IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 November 2023, accepted 24 December 2023, date of publication 2 January 2024,
date of current version 11 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3349287

SURVEY

A Survey of Deep Learning Technologies for Intrusion Detection in Internet of Things

HAN LIAO[1,2], (Member, IEEE), MOHD ZAMRI MURAH[1], MOHAMMAD KAMRUL HASAN[1], (Senior Member, IEEE), AZANA HAFIZAH MOHD AMAN[1], JIN FANG[1], (Member, IEEE), XUTING HU[3], AND ATTA UR REHMAN KHAN[4], (Senior Member, IEEE)

[1]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi 43600, Malaysia
[2]Anhui Business College, Wuhu, Anhui 241002, China
[3]Anhui Technical College of Mechanical and Electrical Engineering, Wuhu, Anhui 241002, China
[4]College of Engineering and Information Technology, Ajman University, Ajman, United Arab Emirates

Corresponding author: Mohammad Kamrul Hasan (mkhasan@ukm.edu.my)

This work was supported by Universiti Kebangsaan Malaysia under Grant DIP-2022-021.

ABSTRACT The Internet of Things (IoT) is transforming how we live and work, and its applications are widespread, spanning smart homes, industrial monitoring, smart cities, healthcare, agriculture, and retail. Considering its wide range of applications, addressing the security challenges arising from IoT devices' massive collection and transmission of user data is vital. Intrusion detection systems (IDS) based on deep learning techniques offer new means and research directions for resolving IoT security issues. Deep learning models can process large volumes of data and extract complex patterns, making them generally more effective than traditional rule based IDSs. While deep learning techniques are gradually gaining popularity in IDS applications, current research needs a comprehensive summary of deep learning-based IDS in IoT. This paper introduces intrusion detection technologies, followed by a detailed comparison, analysis, and discussion of deep learning models, datasets, feature extraction and classifiers, data preprocessing techniques, and experimental design of the models. It also highlights the challenges and issues associated with deep learning models and relevant techniques for IDS. Finally, it concludes by providing recommendations to assist researchers in this domain.

INDEX TERMS IoT, IDS, deep learning, datasets, data preprocessing, feature extraction, classifiers.

I. INTRODUCTION

The concept of IoT is to connect a multitude of physical devices through a network, enabling them to collect and exchange data efficiently for intelligent control and management [1]. The applications of IoT have been widely used in various industries and areas of life. In the home, IoT technology enables devices such as lighting, heating, air conditioning, TVs, and refrigerators to connect and interact over a network for remote control, automated operation and other functions [2]. IoT can monitor industrial production processes to increase productivity and reduce operational costs [3]. In urban management, IoT can be used for traffic control, environmental monitoring, and public safety applications. In addition, IoT has various applications in healthcare, agriculture, and retail [4].

With rapid growth and development, IoT has also encountered challenges, especially regarding data security and privacy. This is a vital issue because IoT devices accumulate and distribute large amounts of user data, which attackers can compromise without adequate safeguards [5]. Recently, artificial intelligence and machine learning-based IDS have been used for IoT. These systems can automatically learn and identify standard network behavior patterns to effectively detect anomalous behavior [6]. Figure 1 shows the deployment of IDS in an IoT environment, illustrating that IoT devices and servers are deployed on the open Internet. IDSs can protect IoT devices from attacks by detecting intrusions and alerting them to anomalous behaviors before the attackers can penetrate the IoT.

The associate editor coordinating the review of this manuscript and approving it for publication was Rongbo Zhu.

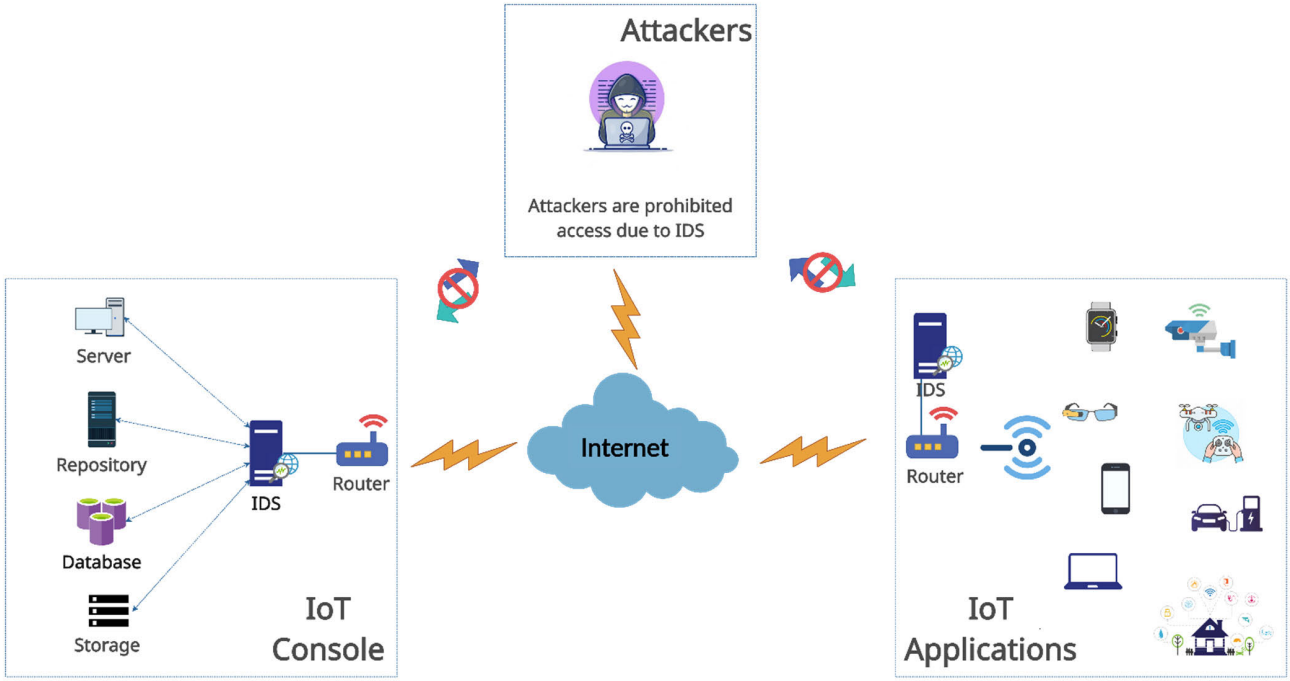


FIGURE 1. An IDS deployment scheme for IoT environment.

There is a growing interest in the application of deep learning and related technologies for IoT security. This development can be attributed to its benefits, including improved detection efficiency, fewer false positives and less reliance on feature engineering [7]. In addition, it automatically and intelligently identifies attack features, thus helping to detect potential security threats [8]. Deep learning models can build efficient IDS models using large amounts of traffic data to distinguish between normal and abnormal network behavior [9]. In addition, deep learning models can accurately identify potential intrusions by scrutinizing features and patterns in network traffic. The efficacy of IDS depends to some extent on the appropriateness of feature extraction and classification methods [10]. In turn, finding suitable and effective intrusion detection datasets is a meaningful way to test the effectiveness of IDS deep learning models, which is also a significant challenge [11]. In addition, the selection and tuning of superparameters play a crucial role in constructing deep learning-based IDS models. For example, in [12], the researchers investigated the ideal number of hidden layers and neurons in Generative Adversarial Networks (GANs).

The major contributions of this paper are highlighted as follows:

- A detailed overview of intrusion detection techniques, their classification, and characteristics.
- Overview of deep learning techniques, comparison of common deep learning models, and highlighting and their advantages and disadvantages.
- Analysis of standard intrusion detection datasets, introduction of dataset characteristics, their composition, distribution, and comparison of application scenarios.
- Comparison of data preparation techniques from various studies, including numerical encoding methods and solutions for data imbalances. In addition, analysis of application of feature extraction methods and classifiers in deep learning-based IDSs and comparison of their respective implementations.

The rest of the paper is organized as follows. Section II presents intrusion detection techniques, including issues related to classification and traditional rule based IDSs. Section III examines common deep learning models applied in IDSs and provides their analysis. Section IV describes standard intrusion detection datasets and their respective characteristics. Section V examines data preprocessing in intrusion detection systems, techniques for addressing dataset imbalance, feature extraction, and classifiers. Finally, Section VI concludes the paper by highlighting limitations and providing future research directions.

II. INTRUSION DETECTION SYSTEMS

Gathering information from critical network points is the hallmark of intrusion detection. Data is scrutinized using predefined rules to determine the presence of adversaries and classify ongoing network attacks. This approach is often referred to as intrusion detection [13]. Figure 2 illustrates the intrusion detection process.

Data sources in networks typically contain valuable information, including details of changes to sensitive files, the operation of uncommon programs, and network traffic. Once extracted, this information requires processing and analysis, known as information analysis. To detect intrusions, various data mining techniques, pattern matching, and

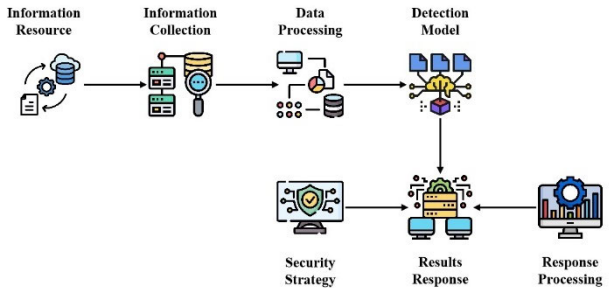


FIGURE 2. The process of intrusion detection.

integrated learning methods are commonly used. The response is founded upon the outcome of the intrusion detection analysis intended for post-processing purposes, which comprise data storage for subsequent reference, reconfiguration of routers, and other equivalent actions [14].

An IDS sits behind the firewall and secures the entire system against intrusion. It effectively complements to the firewall, monitoring data traffic and blocking abnormal traffic connections in conjunction with the firewall when the intrusion detection finds abnormal behavior [13].

Intrusion detection can be classified based on technology or data sources. Technically, it is categorized into two categories: anomaly detection and misuse detection. Host-based Intrusion Detection Systems (HIDS) and Network-based Intrusion Detection Systems (NIDS) are used for data sources.

Anomaly detection is a technique to identify whether a user's behavior or system resource usage indicates intrusion. This method quantitatively describes user behavior characteristics and employs features to differentiate between normal and abnormal behavior [15]. The fundamental concept behind anomaly detection is that everyone's behavior follows a particular pattern, and by analyzing information on normal and abnormal behavior and summarizing these patterns, it becomes possible to distinguish between normal and intrusive behavior [16].

Misuse detection, also known as feature-based detection, is a relatively simple method [17]. It assumes that all network attacks and methods have specific characteristics, and misuse detection analyses the attack behavior, using expert experience to discover features, extract them and build a database of attack features. The success of misuse detection is, therefore, primarily a matter of building a feature base, i.e., compiling a comprehensive feature base of attack behavior and eliminating the interference of subjective expert knowledge key issues. Pattern matching and expert systems are the main misuse detection methods [18].

Anomaly detection and misuse detection are based on different ideas, and their actual detection performance has its advantages and disadvantages. In principle, anomaly detection is based on user behavior and resource usage to determine whether an intrusion exists. In contrast, misuse detection is based on the attack's feature base to determine whether an intrusion exists [19]. In terms of detection accuracy, misuse detection has better detection accuracy for known attacks. However, it is weaker for new attacks, while anomaly detection has better detection ability for new attacks [17].

Host-based intrusion detection technology aims to detect the host system and local users, and its data comes from the host, which mainly obtains audit data, system log information, software logs and other information from the host. This information is recorded in detail on the user, system, and software operations, and the corresponding keywords are extracted through the analysis system to analyze whether there is an intrusion [20].

Network-based intrusion detection techniques no longer depend on the data of the protected host. Instead, the data used comes from the entire protected network. Characteristic data is obtained through network traffic analysis, packet parsing, and other network management techniques. Suspicious behavior is then identified and analyzed within the network [21].

Most intrusion detection technologies are network-based and are transparent to intruders, making them less likely to be attacked and ensuring their security; as they are deployed at critical nodes in the network, they can protect all hosts in the network for specific software to be installed on the protected hosts [14]. Figure 3 shows the range of roles of HIDS and NIDS.

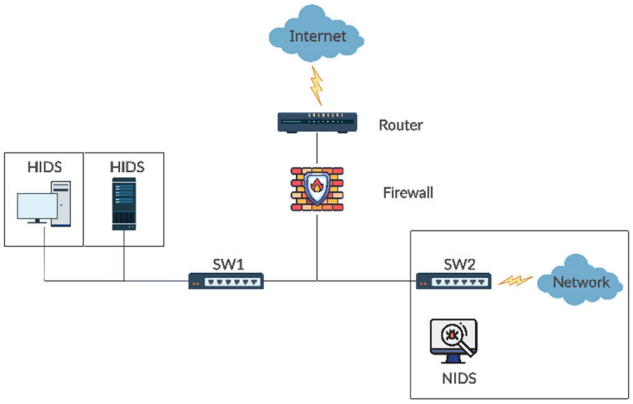


FIGURE 3. Intrusion detection range for HIDS and NIDS.

III. DEEP LEARNING MODELS

This section presents an outline of six deep learning models frequently used in IDS: (i) Deep Neural Network (DNN), (ii) Convolutional Neural Network (CNN), (iii) Recurrent Neural Network (RNN), (iv) Autoencoders (AE), (v) Deep Belief Network (DBN), and (vi) Self-Taught Learning (STL).

A. DEEP NEURAL NETWORK

The DNN is a robust structure in neural networks, designed as a feed-forward neural network (FNN) that avoids recursive connections. Its most notable feature is its ability to contain multiple hidden layers, which can considerably impact learning. Each hidden layer consists of many neurons that receive and process the previous layer's output [22].

By performing a non-linear transformation of the activation function, these neurons can capture the intricate and subtle relationships in the data. The stacked arrangement of hidden layers in a DNN helps to learn complex non-linear patterns and extract highly abstract and meaningful features from the input data [23]. Figure 4 shows a typical DNN model architecture.

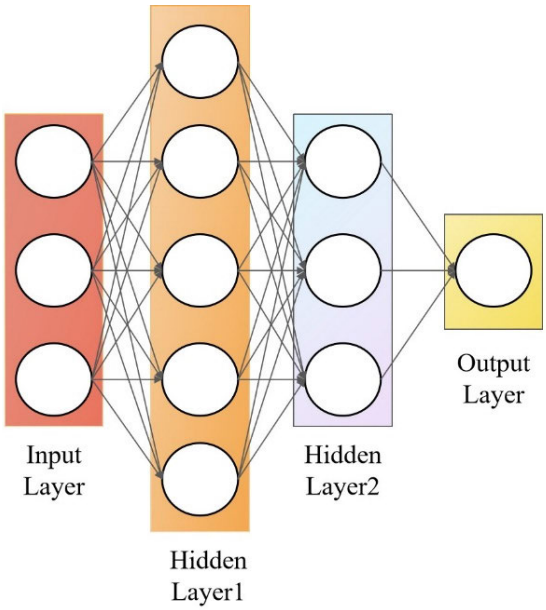


FIGURE 4. A typical DNN model architecture.

In these recent studies, [24] proposed a DNN consisting of 200 hidden layers using an activation function that is a ReLu function. This neural network was trained and tested on the NSL-KDD dataset. Experimental results show that an IDS based on this DNN model can achieve up to 93% classification accuracy. In contrast, the authors of [25] conducted experiments on DNN models with different hidden layers. They utilized a superparameter selection method to determine the optimal number of hidden layers and used the ReLu function as the activation function. In addition, they chose the Softmax function as the output layer classifier. In contrast, [26] developed a basic DNN featuring one input layer, three hidden layers containing ReLu activation functions, and one Sigmoid activation function output layer and evaluated the model's performance using three different datasets. In the study conducted by the authors [27], they compared and analyzed the performance of the DNN with other deep learning models. They also constructed a DNN with one input layer, three hidden layers and one output layer, each consisting of 100 neurons. Through experiments, the authors' proposed IDS achieved 99.22% and 99.59% accuracy for binary classification and multivariate classification on the UNSW-NB15 test dataset, respectively.

References [28] and [29] combine two deep learning models, DNN and stacked auto-encoder (SAE), to apply them to the IDS. However, [28] introduces an additional attention mechanism to construct an SAAE-DNN model. Moreover, the ReLu function serves as the activation function of the hidden layer in [28], whereas the activation function of the hidden layer in [29] is the tanh function. Both [28] and [29] employ DNN as classification algorithms to carry out the classification task for detecting intrusions. Table 1 presents the details of the analysis.

TABLE 1. Comparison of DNN-based IDS in terms of hidden layers, datasets and classification.

Citation	Model	Dataset	Hidden layers	Classification	Result			
					Accuracy	Precision	Recall	F1-score
[24]	DNN	NSL-KDD	200	2-class	95.4%	96.2%	93.5%	95.75%
[25]	DNN	NSL-KDD	5	Multi-class	78.5%	81%	78.5%	76.5%
[26]	DNN	NSL-KDD	3	2-class	99.84%	99.94%	98.81%	99.37%
[27]	DNN	UNSW-NB15	3	Multi-class	99.59%	-	-	-
[28]	SAAE-DNN	NSL-KDD	2	Multi-class	82.14%	87.28%	67.89%	76.37%
[29]	SAE-DNN	CIC-IDS 2017	2	Multi-class	98.22%	100%	100%	100%

Table 1 demonstrates that the accuracy of the IDS with DNN in detecting 2-classes of data in the NSL-KDD dataset exceeds the accuracy in detecting multi-classes of data, which is attributable to the data imbalance problem inherent in the NSL-KDD dataset. In contrast, the accuracy of other intrusion detection datasets (e.g., UNSW-NB15, CIC-IDS2017) usually exceeds that of the NSL-KDD dataset.

B. CONVOLUTIONAL NEURAL NETWORK

The CNN is a unique neural network structure that replaces matrix multiplication with convolution calculation, which makes it different from traditional artificial neural networks. This convolutional operation gives CNNs a unique feature that improves the data processing performance [8]. The structure of CNNs is distinctive in that it can take full advantage of the two-dimensional features of the input data. Compared to other deep learning structures, CNNs are known to have excellent results in speech and image recognition [30]. The structure of CNNs consists of three main layers. The convolutional layer is mainly responsible for feature extraction, i.e., capturing the critical parts of the input data

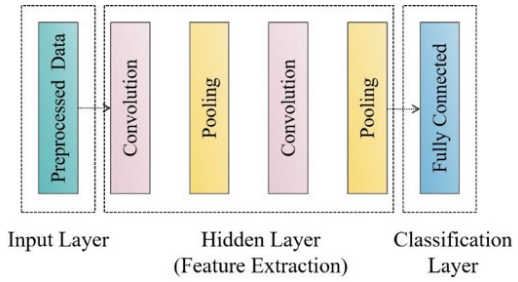


FIGURE 5. A typical convolutional neural network model.

through convolutional operations. The pooling layer serves for feature selection, which reduces the parameter complexity by decreasing the number of features. The final classification task uses a fully connected layer to map the extracted features to individual classes. The result is a hierarchy that helps the CNN to perform excellent feature extraction and classification tasks. Figure 5 shows a typical CNN architecture diagram.

The CNN was applied to IDS by [31], [32], and [33]. All three models use the ReLu activation function but were trained and tested on different datasets. One of the papers [32] also conducted an experimental comparison between the CNN and the other two models and found that the accuracy of the CNN model was significantly higher than that of the LSTM and GRU models, reaching 97.01%.

In contrast, studies in [34], [35], [36], [37], [38], and [39] have attempted to merge CNN and RNN deep learning models to fully utilize the extracting feature advantages of CNN and the powerful classifying capabilities of RNN.

In the [40], CNN combines with the Spark data processing platform for 2-class and then with machine learning for subsequent multiple classification of abnormal targets. The benefit of this approach is that it improves the distinction between normal and anomalous events while reducing the chance of coupling. In addition, it reduces the time required for data preprocessing and transformation. Table 2 presents a summary of the details of the analysis.

Table 2 indicates that IDSs utilizing both CNN and LSTM perform better than those utilizing only CNN. Additionally, the intrusion detection datasets of the NSL-KDD and UNSW-NB15 exhibit favorable results for multi-class tasks.

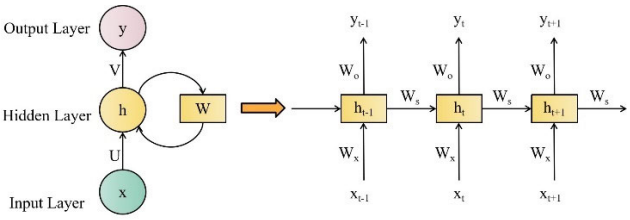
C. RECURRENT NEURAL NETWORK

The RNN is a neural network structure that processes sequential data, including time series or textual data. Its mechanism involves cycling information throughout the network, enabling it to save contextual information from previous inputs and apply it to the current input [41]. Figure 6 illustrates the structure of an RNN.

Figure 6 shows how the RNN achieves weight sharing by implementing a weight matrix W (a cyclic kernel). This matrix uses the previous input to the hidden layer as the weight for the current input. The structure of the RNN determines its ability to process continuous data. Each time

TABLE 2. Comparison of CNN-based IDS in terms of dataset, classification, and modeling.

Citation	Datasets	Classification	Model		Result			
			CNN	LSTM	Accuracy	Precision	Recall	F1-score
[31]	UNSW-NB15	Multi-class	√	×	95.6 %	-	-	-
[34]	KF-ISAC	Multi-class	√	√	98.07 %	97.06 %	99.22 %	98.13 %
[35]	CIC-IDS 2018	-	√	√	99.98 %	-	-	-
[36]	AWID +GAN	-	√	√	93.53 %	57.45 %	31.87 %	41 %
[37]	CIDDS-001	Multi-class	√	√	99.83 %	99 %	99 %	99 %
[38]	NSL-KDD	Multi-class	√	√	99.05 %	-	-	-
[32]	NSL-KDD	2-class	√	×	97.01 %	100%	98.48 %	97.01 %
[39]	UNSW-NB15	Multi-class	√	√	98.43 %	-	-	-
[40]	NSL-KDD	Multi-class	√	√	85.24 %	-	-	-
[33]	CIC-IDS 2018	Multi-class	√	×	95.14 %	-	-	-



Note:
i. x as an input vector.
ii. h as a vector representing the value of the hidden layer.
iii. y is a vector representing the value of the output layer.
iv. U is the weight matrix from the input layer to the hidden layer,
V is the weight matrix from the hidden layer to the output layer.
v. W is the weight of the last value of the hidden layer as the input for this time.

FIGURE 6. A typical Recurrent Neural Network model.

the input X gets processed at a different point, the RNN uses the same weight matrix, thus ensuring that every part benefits. By calling the output characteristics of the previously hidden layer and transferring them to the following input, the RNN can maintain an unwavering focus on and memory for sequential information. This mechanism makes RNNs a powerful tool for processing sequential data [42]. Furthermore, [43] validated an elementary RNN intrusion detection model.

However, the RNN is prone to gradient explosion and gradient vanishing during training, resulting in the gradient not being passed through the longer sequence during training, thus making the RNN unable to capture the effects of longer distances [41]. Because of the large time span, the network often cannot remember the information for such a long time,

and as the time span gets larger, it becomes increasingly difficult for the RNN to learn this information. We can solve the gradient explosion problem of RNN by setting the gradient threshold. But compared to gradient explosion, the gradient vanishing problem of RNN can appear trickier. We commonly use Long Short-Term Memory Network (LSTM) and Gated Recurrent Unit (GRU) to address the gradient vanishing problem, as they are more advanced variations.

The LSTM is specifically used to address the gradient vanishing in traditional RNNs. LSTM selectively remembers and forgets information by introducing gating mechanisms to capture better and convey long-term dependencies [44]. Figure 7 presents the LSTM unit.

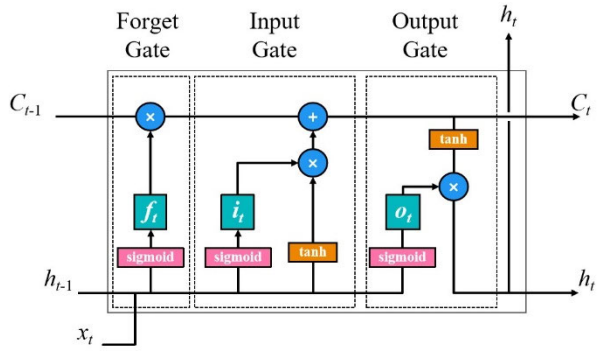


FIGURE 7. Structure of the LSTM unit.

Figure 7 illustrates that each LSTM memory cell has an ''input gate'', a ''forget gate'', and an ''output gate'', which work in concert. The ''output gate'' is an integral part of this coordinated operation. The ''input gate'' determines adding new information and updating the internal state. The ''forget gate'' determines whether the previous internal state should be discarded and regulates the degree of forgetting. The ''output gate'' determines the current output value and oversees the weighting of the output [30].

LSTMs have been applied to IDS in [45], [46], [47], [48], and [49]. The LSTM-based IDS has shown satisfactory results on various data sets. In contrast, in [27], [32], [50], and [51], the IDS using LSTM were compared with those using other models, and the performance of the IDS using LSTM alone was not as good as the performance of other combined models. To address the weak feature extraction capability of LSTM alone, [34], [35], [36], [37], [38], and [40] used a combination of CNN to promote feature extraction capability in IDS. In contrast, [52] took an alternative approach by using AE to reduce the dimensionality of the features and used it in combination with LSTM, which also achieved good intrusion detection results.

On the other hand, [39] used a weighted LSTM (WDLSTM), a variant of the LSTM, to prevent the overfitting of circular connections. In addition, [53], [54], [55], [56] have used bi-directional LSTM (BiLSTM) to overcome the limitation of only predicting the output of the following instant based on the temporal information of the previous instant.

The GRU is also an RNN, like LSTM. Figure 8 presents the GRU. The structure of GRU is different from LSTM in that it uses only two gates to adjust the flow of information. The ''update gate'' decides whether the internal state needs to be updated and controls the transfer of previous state information to the current state. Meanwhile, the ''reset gate'' determines how much the previous internal state affects the current time step [57]. In contrast, the GRU simplifies the gating mechanism while effectively managing information flow and state updates.

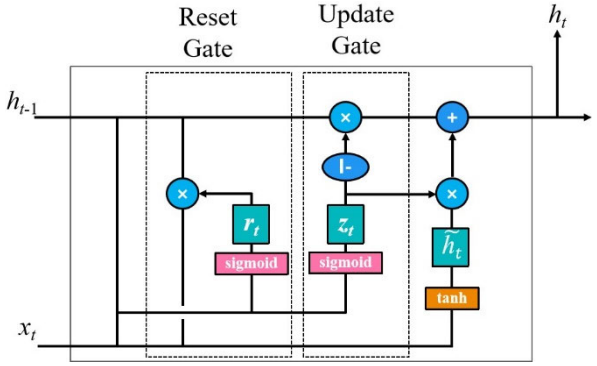


FIGURE 8. A typical GRU structure.

Both [32] and [58] propose a GRU-based IDS that achieves 89% and 50.25% accuracy on NSL-KDD datasets, respectively. The distinction between these two IDSs is that [32] considers the imbalance of the NSL-KDD dataset while [58] does not. Table 3 shows the details of the analyses.

Table 3 shows that the LSTM model is more frequently used than the other two deep learning models in IDSs. Due to the excellent feature extraction capability of the CNN model, it is often paired with the CNN model to form a hybrid IDS that performs well in intrusion detection. BiLSTM is also frequently used in IDSs and shows good intrusion detection capability.

D. AUTOENCODERS

The AE is an unsupervised machine learning algorithm that learns compact features or data representations. It has two major parts: the encoder and the decoder. The encoder maps the data into a low-dimensional representation, and the decoder remaps the low-dimensional representation into a reconstruction of the data [59]. Traditional AEs include standard AE, sparse AE, denoising AE, and other enhancement AE such as SAE.

The AE is mainly used for data dimensionality reduction and feature extraction [60], [61]. Therefore, AE and machine learning are often merged in IDS to create new deep learning models.AE is responsible for feature extraction and data dimensionality reduction, while machine learning oversees classification. For example, [60] proposed an IDS using a stacked asymmetric deep autoencoder (SNDAE) and random forest (RF). In [61], by combining sparse AE and logistic regression (LC), binary classification accuracy reached

TABLE 3. Comparison of RNN, LSTM and GRU in IDS.

Model	Datasets	Classification	Hidden layers	Result				Citation
				Accuracy	Precision	Recall	F1-score	
Simple RNN	NSL-KDD	Multi-class	3	74.19%	-	-	90.26%	[43]
	UNSW-NB15	Multi-class	3	85.38%	-	-	-	[27]
LSTM	CICIDS2017	2-class	4	99.01%	96.71%	98.58%	97.64%	[46]
	NSL-KDD	2-class	4	99.56%	99.52%	99.55%	-	[47]
	KF-ISAC	2-class	12	98.07%	97.06%	99.22%	98.13%	[34]
	AWID-GAN	-	5	93.53%	57.45%	31.87%	41.00%	[36]
CNN-LSTM	CIDDS-001	Multi-class	5	99.83%	99.00%	99.00%	99.00%	[37]
	UNSW-NB15	Multi-class	3	84.98%	-	95.96%	-	[38]
	CICIDS2017	Multi-class	3	99.91%	-	-	-	[40]
LSTM-AE	CICIDS2018	-	3	99.10%	99.07%	99.10%	99.02%	[52]
WDLSTM	UNSW-NB15	Multi-class	5	98.43%	-	-	-	[39]
	NSL-KDD Test$^+$	Multi-class	7	84.25%	-	-	-	[53]
BiLSTM	CICIDS2017	Multi-class	128	98.48%	100.00%	96.10%	98.20%	[54]
	UNSW-NB15	2-class	2	95.71%	100.00%	96.00%	98.00%	[55]
	NSL-KDD Test$^+$	2-class	4	94.26%	99.05%	90.79%	94.74%	[56]
GRU	NSL-KDD	2-class	5	50.25%	48.77%	99.88%	65.53%	[32]
	NSL-KDD	-	6	89.00%	-	-	-	[58]

87.2%. Reference [59] proposes an IDS that utilizes deep sparse AE and STL to recognize various categories of network attacks by pre-training the network and AEs to extract features from network traffic. Similarly, [51] and [62] combine SAE with support vector machines (SVM) applied in IDS and achieve good results.

The AE can also be combined with other deep learning techniques. For example, [63] developed an IDS with a denoising AE and a multilayer perceptron (MLP). On the other hand, [64] combined AE and CNN for intrusion detection. In another study, [52] proposed to combine LSTM and AE to achieve highly accurate classification results.

The SAE comprises numerous AE layers stacked on top of one another. The output of each AE layer is utilized as the input for the following layer, resulting in a deep neural network structure [65]. Stacked AE can learn higher-level feature representations through layer-by-layer training and pre-training, thus improving feature representation and data characterization [28]. Reference [60] used an SNADAE for feature extraction. Reference [66] uses deep SAE and applies them to IDS. References [28] and [29] both use an SAE for data dimensionality reduction and then use DNN to enhance the classification effects of the IDS. The main difference is that Tang et al. added an attention mechanism to SAE to achieve better classification performance [28]. A stacked contractive AE combined with an SVM is proposed [51]. Table 4 presents the details of the analysis.

As shown in Table 4, the AE is often used as a feature extraction method for IDS, and the commonly used ones are Sparse AE, Denoising AE, Stacked AE, and other AE variants, among which SAE is more widely used due to its excellent feature extraction performance by using stacked multi-layer AE. On the contrary, the classification performance of AE is relatively weak. It thus needs to be integrated with machine learning or deep learning techniques by extracting features (e.g., RF, LC, and MLP) to accomplish the classification task. Furthermore, we find that AE-based IDS rely heavily on the KDD Cup99 and the NSL-KDD datasets, where the latter outperforms the former.

E. DEEP BELIEF NETWORK

The DBN is a deep generative model composed of multilayer Restricted Boltzmann Machines (RBMs) [67]. The main goal of DBN is to learn the underlying distribution of the data and produce novel samples. Its distinguishing feature is the multilayer architecture, where each layer consists of an RBM. The RBM is a probabilistic model that employs an energy-based approach involving both visible and hidden layers to model the joint distribution of the data efficiently by adjusting the

TABLE 4. Comparison of the use of AE-based IDS for feature extraction and classifiers.

Datasets	Feature Extraction				Classifier						
	Sparse AE	Denoising AE	Stacked AE	Other AE	RF	LC	SVM	MLP	CNN	DNN	LSTM
KDD Cup99	×	×	√	√					[64]		
					[60]		[51]				
NSL-KDD	√	√	√	√		[62]				[28]	
UNSW-NB	×	√	√	×				[63]			
CIC-IDS	×	×	√	√						[29]	[52]
							[62]				
ISCX 2012	×	×	√	×							

Note: "√" means that the corresponding deep learning model is applied, and "×" means the opposite of "√".

weighting parameters. DBNs are trained by layer-by-layer pre-training and fine-tuning. They have various applications such as feature learning, data generation, migration learning and unsupervised pre-training. DBN can create data, scale down data and extract valuable feature representations with high performance and generalization capabilities [30].

A new hybrid weighting DBN (HW-DBN) was proposed by [67], and the model's effectiveness was tested on web and bot systems with 99.38% and 99.99% accuracy, respectively. Reference [68] used DBN for feature extraction on web data and used the back propagation (BP) neural network as a classifier, and ultimately, the model outperformed machine learning in detection and classification.

F. SELF-TAUGHT LEARNING

The STL is a semi-supervised learning technique designed to train models using a little piece of labelled data and many unlabeled data. The initial model in STL starts with labelled data for training. Then, it is used to predict unlabeled data and add those predictions to the training set with high confidence in the labels. Lastly, the model is retrained using the expanded training set. This process is iterated until the stopping condition is satisfied [69].

The STL was used to enhance the performance of sparse AE by connecting feature extraction based on STL to the authentic features of the dataset, which enables sparse AE to be trained on the combined features and show good generalization [59]. In the paper [69], the authors replaced the old STL deep learning model using a combination of sparse AE and Softmax classifiers with machine learning using stacked AE and SVM on the original STL framework. After the experimental results, the new model achieved 99.40% multi-classification accuracy on the NSL-KDD.

IV. DATASETS

In IDS, selecting appropriate datasets for intrusion detection assumes critical importance, as the quality and heterogeneity of the datasets directly impact the system's performance and resilience. This section explores the disparities between various intrusion detection datasets to gain comprehensive insight into their effect on experimental evaluation and algorithmic testing. By comparing the features and distributions of diverse datasets, researchers can gain valuable insights on selecting datasets appropriate for their specific research goals.

A. KDD CUP99

The KDD Cup99 dataset constitutes the most comprehensive and recognized public dataset within the field of intrusion. It originated from MIT Lincoln Laboratory and the US Department of Defense (DARPA) project in 1998 [25]. The experiments were collected to simulate network traffic generated by attacks under different complex network conditions in military networks, emulating a variety of different attack methods, a variety of user types, and a variety of different network traffic. The entire dataset contains approximately 7 million pieces of data, of which roughly 5 million are training data and 2 million are test data; each piece represents a single network behavior, which is determined to be expected by the data label.

The KDD Cup99 dataset classifies network behavior into normal behavior and abnormal behavior. The abnormal behaviour includes four types: Denial of Service Attacks (DoS), User-To-Root (U2R), Remote-to-Local (R2L) and Scanning Attacks (Probe). Network attacks are classified into four categories, each consisting of several types of attacks, as shown in Table 5.

The DoS attack is an attacker preventing regular computer network access. It targets the connectivity and broadband of

TABLE 5. Composition of KDD Cup99 and NSL-KDD datasets.

Category	10%KDD Cup99		NSL-KDD			
	Training Set	Testing Set	KDD Train+	KDD train+_20%	KDD Test+	KDD Test-21
Normal	97278	60593	67343	13449	9711	2152
DoS	391458	229853	45927	9234	7458	4342
R2L	1126	16189	995	209	995	2754
U2R	52	228	52	11	200	200
Probe	4107	4166	11656	2289	2421	2402
Total	494021	311029	125973	25192	22544	11850

a computer network, causing the relevant network service resources and hosts to be busy and unable to carry out their regular work.

Probe attacks are port attacks that scan a computer system for vulnerabilities or weaknesses in network services and use them to launch attacks on the system.

R2L is a kind of remote user attack; the attacker, through the remote control of the relevant host or network services, looks for the existence of vulnerabilities, usually for the target host, to obtain access to some services through remote means to carry out illegal operations, the primary behaviour is to log in the target host, destroy the regular operation of the system.

U2R attack is an attack on the permissions of the target host or network services; the attacker analyzes the vulnerability of the host, the attacker obtains the highest operating authority of the system through the vulnerability or weakness and performs illegal operations on the network services of the system, affecting the normal work of the system.

Each record in the KDDcup99 dataset contains 41 different features, divided into basic features numbered 1 to 9, content features numbered 10 to 22, and traffic features numbered 23-41. The last feature defines the network data as normal or abnormal. Figure 9 illustrates the KDD Cup99 dataset record.

```
0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,112,18,0.00,0.00,1.00,1.00,0.16,0.07,0
.00,255,18,0.07,0.07,0.00,0.00,0.00,0.00,1.00,
1.00,neptune.

0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,121,11,0.00,0.00,1.00,1.00,0.09,0.07,0
.00,255,11,0.04,0.07,0.00,0.00,0.00,0.00,1.00,
1.00,neptune.

0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,130,20,0.00,0.00,1.00,1.00,0.15,0.06,0
.00,255,20,0.08,0.06,0.00,0.00,0.00,0.00,1.00,
1.00,neptune.

0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,89,8,0.00,0.00,1.00,1.00,0.09,0.07,0.0
0,255,8,0.03,0.06,0.00,0.00,0.00,0.00,1.00,1.0
0,neptune.
```

FIGURE 9. KDD Cup99 dataset record.

Figure 9 shows that 38 of the 41 features are numeric, and three are symbolic. Each data record consists of a category label and 41 dimensions of features. Of these, dimension 2 is a protocol-type symbolic feature, dimension 3 is a network service-type symbolic feature, dimension 4 is a network connection status symbolic feature, and the last dimension is a label that serves as a basis for determining whether the record is normal or abnormal. The rest are numeric features [24]. Table 5 demonstrates the composition of the KDD Cup99.

B. NSL-KDD

The NSL-KDD is an improved and adapted revision of the KDD Cup99 [70]. Unlike the original dataset, the NSL-KDD does not contain redundant data and retains the same data characteristics. The amount of data was reduced by removing connection records numbered 136489 and 136497 from the test set [25], [71]. The NSL-KDD consists of four subdatasets: KDD Train+, KDD train+_20%, KDD Test+ and KDD Test-21 [53]. Table 5 demonstrates the composition of the NSL-KDD.

C. UNSW-NB15

The Australian Cyber Security Centre's Cyber Scope Lab's IXIA PerfectStorm tool generated the raw network packets for the UNSW-NB 15 [63].

The UNSW-NB15 compiles network traffic data, including cyber-attacks and general network traffic. The dataset consists of network traffic records accumulated between 2015 and 2016, including protocols such as TCP, UDP, ICMP, and HTTP. The UNSW-NB15 consists of a training set containing 175,341 records and a testing set containing 82,332 records [55]. The dataset captures a variety of network attacks, including DoS attacks, worms, analyses, backdoors and nine other attacks [25]. In addition, the UNSW-NB15 contains 254,044 instances and 49 features [39]. Table 6 presents the composition of the dataset.

D. CICIDS2021

The CICIDS2017 dataset uses the CICFlowMeter tool to extract over 80 feature attributes from the raw data [71]. There are two methods for extracting features: online and offline modes. The online mode monitors network traffic in real-time, generates features, and saves the feature attributes locally in CSV format when listening is complete. The offline mode is to submit an entire packet in .pcap format to the CICFlowMeter tool, resulting in a CSV file containing the features. The categories of attacks identified in the CICIDS2017 dataset are Botnet attacks, Brute Force attacks, DoS & DDoS attacks, Infiltration attacks, Web attacks, and Port Scan attacks [72]. Table 7 presents the composition of the CICIDS2017.

This paper analyses the use of datasets commonly used in recent years about IDS, intending to provide reference and guidance for selecting subsequent datasets. Table 8 displays that the primary datasets used in IDSs utilizing DNN and

TABLE 6. Composition of UNSW-NB15 dataset.

Attack Types	Description	Training Set	Testing Set
Normal	Normal data	56000	37000
DoS	The attacker consumes the target system's resources, preventing it from providing normal services to legitimate users.	12264	4089
Worms	Worms do not need to rely on a host file to spread, but instead, use network or system vulnerabilities to replicate and spread automatically.	130	44
Analysis	A method of attack in which sensitive information, vulnerabilities or weaknesses are obtained through analysis of the target system.	2000	677
Exploits	A form of attack that exploits a vulnerability, weakness, or error in a computer system to perform a malicious operation.	33393	11132
Fuzzers	An attack based on fuzz testing	18184	6062
Reconnaissance	An attacker's activity of information gathering and reconnaissance of a target system or network.	10491	3496
Backdoors	A covert access mechanism embedded in a computer system is used to bypass normal authentication to gain unauthorized access.	1746	583
Generic	Attackers exploit common vulnerabilities or techniques	40000	18871
Shell code	A code that is executed under the control of an attacker, usually to exploit a weakness in a system to gain illegal access	1133	378
Total		175341	82332

CNN are the NSL-KDD and UNSW-NB15. In the RNN-based IDS, the NSL-KDD is the primary dataset for intrusion detection. In addition, for IDS with AE, the KDD Cup99 and NSL-KDD are the primary datasets used for IDS training and testing. In conclusion, the NSL-KDD dataset is the widely applied and researched dataset in IDS.

V. COMPARISON ON APPROACHES IN DEEP LEARNING-BASED IDS

This section discusses several aspects of deep learning in terms of data preprocessing, feature extraction, and classifiers, and it will help to provide further insights into the processes and characteristics of deep learning models operating in IDSs.

TABLE 7. Composition of CICIDS2017 dataset.

Class	Attack Types	Total
Normal	Benign	2359087
Botnet Attack	Botnet	1966
Web Attack	SQL Injection, Brute Force, XSS	2180
DoS&DDoS	DoS goldeneye, DoS slow loris, DoS http test, DoS hulk, DDoS, Heartbleed	294506
Infiltration Attack	Infiltration	36
Brute Force Attack	SSH patator FTP patator	13835
Port Scan	Port Scan	158930

TABLE 8. Most commonly used datasets in IDS based on various deep learning models.

Model	Datasets				
	KDD Cup99	NSL-KDD	UNSW-NB15	CICIDS2017	Others
DNN	[25]	[24] [26]	[25] [27]	[25]	
CNN		[32] [38] [40]	[31] [37] [38] [39]	[34] [40]	[33] [35] [36]
RNN	[51]	[32] [43] [47] [51] [53] [56] [58] [73]	[27] [43] [55]	[46] [52] [54]	[45] [48] [49] [52]
AE	[51] [60] [64] [66]	[28] [50] [51] [59] [60] [61]	[63] [66]	[29] [62]	[62]
DBN	[68]			[67]	
STL		[59] [69]			

A. DATA PREPROCESSING

Data preprocessing is the cornerstone of the intrusion detection process in IDSs because processed data enables the system to reach optimal performance faster. During the data preprocessing phase, non-numeric attributes in the dataset must first be converted into numeric attributes using numeric coding methods. Normalization is then performed by scaling the features in the range [0, 1].

Most machine learning and deep learning frameworks are constructed through mathematical operations and typically accept numerical inputs and parameters. Non-numerical features (e.g., text, category labels) cannot be directly used for these mathematical operations in their raw form. Therefore, it is necessary to transform non-numerical features into numerical form to ensure model compatibility [74]. Standard numerical coding methods in IDSs based on deep learning models are one-hot and label encoding. Table 9 shows the details.

TABLE 9. Data preprocessing Methods for deep learning-based IDS.

Datasets	Numerical Encoding	Normalization	Reference
KDD Cup99	One-hot Encoding	√	[64] [66]
NSL-KDD			[24] [28] [38] [47] [53] [61] [69] [73]
UNSW-NB15			[31] [55] [63]
CICIDS2017			[26] [46]
Others			[35] [45] [49]
CIDDS-001/UNSW-NB15	Label encoding	√	[37]
NLS-KDD/CIIDS-001/UNSW-NB15			[76]
NSL-KDD/CICIDS2017	Label one hot encoding	√	[40]
KF-ISAC/CSIC-2010/ CICIDS2017	UTF-8 Character Encoding	√	[34]
NSL-KDD	LeaveOneOut encoding	√	[50]
X-IIoTID	Ordinal Encoding	√	[77]
IoT-23			[78]

Note: " √ " means that the corresponding step is included.

After digital coding, it is necessary to normalize the data to standardize its distribution. The Min-Max normalization technique scales the data to a range of [0,1] while maintaining the linear relationship of the original data [75]. The most used normalization method is Min-Max scaling, mathematically defined in Equation 1 below.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

where x is the original feature value of the attribute, $\min(x)$ is the minimum feature value, $\max(x)$ is the maximum feature value, and x' is the normalized feature value of the attribute. Each feature value is transformed to fit the range [0, 1]. Normalizing the data increases model convergence speed and accuracy, as well as preventing gradient explosion issues.

Table 9 shows that several studies on IDS using deep learning techniques have highlighted the importance of numeric coding to convert non-numeric features into numeric features and the normalization process. The most used data preprocessing techniques are One-hot encoding and Min-Max normalization. One-hot encoding is a commonly used numerical coding method in the data preprocessing phase of deep learning-based IDS. This procedure can be implemented intuitively, and each category is converted into a separate binary feature. This approach is easy to understand and interpret. In addition, single encoding effectively eliminates any ordered relationships between different types, making it distinct from other methods, such as Label encoding.

B. DATA IMBALANCE

In 2-class, the skewed nature of the data leads to classification problems due to data imbalance. There are two types of data imbalance: inter-class imbalance and intra-class imbalance. Inter-class imbalance means a significant difference in the sample size between different classes in a dataset. In contrast, intra-class imbalance is defined as multiple subclasses within a particular class with significant differences in the sample size [79]. In data mining, classification algorithms are designed on the premise that the number of categories for each sample is balanced, whereas, in practice, the data is often unbalanced. In the case of imbalanced multi-class datasets, traditional classification methods tend to favor the majority class. The minority classes provide limited information to the classifier, resulting in a higher probability of misclassifying the samples from the minority classes. In many real-world scenarios, minority class data carries crucial information [80].

Strategies for addressing imbalanced datasets entail modifying the sample distribution through data-level interventions, such as applying specific algorithms to increase or reduce the number of samples. Resampling the dataset is pivotal to achieving class balance in this approach [81]. This strategy typically involves either oversampling or undersampling. The notion behind oversampling is to augment the number of minority category samples algorithmically. Representative oversampling techniques include the Synthetic Minority Over-sampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN).

In contrast, the undersampling method aims to balance the sample sizes in every dataset category by removing multiple samples in some classes. Tomek-Links is an example of a typical undersampling algorithm [82]. This section compares the similarities and differences in data imbalance solutions regarding deep learning-based IDS in recent years. Table 10 shows the details.

Table 10 shows that the SMOTE algorithm is frequently employed for solving the data imbalance. Unlike basic oversampling methods such as the random oversampling algorithm, the SMOTE algorithm enhances the inclusiveness of the minority category by making new samples. It helps the

TABLE 10. Data imbalance solutions for deep learning-based IDS.

Datasets	Dataset Imbalance Solutions	Classification	Accuracy (No Solutions)	Accuracy	Citation
CICIDS2018	Stratified K-Fold Cross-Validation Strategy	2-class	/	99.70%	[35]
NSL-KDD	Stratified K-Fold Cross-Validation Strategy	2-class	99.09%(k=2)	99.36%(k=10)	[38]
NSL-KDD	SMOTE	2-class	/	99.56%	[47][66]
NSL-KDD Test+	SMOTE	Multi-class	82.24%	83.57%	[73]
KDD Cup99	SMOTE	Multi-class	98.00%	99.996%	[66]
CIDDS-001	SMOTE and Tomek-Links	Multi-class	/	99.83%	[37]
NSL-KDD	ADASYN	Multi-class	/	85.24%	[40]
Bot-IoT	Focal Loss Function	Multi-class	69.63%(CNN)	86.77%(CNN)	[83]

Note: "/" means unknown or none.

model to understand the underlying patterns of the minority class throughout the training process rather than just replicating the current data. Due to severe class imbalance issues, intrusion detection datasets like the NSL-KDD can lead to false positives and inaccurate detection rates. This issue can negatively impact performance evaluation, and therefore, it is essential to address the data imbalance issue in the NSL-KDD dataset. It is worth noting that only a few references in Table 10 compare the accuracy rates before and after addressing the data imbalance problem. It is recommended that at least one controlled experiment be conducted in future research to illustrate the impact and effectiveness of using the data imbalance resolution technique versus not using it.

C. FEATURE EXTRACTION AND CLASSIFIERS

This section explores the differences between various IDSs regarding feature extraction techniques and classifiers. The feature extraction process is crucial, whether machine learning or deep learning. It transforms unprocessed data into important feature sets that significantly improve the model's performance in subsequent training and prediction phases.

Standard techniques for extracting features in deep learning include CNN, AE, and DBN. CNN can capture local features in data by performing convolutional operations, which is particularly useful for processing network traffic data [84]. In contrast, AE is an unsupervised learning method that extracts valuable feature representations from unlabeled data. It contributes to dimensionality reduction, noise filtering, and deep feature learning [85]. On the contrary, DBN is like AE, which can acquire favorable features from unlabeled data while accomplishing data reduction and deep feature learning [86].

In data mining, classification is a necessary method. The basic idea is to obtain classification functions or models called classifiers based on accessible data. The model maps the data records in the database into predefined categories for forecasting. Such as in [87] and [88], where it is shown that RF is a reasonable classification algorithm for IDS. Moreover, in [89], it is argued that SVM outperforms other classification algorithms in comparison to SVM.

On the other hand, [68] used BP neural networks as classifiers for IDS. Also, Softmax functions are often used as classifiers for IDS in deep learning. Table 11 demonstrates the details of analysis.

TABLE 11. Feature extraction methods and classifiers for deep learning-based IDS.

Datasets	Feature Extraction	Classifier	Classification	Citation
UNSW-NB15	CNN	Softmax	Multi-class	[31]
CICIDS2018				[35]
NSL-KDD/CICIDS2017				[40]
NSL-KDD				[53]
NSL-KDD		Softmax	Multi-class	[28]
KDD Cup99/ UNSW-NB15				[66]
KDD Cup99/ NSL-KDD	AE	SVM	Multi-class	[51]
ISCX 2012/ CICIDS2017			Multi-class/2-class	[62]
NSL-KDD			Multi-class/2-class	[69]
KDD Cup99/ NSL-KDD		RF	Multi-class	[60]
NSL-KDD		LC	2-class	[61]
UNSW-NB15		MLP	2-class	[63]
KDD Cup99	DBN	BP Neural Network	2-class	[68]

In various studies (e.g., [31] [35], [40], [53]), CNN models are employed to extract features from the data. The CNN models are built as a hierarchical structure that consists of multiple convolution and pooling layers. The lower layers acquire simple and local features, whereas the higher layers

TABLE 12. Comparative analysis of the current status and future development trends of IDS development in the IoT.

Model	Dataset	Data preprocessing	Innovation	Citation
FNN-Focal CNN-Focal	Bot-IoT IIoT-2021 EHMS-2020	Focal loss function	Demonstrating the Effective Training of DL-Based Models Using Focused Loss Functions to Overcome Unbalanced Data in IoT Datasets.	[83]
LSTM(2-class) ANN+LSTM+CNN(Multi-class)	IoT-23	Ordinal Encoding /Normalization	Incorporating Lambda architecture in deep learning models can increase data processing power and enhance the real-time performance of the model.	[78]
ANN/LSTM/GRU	-	One-hot Encoding /Normalization	Propose an AI technology-based model to detect IoT users and classify blockchain based smart contracts using DL model.	[77]
EIDM (Convolutional+ Dense layers)	CICIDS2017	One-hot Encoding /Normalization /SMOTE	All 15 classes in the CICIDS2017 dataset were realized for classification purposes, rather than grouping by similar features to achieve classification.	[90]
MM-WMVEDL (BiLSTM+ELM+GRU)	IoT-23 UNSW-NB15	Harris Hawk optimization-based elite fractional derivative mutation (HHO-EFDM)	A multi-modal architecture is used to deal with complex relationships in network traffic data, as well as the use of wavelet-based feature extraction methods to enhance the recognition capabilities of the model.	[91]
CNN	KDDCup-99 NSL-KDD BoT-IoT CICIDS2017	MGO	A new feature selection algorithm MGO using Whale Optimization Algorithm (WOA) modified Growth Optimizer (GO) is proposed.	[92]
Simple RNN/BiLSTM	BoT-IoT	Group Method of Data Handling (GMDH) /Mutual Information (MI) /Chi-Square Statistic	A fog-cloud based IoT IDS is proposed to process the dataset using different feature selection algorithms.	[93]
CNN-CapSA	KDDCup-99 NSL-KDD BoT-IoT CICIDS201	Capuchin Search Algorithm (CapSA)	An improved version of CapSA feature selection algorithm is proposed to improve intrusion detection in IoT environment.	[94]
BHS-ALOHDL (CNN/LSTM)	ToN-IoT CICIDS-2017	Ant Lion Optimizer (ALO)	A BHS-ALOHDL model based on ALO feature selection algorithm and Flower pollination algorithm (FPA) superparameter optimization algorithm with CNN-LSTM fusion is proposed.	[95]
WILS-TRS (LSTM)	NLS-KDD CIIDS-001 UNSW-NB15	Label Encoding	Optimizing the weights of the LSTM model using WOA reduces time complexity, speeds up convergence and improves intrusion detection.	[76]

effectively merge these essential characteristics to recognize intricate and abstract patterns. This hierarchical mechanism for feature extraction enables convolutional neural networks to learn feature representations from lower to higher levels automatically. AE map the input data to a low-dimensional latent space for feature extraction. AE combined with Softmax classifiers, were used by [28] and [66] in forming an IDS for multi-classification. However, [51], [62], and [69] used AE to extract features and SVM to achieve classification for intrusion detection. Using DBN to extract features, as outlined in [68], is a suitable approach. This is due to DBN's ability to learn a layered and abstract representation of the input data through its deep structure, layer-by-layer pre-training, and generative modelling properties.

Softmax is a popular classifier in deep learning-based IDS, capable of mapping network traffic data into probability distributions for different categories and, thus, being well-suited for multi-category classification tasks. It is frequently used due to its capability to be trained end-to-end by deep learning models like CNN. While Softmax classifiers are commonly used in IDS, it is worth considering other classifiers (such as RF [60], LC [61], MLP [63], etc.) that may prove equally effective depending on the specific intrusion detection task and dataset characteristics. The selection of classifiers should be rationalized and assessed based on specific scenarios and performance requirements.

D. THE STATUS AND TRENDS OF IDS DEVELOPMENT IN THE IOT

This section discusses the current state of IDS development and future research trends in the IoT environment by examining relevant literature and research from the past two years. Such examination bears considerable theoretical and practical significance in improving network security defense capability and ensuring a more secure and dependable IoT environment. Table 12 reveals specific analysis outcomes.

From Table 12, deep learning models applied to IDS in IoT are mainly moving towards integrating hybrid deep learning models. For example, [78] proposed a hybrid IDS combining ANN, LSTM and CNN. In [91], a deep learning-based IoT IDS called MM-WMVEDL was designed, where the model includes BiLSTM, ELM and GRU, achieving the ability to process complex network traffic data in a multimodal

structure. Similarly, in [95], an IDS with excellent intrusion detection performance is built using CNN and LSTM models combined with feature selection and superparameter optimization algorithms. Meanwhile, [90] proposed a deep learning model for IoT IDS consisting of only convolutional and dense layers. Others have applied a focal loss function to IDS to address the data imbalance problem [93]. In the future, we may see more research on how to effectively integrate multiple deep learning models, and future deep learning models may focus more on the ability of multimodal and cross-modal learning to understand better and utilize complex data. Furthermore, the use of datasets in IoT environments highlights the specialization, and there will be more datasets used specifically for IoT environments such as Bot-IoT [83], [92], [93], [94], IoT-23 [78], [91], ToN-IoT [95], and so on.

The research in IoT also focuses on the feature selection of data [91] and uses HHO-EFDM, a wavelet-based feature selection algorithm, to enhance deep learning models for intrusion detection. Reference [92] proposed an MGO feature selection algorithm combining WOA and GO applied in cloud and IoT environments. In [93], the authors compare the performance of three feature selection algorithms, GMDH, MI and Chi-Square Statistic and propose a fog-cloud-based IDS for IoT. Reference [94] combines the CapSA feature selection technique with CNN to design an IDS for cloud and IoT environments.

Recently, the conjunction of blockchain technology and the IoT has sparked increased interest among researchers. This synergy presents significant security challenges that have drawn attention from experts. Intrusion detection technology has emerged as a promising solution for securing blockchain IoT systems, as evidenced by researchers' adoption of it. Reference [77] proposed a deep learning IDS model for malicious users and smart contracts in blockchain-based IoT. The study evaluated the model's performance and effectiveness using the corresponding malicious user detection dataset X-IIoTID and the intelligent contract detection dataset. Meanwhile, [95] employed the ALO feature selection algorithm and FPA superparameter optimization algorithm, fused with the CNN-LSTM model to create an IDS for a blockchain-assisted IoT healthcare system.

VI. CONCLUSION

This paper provided a detailed analysis of current deep learning methods used for IoT security, covering prevalent algorithms and architectures, intrusion detection datasets, data preprocessing and feature extraction techniques, and various classifiers. This research addressed the use of deep learning in detecting anomalies and analyzing behavioral patterns, which has demonstrated the potential to improve detection accuracy and reduce false alarms and the data imbalance issue in intrusion detection datasets. We also provided a detailed comparative analysis of various solutions to this problem. The primary importance of this study's outcomes is to recapitulate the present research on deep learning in intrusion detection for IoT, providing meaningful perspectives for IoT security analysts and practitioners on selecting appropriate deep learning models, datasets, numerical encoding methods, and strategies to tackle data imbalance.

ACKNOWLEDGMENT

The authors would like to thank to the Universiti Kebangsaan Malaysia for supporting this work under DIP-2022-021.

REFERENCES

[1] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A review of intrusion detection systems using machine and deep learning in Internet of Things: Challenges, solutions and future directions," *Electronics*, vol. 9, no. 7, p. 1177, Jul. 2020, doi: 10.3390/electronics9071177.

[2] G. Thamilarasu and S. Chawla, "Towards deep-learning-driven intrusion detection for the Internet of Things," *Sensors*, vol. 19, no. 9, p. 1977, Apr. 2019, doi: 10.3390/s19091977.

[3] I. Idrissi, O. Moussaoui, and M. Azizi, "A lightweight optimized deep learning-based host-intrusion detection system deployed on the edge for IoT," *Int. J. Comput. Digit. Syst.*, vol. 11, no. 1, pp. 209–216, Jan. 2022, doi: 10.12785/ijcds/110117.

[4] P. L. S. Jayalaxmi, R. Saha, G. Kumar, M. Conti, and T.-H. Kim, "Machine and deep learning solutions for intrusion detection and prevention in IoTs: A survey," *IEEE Access*, vol. 10, pp. 121173–121192, 2022, doi: 10.1109/ACCESS.2022.3220622.

[5] A. R. Khan, M. Kashif, R. H. Jhaveri, R. Raut, T. Saba, and S. A. Bahaj, "Deep learning for intrusion detection and security of Internet of Things (IoT): Current analysis, challenges, and possible solutions," *Secur. Commun. Netw.*, vol. 2022, pp. 1–13, Jul. 2022, doi: 10.1155/2022/4016073.

[6] A. Thakkar and R. Lohiya, "A review on machine learning and deep learning perspectives of IDS for IoT: Recent updates, security issues, and challenges," *Arch. Comput. Methods Eng.*, vol. 28, no. 4, pp. 3211–3243, Jun. 2021, doi: 10.1007/s11831-020-09496-0.

[7] M. Zhong, Y. Zhou, and G. Chen, "Sequential model based intrusion detection system for IoT servers using deep learning methods," *Sensors*, vol. 21, no. 4, p. 1113, Feb. 2021, doi: 10.3390/s21041113.

[8] J. Lansky, S. Ali, M. Mohammadi, M. K. Majeed, S. H. T. Karim, S. Rashidi, M. Hosseinzadeh, and A. M. Rahmani, "Deep learning-based intrusion detection systems: A systematic review," *IEEE Access*, vol. 9, pp. 101574–101599, 2021, doi: 10.1109/ACCESS.2021.3097247.

[9] M. S. Al-Daweri, K. A. Z. Ariffin, S. Abdullah, and M. F. E. M. Senan, "An analysis of the KDD99 and UNSW-NB15 datasets for the intrusion detection system," *Symmetry*, vol. 12, no. 10, p. 1666, Oct. 2020, doi: 10.3390/sym12101666.

[10] M. S. Al-Daweri, S. Abdullah, and K. A. Z. Ariffin, "A homogeneous ensemble based dynamic artificial neural network for solving the intrusion detection problem," *Int. J. Crit. Infrastruct. Protection*, vol. 34, Sep. 2021, Art. no. 100449, doi: 10.1016/j.ijcip.2021.100449.

[11] M. S. Al-Daweri, S. Abdullah, and K. A. Z. Ariffin, "An adaptive method and a new dataset, UKM-IDS20, for the network intrusion detection system," *Comput. Commun.*, vol. 180, pp. 57–76, Dec. 2021, doi: 10.1016/j.comcom.2021.09.007.

[12] A. Lamjid, K. A. Z. Ariffin, M. J. A. Aziz, and N. S. Sani, "Determine the optimal hidden layers and neurons in the generative adversarial networks topology for the intrusion detection systems," in *Proc. Int. Conf. Cyber Resilience (ICCR)*, Oct. 2022, pp. 1–7.

[13] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, Dec. 2019, doi: 10.1186/s42400-019-0038-7.

[14] C. Vij and H. Saini, "Intrusion detection systems: Conceptual study and review," in *Proc. 6th Int. Conf. Signal Process., Comput. Control (ISPCC)*, Oct. 2021, pp. 694–700.

[15] W. Liang, L. Xiao, K. Zhang, M. Tang, D. He, and K.-C. Li, "Data fusion approach for collaborative anomaly intrusion detection in blockchain-based systems," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14741–14751, Aug. 2022, doi: 10.1109/JIOT.2021.3053842.

[16] T. T. Khoei, G. Aissou, W. C. Hu, and N. Kaabouch, "Ensemble learning methods for anomaly intrusion detection system in smart grid," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, May 2021, pp. 129–135.

[17] A. A. Z. Khan, "Misuse intrusion detection using machine learning for gas pipeline SCADA networks," in *Proc. Int. Conf. Secur. Manag. (SAM), Steering Committee World Congr. Comput. Sci., Comput.*, 2019, pp. 84–90.

[18] D. Papamartzivanos, F. Gómez Mármol, and G. Kambourakis, "Introducing deep learning self-adaptive misuse network intrusion detection systems," *IEEE Access*, vol. 7, pp. 13546–13560, 2019, doi: 10.1109/ACCESS.2019.2893871.

[19] W. Alhakami, A. Alharbi, S. Bourouis, R. Alroobaea, and N. Bouguila, "Network anomaly intrusion detection using a nonparametric Bayesian approach and feature selection," *IEEE Access*, vol. 7, pp. 52181–52190, 2019, doi: 10.1109/ACCESS.2019.2912115.

[20] R. Gassais, N. Ezzati-Jivan, J. M. Fernandez, D. Aloise, and M. R. Dagenais, "Multi-level host-based intrusion detection system for Internet of Things," *J. Cloud Comput.*, vol. 9, no. 1, pp. 1–16, Dec. 2020.

[21] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Comput. Secur.*, vol. 86, pp. 147–167, Sep. 2019, doi: 10.1016/j.cose.2019.06.005.

[22] P. Devan and N. Khare, "An efficient XGBoost–DNN-based classification model for network intrusion detection system," *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12499–12514, Aug. 2020, doi: 10.1007/s00521-020-04708-x.

[23] B. Lee, S. Amaresh, C. Green, and D. Engels, "Comparative study of deep learning models for network intrusion detection," *SMU Data Sci. Rev.*, vol. 1, no. 1, p. 8, 2018.

[24] Z. Liu, M. U. D. Ghulam, Y. Zhu, X. Yan, L. Wang, Z. Jiang, and J. Luo, "Deep learning approach for IDS: Using DNN for network anomaly detection," in *Fourth International Congress on Information and Communication Technology* (Advances in Intelligent Systems and Computing), vol. 1041, X.-S. Yang, S. Sherratt, N. Dey, and A. Joshi, Eds. Singapore: Springer, 2020, pp. 471–479, doi: 10.1007/978-981-15-0637-6_40.

[25] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[26] A. Thakkar and R. Lohiya, "Fusion of statistical importance for feature selection in deep neural network-based intrusion detection system," *Inf. Fusion*, vol. 90, pp. 353–363, Feb. 2023, doi: 10.1016/j.inffus.2022.09.026.

[27] A. M. Aleesa, M. Younis, A. A. Mohammed, and N. M. Sahar, "Deep-intrusion detection system with enhanced UNSW-NB15 dataset based on deep learning techniques," *J. Eng. Sci. Technol.*, vol. 16, no. 1, pp. 711–727, 2021.

[28] C. Tang, N. Luktarhan, and Y. Zhao, "SAAE-DNN: Deep learning method on intrusion detection," *Symmetry*, vol. 12, no. 10, p. 1695, Oct. 2020, doi: 10.3390/sym12101695.

[29] H. Mennour and S. Mostefai, "A hybrid deep learning strategy for an anomaly based N-IDS," in *Proc. Int. Conf. Intell. Syst. Comput. Vis. (ISCV)*, Fez, Morocco, Jun. 2020, pp. 1–6, doi: 10.1109/ISCV49265.2020.9204227.

[30] I. H. Sarker, "Deep cybersecurity: A comprehensive overview from neural network and deep learning perspective," *Social Netw. Comput. Sci.*, vol. 2, no. 3, p. 154, May 2021, doi: 10.1007/s42979-021-00535-6.

[31] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Proc. Comput. Sci.*, vol. 185, pp. 239–247, Jan. 2021, doi: 10.1016/j.procs.2021.05.025.

[32] S. Al-Emadi, A. Al-Mohannadi, and F. Al-Senaid, "Using deep learning techniques for network intrusion detection," in *Proc. IEEE Int. Conf. Informat., IoT, Enabling Technol. (ICIoT)*, Doha, Qatar, Feb. 2020, pp. 171–176, doi: 10.1109/ICIoT48696.2020.9089524.

[33] J. Kim, Y. Shin, and E. Choi, "An intrusion detection model based on a convolutional neural network," *J. Multimedia Inf. Syst.*, vol. 6, no. 4, pp. 165–172, Dec. 2019, doi: 10.33851/JMIS.2019.6.4.165.

[34] A. Kim, M. Park, and D. H. Lee, "AI-IDS: Application of deep learning to real-time web intrusion detection," *IEEE Access*, vol. 8, pp. 70245–70261, 2020, doi: 10.1109/ACCESS.2020.2986882.

[35] A. Dey, "Deep IDS: A deep learning approach for intrusion detection based on IDS 2018," in *Proc. 2nd Int. Conf. Sustain. Technol. Ind. 4.0 (STI)*, Dhaka, Bangladesh, Dec. 2020, pp. 1–5, doi: 10.1109/STI50764.2020.9350411.

[36] M. Asaduzzaman and M. M. Rahman, "An adversarial approach for intrusion detection using hybrid deep learning model," in *Proc. Int. Conf. Inf. Technol. Res. Innov. (ICITRI)*, Jakarta, Indonesia, Nov. 2022, pp. 18–23, doi: 10.1109/ICITRI56423.2022.9970221.

[37] S. Al and M. Dener, "STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment," *Comput. Secur.*, vol. 110, Nov. 2021, Art. no. 102435, doi: 10.1016/j.cose.2021.102435.

[38] P. Wu and H. Guo, "LuNet: A deep neural network for network intrusion detection," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Xiamen, China, Dec. 2019, pp. 617–624, doi: 10.1109/SSCI44817.2019.9003126.

[39] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Inf. Sci.*, vol. 513, pp. 386–396, Mar. 2020, doi: 10.1016/j.ins.2019.10.069.

[40] C. Liu, Z. Gu, and J. Wang, "A hybrid intrusion detection system based on scalable K-means+random forest and deep learning," *IEEE Access*, vol. 9, pp. 75729–75740, 2021, doi: 10.1109/ACCESS.2021.3082147.

[41] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Appl. Sci.*, vol. 9, no. 20, p. 4396, Oct. 2019, doi: 10.3390/app9204396.

[42] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018, doi: 10.1109/ACCESS.2018.2836950.

[43] S. M. Kasongo, "A deep learning technique for intrusion detection system using a recurrent neural networks based framework," *Comput. Commun.*, vol. 199, pp. 113–125, Feb. 2023, doi: 10.1016/j.comcom.2022.12.010.

[44] F. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "Intrusion detection systems using long short-term memory (LSTM)," *J. Big Data*, vol. 8, no. 1, p. 65, May 2021, doi: 10.1186/s40537-021-00448-4.

[45] S. Amutha, R. Kavitha, R. Srinivasan, and Kavitha, "Secure network intrusion detection system using NID-RNN based deep learning," in *Proc. Int. Conf. Adv. Comput., Commun. Appl. Informat. (ACCAI)*, Chennai, India, Jan. 2022, pp. 1–5, doi: 10.1109/ACCAI53970.2022.9752526.

[46] J. Figueiredo, C. Serrão, and A. M. de Almeida, "Deep learning model transposition for network intrusion detection systems," *Electronics*, vol. 12, no. 2, p. 293, Jan. 2023, doi: 10.3390/electronics12020293.

[47] V. Rajasekar, S. Sarika, S. Velliangiri, and K. S. Kalaivani, "An efficient intrusion detection model based on recurrent neural network," in *Proc. IEEE Int. Conf. Distrib. Comput. Electr. Circuits Electron. (ICDCECE)*, Ballari, India, Apr. 2022, pp. 1–6, doi: 10.1109/ICDCECE53908.2022.9793016.

[48] S. Althubiti, W. Nick, J. Mason, X. Yuan, and A. Esterline, "Applying long short-term memory recurrent neural network for intrusion detection," in *Proc. SoutheastCon*, St. Petersburg, FL, USA, Apr. 2018, pp. 1–5, doi: 10.1109/SECON.2018.8478898.

[49] M. Shurman, R. Khrais, and A. Yateem, "DoS and DDoS attack detection using deep learning and IDS," *Int. Arab J. Inf. Technol.*, vol. 17, no. 4, pp. 655–661, Jul. 2020, doi: 10.34028/iajit/17/4A/10.

[50] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018, doi: 10.1109/ACCESS.2018.2863036.

[51] P. Jisna, T. Jarin, and P. N. Praveen, "Advanced intrusion detection using deep learning-LSTM network on cloud environment," in *Proc. 4th Int. Conf. Microelectron., Signals Syst. (ICMSS)*, Kollam, India, Nov. 2021, pp. 1–6, doi: 10.1109/ICMSS53060.2021.9673607.

[52] V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa-Kim, "A novel two-stage deep learning model for network intrusion detection: LSTM-AE," *IEEE Access*, vol. 11, pp. 37131–37148, 2023, doi: 10.1109/ACCESS.2023.3266979.

[53] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset," *IEEE Access*, vol. 8, pp. 29575–29585, 2020, doi: 10.1109/ACCESS.2020.2972627.

[54] S. Sivamohan, S. S. Sridhar, and S. Krishnaveni, "An effective recurrent neural network (RNN) based intrusion detection via bi-directional long short-term memory," in *Proc. Int. Conf. Intell. Technol. (CONIT)*, Hubli, India, Jun. 2021, pp. 1–5, doi: 10.1109/CONIT51480.2021.9498552.

[55] B. Roy and H. Cheung, "A deep learning approach for intrusion detection in Internet of Things using bi-directional long short-term memory recurrent neural network," in *Proc. 28th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2018, pp. 1–6, doi: 10.1109/ATNAC.2018.8615294.

[56] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Exp. Syst. Appl.*, vol. 185, Dec. 2021, Art. no. 115524, doi: 10.1016/j.eswa.2021.115524.

[57] H. Ma, J. Cao, B. Mi, D. Huang, Y. Liu, and S. Li, "A GRU-based lightweight system for CAN intrusion detection in real time," *Secur. Commun. Netw.*, vol. 2022, Jun. 2022, Art. no. e5827056, doi: 10.1155/2022/5827056.

[58] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep recurrent neural network for intrusion detection in SDN-based networks," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Jun. 2018, pp. 202–206, doi: 10.1109/NETSOFT.2018.8460090.

[59] A. S. Qureshi, A. Khan, N. Shamim, and M. H. Durad, "Intrusion detection using deep sparse auto-encoder and self-taught learning," *Neural Comput. Appl.*, vol. 32, no. 8, pp. 3135–3147, Apr. 2020, doi: 10.1007/s00521-019-04152-6.

[60] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.

[61] S. Gurung, M. K. Ghose, and A. Subedi, "Deep learning approach on network intrusion detection system using NSL-KDD dataset," *Int. J. Comput. Netw. Inf. Secur.*, vol. 11, no. 3, pp. 8–14, Mar. 2019, doi: 10.5815/ijcnis.2019.03.02.

[62] S. N. Mighan and M. Kahani, "A novel scalable intrusion detection system based on deep learning," *Int. J. Inf. Secur.*, vol. 20, no. 3, pp. 387–403, Jun. 2021, doi: 10.1007/s10207-020-00508-5.

[63] H. Zhang, C. Q. Wu, S. Gao, Z. Wang, Y. Xu, and Y. Liu, "An effective deep learning based scheme for network intrusion detection," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Beijing, China, Aug. 2018, pp. 682–687, doi: 10.1109/ICPR.2018.8546162.

[64] Y. Dong, R. Wang, and J. He, "Real-time network intrusion detection system based on deep learning," in *Proc. IEEE 10th Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Beijing, China, Oct. 2019, pp. 1–4, doi: 10.1109/ICSESS47205.2019.9040718.

[65] D. Berman, A. Buczak, J. Chavis, and C. Corbett, "A survey of deep learning methods for cyber security," *Information*, vol. 10, no. 4, p. 122, Apr. 2019, doi: 10.3390/info10040122.

[66] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "A novel two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019, doi: 10.1109/ACCESS.2019.2899721.

[67] Z. K. Maseer, R. Yusof, S. A. Mostafa, N. Bahaman, O. Musa, and B. A. S. Al-Rimy, "DeepIoT.IDS: hybrid deep learning for enhancing IoT network intrusion detection," *Comput., Mater. Continua*, vol. 69, no. 3, pp. 3945–3966, 2021, doi: 10.32604/cmc.2021.016074.

[68] W. Peng, X. Kong, G. Peng, X. Li, and Z. Wang, "Network intrusion detection based on deep learning," in *Proc. Int. Conf. Commun., Inf. Syst. Comput. Eng. (CISCE)*, Haikou, China, Jul. 2019, pp. 431–435, doi: 10.1109/CISCE.2019.00102.

[69] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018, doi: 10.1109/ACCESS.2018.2869577.

[70] A. Vinolia, N. Kanya, and V. N. Rajavarman, "Machine learning and deep learning based intrusion detection in cloud environment: A review," in *Proc. 5th Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, Jan. 2023, pp. 952–960, doi: 10.1109/ICSSIT55814.2023.10060868.

[71] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102419, doi: 10.1016/j.jisa.2019.102419.

[72] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, Funchal, Portugal, 2018, pp. 108–116, doi: 10.5220/0006639801080116.

[73] M. Haggag, M. M. Tantawy, and M. M. S. El-Soudani, "Implementing a deep learning model for intrusion detection on apache spark platform," *IEEE Access*, vol. 8, pp. 163660–163672, 2020, doi: 10.1109/ACCESS.2020.3019931.

[74] A. Y. Hussein, P. Falcarin, and A. T. Sadiq, "Enhancement performance of random forest algorithm via one hot encoding for IoT IDS," *Periodicals Eng. Natural Sci. (PEN)*, vol. 9, no. 3, p. 579, Aug. 2021, doi: 10.21533/pen.v9i3.2204.

[75] M. Azizjon, A. Jumabek, and W. Kim, "1D CNN based network intrusion detection with normalization on imbalanced data," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2020, pp. 218–224, doi: 10.1109/ICAIIC48513.2020.9064976.

[76] B. Jothi and M. Pushpalatha, "WILS-TRS—A novel optimized deep learning based intrusion detection framework for IoT networks," *Pers. Ubiquitous Comput.*, vol. 27, no. 3, pp. 1285–1301, Jun. 2023, doi: 10.1007/s00779-021-01578-5.

[77] H. Shah, D. Shah, N. K. Jadav, R. Gupta, S. Tanwar, O. Alfarraj, A. Tolba, M. S. Raboaca, and V. Marina, "Deep learning-based malicious smart contract and intrusion detection system for IoT environment," *Mathematics*, vol. 11, no. 2, p. 418, Jan. 2023, doi: 10.3390/math11020418.

[78] R. Alghamdi and M. Bellaiche, "An ensemble deep learning based IDS for IoT using lambda architecture," *Cybersecurity*, vol. 6, no. 1, p. 5, Mar. 2023, doi: 10.1186/s42400-022-00133-w.

[79] N. Abedzadeh and M. Jacobs, "A survey in techniques for imbalanced intrusion detection system datasets," *Int. J. Comput. Syst. Eng.*, vol. 17, no. 1, pp. 9–18, Jan. 2023.

[80] Y. Fu, Y. Du, Z. Cao, Q. Li, and W. Xiang, "A deep learning model for network intrusion detection with imbalanced data," *Electronics*, vol. 11, no. 6, p. 898, Mar. 2022, doi: 10.3390/electronics11060898.

[81] S. S. Gopalan, D. Ravikumar, D. Linekar, A. Raza, and M. Hasib, "Balancing approaches towards ML for IDS: A survey for the CSE-CIC IDS dataset," in *Proc. Int. Conf. Commun., Signal Process., Appl. (ICCSPA)*, Mar. 2021, pp. 1–6, doi: 10.1109/ICCSPA49915.2021.9385742.

[82] J. Cui, L. Zong, J. Xie, and M. Tang, "A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data," *Int. J. Speech Technol.*, vol. 53, no. 1, pp. 272–288, Jan. 2023, doi: 10.1007/s10489-022-03361-2.

[83] A. S. Dina, A. B. Siddique, and D. Manivannan, "A deep learning approach for intrusion detection in Internet of Things using focal loss function," *Internet Things*, vol. 22, Jul. 2023, Art. no. 100699, doi: 10.1016/j.iot.2023.100699.

[84] O. D. Okey, D. C. Melgarejo, M. Saadi, R. L. Rosa, J. H. Kleinschmidt, and D. Z. Rodríguez, "Transfer learning approach to IDS on cloud IoT devices using optimized CNN," *IEEE Access*, vol. 11, pp. 1023–1038, 2023, doi: 10.1109/ACCESS.2022.3233775.

[85] T. K. Boppana and P. Bagade, "GAN-AE: An unsupervised intrusion detection system for MQTT networks," *Eng. Appl. Artif. Intell.*, vol. 119, Mar. 2023, Art. no. 105805, doi: 10.1016/j.engappai.2022.105805.

[86] M. Mayuranathan, M. Murugan, and V. Dhanakoti, *Retraction Note to: Best Features Based Intrusion Detection System by RBM Model for Detecting DDoS in Cloud Environment*. Berlin, Germany: Springer, 2023.

[87] G. Logeswari, S. Bose, and T. Anitha, "An intrusion detection system for SDN using machine learning," *Intell. Autom. Soft Comput.*, vol. 35, no. 1, pp. 867–880, 2023, doi: 10.32604/iasc.2023.026769.

[88] H. Güney, "Preprocessing impact analysis for machine learning-based network intrusion detection," *Sakarya Univ. J. Comput. Inf. Sci.*, vol. 6, no. 1, pp. 67–79, Apr. 2023, doi: 10.35377/saucis...1223054.

[89] M. Arunkumar and K. A. Kumar, "GOSVM: Gannet optimization based support vector machine for malicious attack detection in cloud environment," *Int. J. Inf. Technol.*, vol. 15, no. 3, pp. 1653–1660, Mar. 2023, doi: 10.1007/s41870-023-01192-z.

[90] O. Elnakib, E. Shaaban, M. Mahmoud, and K. Emara, "EIDM: Deep learning model for IoT intrusion detection systems," *J. Supercomput.*, vol. 79, no. 12, pp. 13241–13261, Aug. 2023, doi: 10.1007/s11227-023-05197-0.

[91] P. Sanju, "Enhancing intrusion detection in IoT systems: A hybrid metaheuristics-deep learning approach with ensemble of recurrent neural networks," *J. Eng. Res.*, Jun. 2023, Art. no. 100122, doi: 10.1016/j.jer.2023.100122.

[92] A. Fatani, A. Dahou, M. A. Elaziz, M. A. A. Al-Qaness, S. Lu, S. A. Alfadhli, and S. S. Alresheedi, "Enhancing intrusion detection systems for IoT and cloud environments using a growth optimizer algorithm and conventional neural networks," *Sensors*, vol. 23, no. 9, p. 4430, Apr. 2023, doi: 10.3390/s23094430.

[93] N. F. Syed, M. Ge, and Z. Baig, "Fog-cloud based intrusion detection system using recurrent neural networks and feature selection for IoT networks," *Comput. Netw.*, vol. 225, Apr. 2023, Art. no. 109662, doi: 10.1016/j.comnet.2023.109662.

[94] M. A. Elaziz, M. A. A. Al-qaness, A. Dahou, R. A. Ibrahim, and A. A. A. El-Latif, "Intrusion detection approach for cloud and IoT environments using deep learning and capuchin search algorithm," *Adv. Eng. Softw.*, vol. 176, Feb. 2023, Art. no. 103402, doi: 10.1016/j.advengsoft.2022.103402.

[95] H. Alamro, R. Marzouk, N. Alruwais, N. Negm, S. S. Aljameel, M. Khalid, M. A. Hamza, and M. I. Alsaid, "Modeling of blockchain assisted intrusion detection on IoT healthcare system using ant lion optimizer with hybrid deep learning," *IEEE Access*, vol. 11, pp. 82199–82207, 2023, doi: 10.1109/ACCESS.2023.3299589.

HAN LIAO (Member, IEEE) received the B.S. degree in communication engineering from South China Agricultural University, China, in 2016, and the M.S. degree in computer science from Universiti Kebangsaan Malaysia, in 2023. From 2016 to 2020, he was a Communication Network Architecture Engineer with the Communication Industry. His main research interests include cybersecurity, network communication, artificial intelligence, and the IoT.

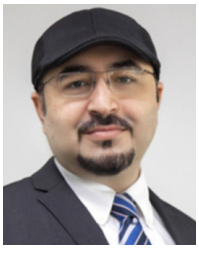
MOHD ZAMRI MURAH received the B.Sc. and M.Sc. degrees in statistics from The University of Iowa, Iowa, USA, in 1987 and 1989, respectively. He is currently a Senior Lecturer with the Center for Cyber security, Universiti Kebangsaan Malaysia (UKM), Malaysia. His current research interests include the development of deep learning models for cybersecurity, automated penetration testing, and cyber range.

MOHAMMAD KAMRUL HASAN (Senior Member, IEEE) received the Ph.D. degree in electrical and communication engineering from the Faculty of Engineering, International Islamic University, Malaysia, in 2016. He is currently an Associate Professor and the Head of the Network and Communication Technology Research Laboratory, Center for Cyber Security, Universiti Kebangsaan Malaysia. He specializes in elements pertaining to cutting-edge information-centric networks, computer networks, data communication and security, mobile networks and privacy protection, cyber-physical systems, the Industrial IoT, transparent AI, and electric vehicle networks. He has published more than 220 indexed papers in ranked journals and conference proceedings. He is a member of the Institution of Engineering and Technology and the Internet Society. He is a certified Professional Technologist in Malaysia. He served as the Chair for the IEEE Student Branch, from 2014 to 2016. He has actively participated in many events/workshops/trainings for the IEEE humanity programs. He is the general chair, the co-chair, and a speaker for conferences and workshops for the shake of society and academy knowledge building and sharing and learning. He is an Editorial Member in many prestigious high-impact journals, such as IEEE, IET, Elsevier, Frontier, and MDPI.

AZANA HAFIZAH MOHD AMAN received the B.Eng., M.Sc., and Ph.D. degrees in computer and information engineering from International Islamic University Malaysia. She is currently a Senior Lecturer with the Research Center for Cyber Security, Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia (UKM), Malaysia. Her research interests include computer systems and networking, information, and network security, the IoT, cloud computing, and big data.

JIN FANG (Member, IEEE) received the M.S. degree in computer science from Universiti Kebangsaan Malaysia. He participated in and completed one national-level project in China. Moreover, he has presided over and participated in three provincial-level projects in Guangdong Province. His primary research interests include artificial intelligence, modern educational technology, and the IoT.

XUTING HU received the M.S. degree in educational technology from Anhui Normal University, China, in 2019. She is currently a University Lecturer. From 2019 to 2023, she hosted and participated in two provincial-level projects in Anhui Province. Her main courses of teaching are computer fundamentals and office automation. Her research interests include higher education, information technology education, and computer application technology.



ATTA UR REHMAN KHAN (Senior Member, IEEE) is currently an Associate Professor with the College of Engineering and Information Technology, Ajman University, United Arab Emirates. In the past, he was a Postgraduate Program Coordinator with Sohar University, the Director of the National Cybercrime Forensics Laboratory Pakistan, and the Head of the Cybersecurity Center, Air University. His research interests include cybersecurity, mobile cloud computing, ad hoc networks, and the IoT. He serves as a domain expert for multiple international research funding bodies and has received multiple awards, fellowships, and research grants. He is a Senior Member of ACM and a steering committee member/track chair/technical program committee (TPC) member of more than 85 international conferences. He is serving as an Associate Editor for IEEE Access (Elsevier) and *Journal of Network and Computer Applications*, an Associate Technical Editor for *IEEE Communications Magazine*, and an Editor for *Cluster Computing* (Springer), *Computer Journal* (Oxford), IEEE SDN Newsletter, *KSII Transactions on Internet and Information Systems*, and *Ad hoc and Sensor Wireless Networks*. For more updated information, visit his website at www.attaurrehman.com.