaches that used the performance metric

The following are the future research directions guided by our findings in DL methods for DDoS attacks detection:

- *Lack of DL models validated on real-time scenarios:* The above literature shows a lack of real-time deployed DL models. Most of the literature had conducted an offline analysis of their model. But we need to deploy these models in real-time scenarios. As the DDoS attacks happen in real-time, not in offline mode, therefore, there are no benefits of doing offline analysis unless we do not check our approaches over real-time scenarios. There is thus a requirement for DL models that are validated over real-time scenarios.
- *Requirement of an automatically and regularly updated DL models:* With the fast change in patterns of attacks, there is also the need for a model that can be automatically and regularly updated according to the new instances of attacks. It is essential in today's world of fast-growing new technologies that bring along with them more advanced attacks. But the literature lacks these types of DL models.
- *Requirement of lightweight DL models:* There is a requirement of lightweight DL approaches in the networks like IoT, MANETS, WSN, etc., because these networks have limited computing resources and memory, but also these networks are more prone to attacks. Thus, there is a requirement to develop efficient and lightweight DL models.
- *Requirement of suitable datasets:* The existing datasets do not have varieties of attacks and balanced data records. Thus, the detection techniques become biased and cannot detect all kinds of attacks as the existing datasets lack various attacks. Therefore, a suitable dataset is required for the efficient and accurate detection model.

The above observations would pave way for the researchers to carry out research in this field and would to a great extent shrink the existing research gaps.

Author Contributions All authors contributed to the study's conception and design. The first draft of the manuscript was written by [Meenakshi], and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding No funding was received for conducting this study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

7 of the most famous recent DDoS attacks. https://www.vxchnge.com/blog/recent-ddos-attacks-on-companies. 12 July 2021

Accuracy and its shortcomings: precision, recall to the rescue. https://www.analyticsvidhya.com/blog/2020/12/accuracy-and-its-shortcomings-precision-recall-to-the-rescue/. 8 July 2021

Ahmad R, Alsmadi I (2021) Machine learning approaches to IoT security: a systematic literature review. Internet Things 14:100365

Ahmad Z, Khan AS, Shiang CW, Abdullah J, Ahmad F (2021) Network intrusion detection system: a systematic study of machine learning and deep learning approaches. Trans Emerg Telecommun Technol 32:e4150

Aldweesh A, Derhab A, Emam AZ (2020) Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues. Knowl Based Syst 189:2020

Aleesa AM, Zaidan BB, Zaidan AA, Sahar NM (2020) Review of intrusion detection systems based on deep learning techniques: coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions. Neural Comput Appl 32:9827–9858

Ali S, Li Y (2019) Learning multilevel auto-encoders for DDoS attack detection in smart grid network. IEEE Access 7:108647–108659

Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Van Esesn BC, Awwal AAS, Asari VK (2018) The history began from AlexNet: a comprehensive survey on deep learning approaches

Amaizu GC, Nwakanma CI, Bhardwaj S, Lee JM, Kim DS (2021) Composite and efficient DDoS attack detection framework for B5G networks. Comput Netw 188:107871

Amanullah MA, Habeeb RAA, Nasaruddin FH, Gani A, Ahmed E, Nainar ASM, Akim NM, Imran M (2020) Deep learning and big data technologies for IoT security. Comput Commun 151:495–517

Amma NGB, Subramanian S (2019) VCDeepFL: Vector Convolutional Deep Feature Learning approach for identification of known and unknown Denial of Service Attacks. In: IEEE Region 10 Annual International Conference, Proceedings/TENCON, vol

2018-October. Institute of Electrical and Electronics Engineers Inc., pp 640–645

Asad M, Asim M, Javed T, Beg MO, Mujtaba H, Abbas S (2020) Deep-Detect: detection of Distributed Denial of Service attacks using deep learning. Comput J 63:983–994

Assis MV, Carvalho LF, Lloret J, Proença ML (2021) A GRU deep learning system against attacks in software defined networks. J Netw Comput Appl 177:102942

Bhardwaj A, Mangat V, Vig R (2020) Hyperband tuned deep neural network with well posed stacked sparse AutoEncoder for detection of DDoS attacks in Cloud. IEEE Access 8:181916–181929

Catak FO, Mustacoglu AF (2019) Distributed denial of service attack detection using autoencoder and deep neural networks. J Intell Fuzzy Syst 37:3969–3979

Chen J, tao Yang Y, ke Hu K, bin Zheng H, Wang Z (2019) DAD-MCNN: DDoS attack detection via multi-channel CNN. In: ACM international conference proceeding series, vol Part F1481. Association for Computing Machinery, New York, pp 484–488

Cil AE, Yildiz K, Buldu A (2021) Detection of DDoS attacks with feed forward based deep neural network model. Expert Syst Appl 169:114520

CNN for deep learning—convolutional neural networks (CNN). https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/. 8 July 2021

DDoS attacks in Q4 2019—Securelist. https://securelist.com/ddos-report-q4-2019/96154/. 27 Feb 2020

Ddos 2019—datasets—research—Canadian Institute for cybersecurity—UNB. https://www.unb.ca/cic/datasets/ddos-2019.html. 8 July 2021

de Assis MV, Carvalho LF, Rodrigues JJ, Lloret J, Proença ML (2020) Near real-time security system applied to SDN environments in IoT networks using convolutional neural network. Comput Electr Eng 86:106738

Deep learning definition.https://www.investopedia.com/terms/d/deep-learning.asp. 12 July 2021

Deshmukh DH, Ghorpade T, Padiya P (2015) Improving classification using preprocessing and machine learning algorithms on NSL-KDD dataset. In: Proceedings—2015 International Conference on Communication, Information and Computing Technology, ICCICT 2015

Doriguzzi-Corin R, Millar S, Scott-Hayward S, Martinez-Del-Rincon J, Siracusa D (2020) Lucid: a practical, lightweight deep learning solution for DDoS attack detection. IEEE Trans Netw Serv Manag 17:876–889

Elsayed MS, Le-Khac NA, Dev S, Jurcut AD (2020) DDoSNet: a deep-learning model for detecting network attacks. In: Proceedings—21st IEEE international symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2020. Institute of Electrical and Electronics Engineers Inc., pp 391–396

Ferrag MA, Maglaras L, Moschoyiannis S, Janicke H (2020) Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study. J Inf Secur Appl 50:102419

Gamage S, Samarabandu J (2020) Deep learning methods in network intrusion detection: a survey and an objective comparison. J Netw Comput Appl 169:102767

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge

Google services resume after massive gmail, youtube outage. https://www.livemint.com/technology/apps/google-services-youtube-gmail-google-drive-face-outage-11607947475759.html. 18 Apr 2021

Gopika P, Krishnendu C, Hari Chandana M, Ananthakrishnan S, Sowmya V, Gopalakrishnan E, Soman K (2020) Single-layer convolution neural network for cardiac disease classification using electrocardiogram signals. In: Deep learning for data analytics. Academic Press, pp 21–35

Gümüşbaş D, Yíldírím T, Genovese A, Scotti F (2020) A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. IEEE Syst J

Haider S, Akhunzada A, Mustafa I, Patel TB, Fernandez A, Choo KKR, Iqbal J (2020) A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks. IEEE Access 8:53972–53983

Han J, Kamber M, Pei J (2011) Data Mining. Concepts and Techniques, 3 edn (The Morgan Kaufmann Series in Data Management Systems)

Han J, Kamber M, Pei J (2012) Introduction. In: Data mining. Morgan Kaufmann, pp 1–38

Hasan MZ, Hasan KMZ, Sattar A (2018) Burst header packet flood detection in optical burst switching network using deep learning model. Procedia Comput Sci 143:970–977

He J, Tan Y, Guo W, Xian M (2020) A small sample DDoS attack detection method based on deep transfer learning. In: Proceedings—2020 International Conference on Computer Communication and Network Security, CCNS 2020. Institute of Electrical and Electronics Engineers Inc., pp 47–50

Holzinger A (2019) Big data calls for machine learning. Encycl Biomed Eng 1–3:258–264

Hoque N, Kashyap H, Bhattacharyya DK (2017) Real-time DDoS attack detection using FPGA. Comput Commun 110:48–58

Hussain F, Ghazanfar S, Al-Khawarizmi A, Husnain M, Fayyaz UU, Shahzad F, Al-Khawarizmi GAS (2020) IoT DoS and DDoS attack detection using ResNet. Tech. rep., 2020

Ids 2012—datasets—research—Canadian Institute for cybersecurity—UNB. https://www.unb.ca/cic/datasets/ids.html. 12 July 2021

Ids 2017—datasets—research—Canadian Institute for cybersecurity—UNB. https://www.unb.ca/cic/datasets/ids-2017.html. 8 July 2021

Ids 2018—datasets—research—Canadian Institute for cybersecurity—UNB. https://www.unb.ca/cic/datasets/ids-2018.html. 8 July 2021

Illustrated guide to recurrent neural networks—by Michael Phi—towards data science. https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9. 8 July 2021

Kasim Ö (2020) An efficient and robust deep learning based network anomaly detection against distributed denial of service attacks. Comput Netw 180:107390

Kdd cup 1999 data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. 8 July 2021

Keele S (2007) Guidelines for performing systematic literature reviews in software engineering. Technical report, Ver. 2.3 EBSE Technical Report. EBSE

Ke Q, Liu J, Bennamoun M, An S, Sohel F, Boussaid F (2018) Computer vision for human–machine interaction. In: Computer vision for assistive healthcare. Academic Press, pp 127–145

Kim M (2019) Supervised learning-based DDoS attacks detection: tuning hyperparameters. ETRI J 41:560–573

Kim J, Kim J, Kim H, Shim M, Choi E (2020) CNN-based network intrusion detection against Denial-of-Service attacks. Electronics 9:916

Li C, Wu Y, Yuan X, Sun Z, Wang W, Li X, Gong L (2018) Detection and defense of DDoS attack-based on deep learning in OpenFlow-based SDN. Int J Commun Syst 31:e3497

Liang X, Znati T (2019) A long short-term memory enabled framework for DDoS detection. In: 2019 IEEE Global Communications Conference, GLOBECOM 2019—Proceedings. Institute of Electrical and Electronics Engineers Inc

Li Y, Lu Y (2019) LSTM-BA: DDoS detection approach combining LSTM and bayes. In: Proceedings—2019 7th international conference on advanced Cloud and Big Data, CBD 2019. Institute of Electrical and Electronics Engineers Inc., pp 180–185

Longa A (2021) Long short term memory—architecture of LSTM. https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/. 8 July 2021

Metrics to evaluate your machine learning algorithm—by Aditya Mishra—towards data science. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234. 12 July 2021

Moustafa N, Slay J (2015) UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference, MilCIS 2015—Proceedings

Muraleedharan N, Janet B (2020) A deep learning based HTTP slow DoS classification approach using flow data. In: ICT Express

Nisha SS, Sathik MM, Meeral MN (2021) Application, algorithm, tools directly related to deep learning. In: Handbook of deep learning in biomedical engineering. Academic Press, pp 61–84

Nsl-kdd—datasets—research—Canadian Institute for cybersecurity—UNB. https://www.unb.ca/cic/datasets/nsl.html. 8 July 2021

Nugraha B, Murthy RN (2020) Deep learning-based slow DDoS attack detection in SDN-based networks. In: 2020 IEEE conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2020—Proceedings. Institute of Electrical and Electronics Engineers Inc., pp 51–56

Panigrahi R, Panigrahi R, Borah S (2018) A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems. Int J Eng Technol 7:479–482

Premkumar M, Sundararajan TV (2020) DLDM: deep learning-based defense mechanism for denial of service attacks in wireless sensor networks. Microprocess Microsyst 79:103278

Priyadarshini R, Barik RK (2019) A deep learning based intelligent framework to mitigate DDoS attack in fog environment. J King Saud Univ Comput Inf Sci

Protić DD (2018) Review of KDD CUP '99, NSL-KDD and KYOTO 2006+ datasets, vol 66, p 3

Quantum artificial intelligence in 2021: in-depth guide. https://research.aimultiple.com/quantum-ai/. 15 Oct 2021

Recurrent neural networks and LSTM explained—by purnasai gudikandula—medium. https://purnasaigudikandula.medium.com/recurrent-neural-networks-and-lstm-explained-7f51c7f6bbb9. 12 July 2021

Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A (2019) A survey of network-based intrusion detection data sets. Comput Secur 86:147–167

Roopak M, Tian GY, Chambers J (2019) Deep learning models for cyber security in IoT networks. In: 2019 IEEE 9th annual Computing and Communication Workshop and Conference, CCWC 2019, pp 452–457

Roopak M, Tian GY, Chambers J (2020) An intrusion detection system against DDoS attacks in IoT networks. In: 2020 10th annual Computing and Communication Workshop and Conference, CCWC 2020. Institute of Electrical and Electronics Engineers Inc., pp 562–567

Sabeel U, Heydari SS, Mohanka H, Bendhaou Y, Elgazzar K, El-Khatib K (2019) Evaluation of deep learning in detecting unknown network attacks. In: 2019 international conference on Smart Applications, Communications and Networking, SmartNets 2019. Institute of Electrical and Electronics Engineers Inc

Sbai O, El Boukhari M (2020) Data flooding intrusion detection system for manets using deep learning approach. In: ACM international conference proceeding series. Association for Computing Machinery, New York, pp 281–286

Shaaban AR, Abd-Elwanis E, Hussein M (2019) DDoS attack detection and classification via Convolutional Neural Network (CNN). In: Proceedings—2019 IEEE 9th International Conference on Intelligent Computing and Information Systems, ICICIS 2019. Institute of Electrical and Electronics Engineers Inc., pp 233–238

Sharafaldin I, Lashkari AH, Hakak S, Ghorbani AA (2019) Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: Proceedings—International Carnahan Conference on Security Technology, vol 2019-October

Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA (2012) Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput Secur 31:357–374

Shurman M, Khrais R, Yateem A (2020) DoS and DDoS attack detection using deep learning and IDS. Int Arab J Inf Technol 17(4A):2020

Subasi A (2020) Machine learning techniques. Academic Press, London

Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the KDD CUP 99 data set. In: IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009

The largest DDoS attack in history just happened... and it didn't work. https://www.thesslstore.com/blog/largest-ddos-attack-in-history/. 8 Aug 2020

The beat goes on—netscout. https://www.netscout.com/blog/asert/beat-goes. 8 July 2021

Types of neural networks and definition of neural network. https://www.mygreatlearning.com/blog/types-of-neural-networks/. 12 July 2021

UK cryptocurrency exchange EXMO knocked offline by 'massive' DDoS attack—the Daily Swig. https://portswigger.net/daily-swig/uk-cryptocurrency-exchange-exmo-knocked-offline-by-massive-ddos-attack. 18 July 2021

Understanding AUC-ROC curve—by Sarang Narkhede—towards data science. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5. 8 July 2021

Understanding confusion matrix—by Sarang Narkhede—towards data science. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62. 8 July 2021

Understanding hyperparameters and its optimisation techniques—by Prabhu—towards data science. https://towardsdatascience.com/understanding-hyperparameters-and-its-optimisation-techniques-f0debba07568. 8 July 2021

Van NT, Thinh TN, Sach LT (2017) An anomaly-based network intrusion detection system using deep learning. In: Proceedings—2017 International Conference on System Science and Engineering, ICSSE 2017. Institute of Electrical and Electronics Engineers Inc., pp 210–214

Vinayakumar R, Soman KP, Poornachandrany P (2017) Applying convolutional neural network for network intrusion detection. In: 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017, vol 2017-January. Institute of Electrical and Electronics Engineers Inc., pp 1222–1228

Virupakshar KB, Asundi M, Channal K, Shettar P, Patil S, Narayan DG (2020) Distributed Denial of Service (DDoS) attacks detection system for OpenStack-based Private Cloud. Procedia Comput Sci 167:2297–2307

Wang L, Liu Y (2020) A DDoS attack detection method based on information entropy and deep learning in SDN. In: Proceedings of 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2020. Institute of Electrical and Electronics Engineers Inc., pp 1084–1088

What are convolutional neural networks? IBM. https://www.ibm.com/cloud/learn/convolutional-neural-networks. 12 July 2021

What is a denial-of-service (DoS) attack? Cloudflare. https://www.cloudflare.com/en-in/learning/ddos/glossary/denial-of-service/. 12 July 2021

What is a distributed denial-of-service (DDoS) attack? Cloudflare. https://www.cloudflare.com/en-in/learning/ddos/what-is-a-ddos-attack/. 22 July 2021

What are hyperparameters? And how to tune the hyperparameters in a deep neural network?—by Pranoy Radhakrishnan—towards data science. https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a. 12 July 2021

Wu J, Chen XY, Zhang H, Xiong LD, Lei H, Deng SH (2019) Hyperparameter optimization for machine learning models based on Bayesian optimization. J Electron Sci Technol 17:26–40

Xin Y, Kong L, Liu Z, Chen Y, Li Y, Zhu H, Gao M, Hou H, Wang C (2018) Machine learning and deep learning methods for cybersecurity. IEEE Access 6:35365–35381

Yamashita R, Nishio M, Do RKG, Togashi K (2018) Convolutional neural networks: an overview and application in radiology. Insights Imaging 9(4):611–629

Yang K, Zhang J, Xu Y, Chao J (2020) DDoS attacks detection with AutoEncoder. In: Proceedings of IEEE/IFIP Network Operations and Management Symposium 2020: management in the age of softwarization and artificial intelligence, NOMS 2020. Institute of Electrical and Electronics Engineers Inc

Yuan X, Li C, Li X (2017) DeepDefense: identifying DDoS attack via deep learning. In: 2017 IEEE International Conference on Smart Computing, SMARTCOMP 2017. Institute of Electrical and Electronics Engineers Inc

Yuvaraj N, Raja RA, Kousik N, Johri P, Diván MJ (2020) Analysis on the prediction of central line-associated bloodstream infections (CLABSI) using deep neural network classification. In: Computational intelligence and its applications in healthcare. Academic Press, pp 229–244

Zhu W, Ma Y, Zhou Y, Benton M, Romagnoli J (2018) Deep learning based soft sensor and its application on a pyrolysis reactor for compositions predictions of gas phase components. Comput Aided Chem Eng 44:2245–2250