Hindawi
Complexity
Volume 2021, Article ID 5538896, 23 pages
https://doi.org/10.1155/2021/5538896

Research Article

Trust in Intrusion Detection Systems: An Investigation of Performance Analysis for Machine Learning and Deep Learning Models

Basim Mahbooba,[1] **Radhya Sahal**,[2] **Wael Alosaimi**,[3] **and Martin Serrano**[1]

[1]*Data Science Institute Insight Centre for Data Analytics, National University of Ireland Galway, Galway, Ireland*
[2]*Faculty of Computer Science and Engineering, Hodeidah University, Al Hudaydah, Yemen*
[3]*Department of Information Technology, College of Computers and Information Technology, Taif University, P.O.Box 11099, Taif 21944, Saudi Arabia*

Correspondence should be addressed to Radhya Sahal; radhya.sahal.dsi@gmail.com

Received 20 January 2021; Accepted 21 February 2021; Published 8 March 2021

Academic Editor: Ahmed Mostafa Khalil

To design and develop AI-based cybersecurity systems (e.g., intrusion detection system (IDS)), users can justifiably trust, one needs to evaluate the impact of trust using machine learning and deep learning technologies. To guide the design and implementation of trusted AI-based systems in IDS, this paper provides a comparison among machine learning and deep learning models to investigate the trust impact based on the accuracy of the trusted AI-based systems regarding the malicious data in IDs. The four machine learning techniques are decision tree (DT), K nearest neighbour (KNN), random forest (RF), and naïve Bayes (NB). The four deep learning techniques are LSTM (one and two layers) and GRU (one and two layers). Two datasets are used to classify the IDS attack type, including wireless sensor network detection system (WSN-DS) and KDD Cup network intrusion dataset. A detailed comparison of the eight techniques' performance using all features and selected features is made by measuring the accuracy, precision, recall, and F1-score. Considering the findings related to the data, methodology, and expert accountability, interpretability for AI-based solutions also becomes demanded to enhance trust in the IDS.

1. Introduction

Cybersecurity system is developed based on different peers including technology, processes, and people. The relationships among these peers are the core of the trust management in the cybersecurity. For example, (1) the relationship between people and groups, (2) the relationship between people and organizations, and (3) the relationship between people and technology. The trusted peers are deployed in the cybersecurity system which aims to detect the cyberattacks [1].

Artificial intelligence (AI) defines a set of techniques that simulates the human intelligence in machines. The core idea of these techniques is extracting the knowledge from a collection of data. Consequently, there is no certain grantee level to trust the AI-based techniques due to the three aspects: (1) quality of data, (2) the degree of complexity for the methodology that was used to design the AI systems, and (3) AI engineer's experiences. According to the context of this work, a set of peers of AI technologies can be interacted to perform the trust for the cybersecurity systems. Therefore, we cannot do trust AI technologies to prevent the cybersecurity systems against cyberattacks. Consequently, this work in this paper study the trust for AI-based solutions including machine learning and deep learning in cybersecurity systems considering (1) data, (2) methodology, and (3) expert accountability. To do so, we will investigate AI-based solutions for trust context of cybersecurity in terms of the quality of data, methodologies, and experiences.

1.1. Trust in Intrusion Detection Systems. Cybersecurity system is developed based on different peers, including technology, processes, and people. The relationships among these peers are the core of the trust management in the cybersecurity. For example, (1) the relationship between people and groups, (2) the relationship between people and organizations, and (3) the relationship between people and technology. The trusted peers are deployed in the cybersecurity system to detect the cyberattacks [1].

"Trust" is commonly used word in cybersecurity which describes the connection a foundation that must be established for cybersecurity systems including machine-to-machine (M2M) system. In M2M systems, trust can be defined as the confidence between machines to identify and manage their information technology assets. To achieve the trust chain between machines, cryptography, digital signatures, electronic certificates, and AI-based solutions are used. As these techniques are established to perform trust for M2M systems, trust seems like a simple function; it is often a fundamental challenge. In particular, the challenge of trust in cybersecurity is a broader notion about the quality of information being exchanged among machines, the methodologies used to design these techniques, and the expert accountability that use these techniques. According to this work, we will investigate AI-based solutions for trust context of cybersecurity. Many efforts have been made in research and industry to prevent the critical system from the cyberattacks. The IDS have received attention due to the continuously increasing cost to fight cybercrime [2]. The cybercrime type includes (1) malicious insiders, (2) denial of services, and (3) web-based attacks. Therefore, most companies and enterprises deploy cybersecurity systems (e.g., antivirus, firewall, and IDS).

The core function of the IDS is identifying the malicious attacks' activities in advanced before they access the information and harm the confidentiality of the critical systems [3]. This demand of the security systems from both known and unknown threats opens a challenge for the research communities and industry to design secure and trustful systems against the cyberattacks [4]. This also opens up the issue about how to successfully secure from both known and unknown threats. There is no straightforward answer to this because of the increasing number of threats every year [4]. Recently, AI-based technologies, including machine learning and deep learning, play a vital role in learning from the previous attacks' collected historical data. These models' extracted knowledge is used to enhance the trust in IDS [5].

1.2. Contribution. Our main contributions are summarized as follows:

- We develop investigation methodology to study the trust impact in intrusion detection, including the data, methodology, and expert accountability by analyzing machine learning and deep learning models' performance
- Collect intrusion detection using wireless sensor network detection system (WSN-DS) and KDD Cup network intrusion dataset
- Apply different feature engineering techniques, including correlation matrix
- Compare four machine learning (DT, KNN, RF, and NB) and deep learning (LSTM and GRU, using one and two layers) to study the trusted AI-based systems' accuracy regarding the malicious data to detect any intrusion in the system

1.3. Paper Organization. The rest of this paper is organized as follows. A review of relevant works is conducted in Section 2. The methodology is provided in Section 3. The experiments and results are described in Section 4. The discussion is introduced in Section 5. Finally, the paper is concluded in Section 6.

2. Related Work

Vinayakumar et al. [6] have used a deep neural network to develop IDS to predict unforeseen and unpredictable cyberattacks. Almomani et al. [7] have used artificial neural network (ANN) to develop IDS to classify different DoS attacks. The authors in [8] have used a multistage machine learning-based intrusion detection to detect and classify four types of jamming attacks. Abhale and Manivannan [9] have used different types of supervised machine learning to classify anomaly type of IDS. On the other hand, Alqahtani et al. [10] have proposed genetic-based extreme gradient boosting (XGBoot) to detect minority classes of attacks in highly imbalanced data traffics of wireless sensor networks. The authors in [11] have introduced an ensemble learning scheme for classifying network intrusion detection. However, Farrahi and Ahmadzadeh [12] have used various algorithms such as k-means clustering, Naïve Bayes, support vector machine, and OneR algorithms to classify regular traffic and DoS attack. Also, the genetic algorithm (GA) was implemented to detect the different types of intrusions [13].

Some researchers have used feature selection methods to select essential features will reduce the computational time of the algorithms. Due to network data's significant features, many IDS were developed with feature selection [14]. Chebrolu et al. [15] classified primary elements in constructing IDS that is very crucial for real-world intrusion detection. Zaman and Karray [16] implemented a feature selection technique to construct a lightweight IDS. Vimalkumar and Radhika [17] implemented principal component analysis- (PCA-) based feature selection technique in the big data framework for IDS. Balakrishnan et al. [18] developed an IDS model with a gain ratio as a feature selection technique. Most of the IDS-based studies focused on the performance of the implemented model. Alkasassbeh et al. [19] concentrate on different types of attack, such as **http flood, smurf, siddos, and udp flood**. They implemented various machine learning algorithms to detect DOS intrusions and demonstrated the high accuracy of 98.36% using multilevel perceptron (MLP). Peng et al. [20] proposed an IDS system based on a decision tree to improve detection efficiency. Their method showed better performance over Naïve Bayesian and KNN methods.

3. Research Methodology

This section will describe our approach to investigate the trust impact in intrusion detection using machine learning and deep learning models. To do so, five phases are developed including (1) data collection to describe the datasets and their characteristics, (2) splitting datasets, (3) feature extraction methods, (4) optimization and training models, and (5) the evaluation metrics that will be used [19] for performance comparison (see Figure 1). Further details about the developed phases are described as follows.

3.1. Data Collection. In this section, we provide a description of the datasets used to find the optimal machine learning model and deep learning model that obtains the best performance for attack type classification in IDS. Two datasets were collected from wireless sensor network detection system (WSN-DS) and KDD Cup network intrusion dataset.

3.1.1. WSN-DS Dataset. The first dataset is WSN-DS which is a specialized dataset for detecting intrusions in wireless sensor networks. The WSN-DS dataset is collected by [7] to help better detect and classify types of denial-of-service (DoS) attacks. According to this work, we have used the WSN-DS dataset to study the machine learning and deep learning models performances with respect to the sensor nodes that can be able to detect attacks' patterns from the normal traffic. Then, we have compared the machine learning and deep learning models' performances to study the impact of trust in machine learning and deep learning models' IDS.

The WSN-DS dataset contains 23 features extracted using LEACH routing protocol including *Id, Time, Is_CH, who_CH, RSSI, Dist_To_CH, M_D_CH, A_D_CH, ADV_S, ADV_R, JOIN_S, JOIN_R, ADV_SCH_S, ADV_SCH_R, Rank, DATA_S, DATA_R, Data_Sent_BS, Dist_CH_BS, Send_code, Current_Energy, Consumed_Energy, and Attack_Type* [7].

The dataset file has only 19 features including the class label [10]. These 19 features were *Id, Time, Is_CH, who_CH, Dist_To_CH, ADV_S, ADV_R, JOIN_S, JOIN_R, ADV_SCH_S, ADV_SCH_R, Rank, DATA_S, DATA_R, Data_Sent_BS, Dist_CH_BS, Send_code, Consumed_Energy, and Attack_Type*. The number of samples within the WSN-DS dataset is 374,662. These samples are distributed among five main groups of which four of them are types of DoS attack which are labeled as attacks including *Blackhole*, *Grayhole*, *Flooding*, and *Scheduling* attacks and *Normal*. The description of the attacks is as follows (see Table 1):

Blackhole attack: type of DoS attack where the attacker advertises itself at the beginning of the round to affect the LEACH protocol

Grayhole attack: type of DoS attack where the attacker advertises itself as a CH for other nodes to affect the LEACH protocol

Flooding attack: type of DoS attack where the attacker advertises itself by sending a large number of advertising CH messages to affect the LEACH protocol

Scheduling attack: type of DoS attack where the attacker acts as a CH and assigns all nodes the same time slot to send data during the setup phase of the LEACH protocol

Normal: it means no threat

Furthermore, Table 2 shows a set of descriptive statistics of the WSN-DS dataset using a set of statistical functions including count, mean, std, min, and max. We have ignored *Id* because it has been used to provide a unique symbolized number of the sensor node and no sense for compute statistics for it. Therefore, 18 features will be used in the next phases.

3.1.2. KDD Dataset. The second dataset is KDD Cup network intrusion dataset (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.htmachine learning). The data comes from DARPA 98 Intrusion Detection Evaluation by Lincoln laboratory at MIT. According to [21], these datasets were collected using multiple computers connected to the Internet to model a small US Air Force base of qualified personnel by using several simulated intrusions. According to this work, we have used KDD dataset to study the machine learning and deep learning models' performances concerning the sensor nodes that can detect attack patterns from the normal traffic. We then compared the machine learning and deep learning models' performances to study the impact of trust in machine learning and deep learning models' IDS. There are 42 attributes used in this dataset. The number of samples in the KDD dataset is 311,029. These samples are distributed among five main groups. Four of them are labeled as attacks including *Denial of Service, User to Root Remote-to-local, and PROBING* and *Normal*. The description of the attacks is as follows (see Table 3):

Denial-of-service (DOS) attack: it is done by illegal users causing resource constraint for the targeted systems. Consequently, the targeted system is being unable to provide efficient services to the legal users.

User to root (U2R) attack: the attacker belongs to the same group which tries to access the root of the system using a normal account within the network.

Remote to local (R2L) attack: the remote user has no account to access a specific node within the network. The attacker tries to gain local access by sending packets to explore any vulnerabilities within the network.

Probe attack (Probe): the attacker collects data about the network configuration to discover vulnerabilities and then accesses the network by loopholes.

Normal: it means no threat.

Furthermore, Table 4 shows a set of descriptive statistics of the KDD dataset using a set of statistical functions including count, mean, std, min, and max.

3.2. Splitting Dataset. In this step, the WSN-DS and KDD datasets are split into 30% training dataset and 70% testing dataset. The training set is fed into the machine learning/

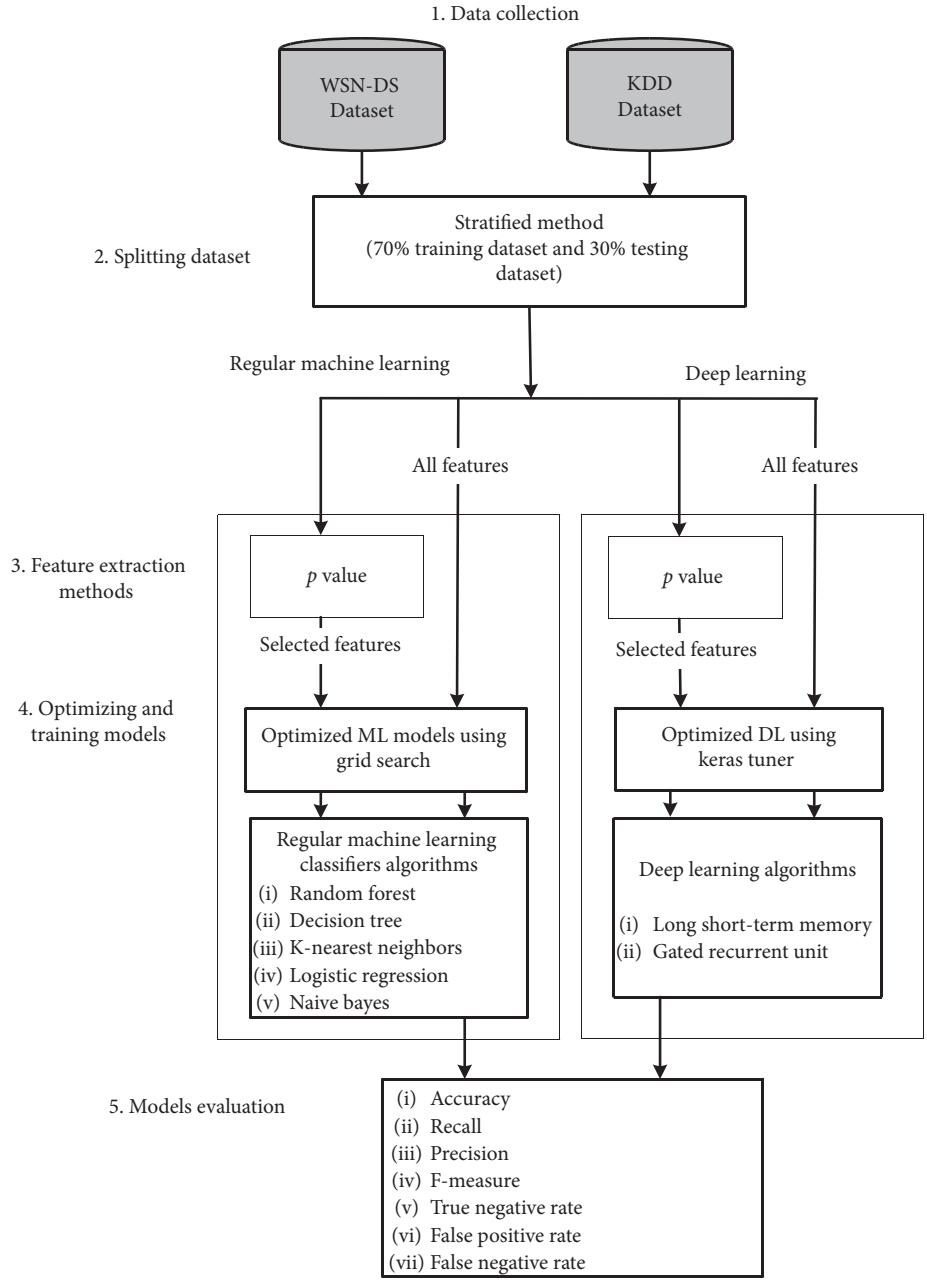

Figure 1: The workflow of developed machine learning and deep learning models.

Table 1: Types of attack in WSN datasets.

Types of attack	Quantity	Proportion (%)
Normal	340067	90.77
Grayhole	14597	3.90
Blackhole	10050	2.68
Scheduling	6639	1.77
Flooding	3313	0.88

deep learning models to let models learn from this data, while the unseen test set is used to evaluate machine learning/deep learning models. Table 5 and Table 6 present the number of instances in these two sets for WSN-DS and KDD datasets, respectively.

3.3. Feature Extraction. The key benefit of using feature selection methods is determining the relevant feature in the dataset. Therefore, feature selection is necessary for the machine learning and deep learning processes since sometimes irrelevant features affect the models' performance. According to this work context, feature selection enhances the classification accuracy of the attack types and reduces the model execution time. Also, we have used correlation matrix and p value to reduce the features that have less significance in the classified attack and affect the models' performances.

3.3.1. WSN-DS Dataset. For WSN-DS dataset, we have used a correlation matrix for feature analysis to calculate each

Table 2: Statistical analysis of WSN-DS dataset.

Feature No	Feature name	Count	Mean	std	min	25%	50%	75%	Max
1	Time	374661	1064.749	899.6462	50	353	803	1503	3600
2	Is_CH	374661	0.115766	0.319945	0	0	0	0	1
3	who_CH	374661	274980.4	389911.2	101000	107096	116072	215073	3402100
4	Dist_To_CH	374661	22.59938	21.95579	0	4.73544	18.37261	33.776	214.2746
5	ADV_S	374661	0.267698	2.061148	0	0	0	0	97
6	ADV_R	374661	6.940562	7.044319	0	3	5	7	117
7	JOIN_S	374661	0.779905	0.414311	0	1	1	1	1
8	JOIN_R	374661	0.737493	4.691498	0	0	0	0	124
9	SCH_S	374661	0.288984	2.754746	0	0	0	0	99
10	SCH_R	374661	0.747452	0.434475	0	0	1	1	1
11	Rank	374661	9.687104	14.6819	0	1	3	13	99
12	DATA_S	374661	44.85792	42.57446	0	13	35	62	241
13	DATA_R	374661	73.89004	230.2463	0	0	0	0	1496
14	Data_Sent_To_BS	374661	4.569448	19.67916	0	0	0	0	241
15	dist_CH_To_BS	374661	22.56274	50.2616	0	0	0	0	201.9349
16	send_code	374661	2.497957	2.407337	0	1	2	4	15
17	Consumed energy	374661	0.305661	0.669462	0	0.05615	0.09797	0.21776	45.09394
18	Attack type	374661	2.880615	0.564958	0	3	3	3	4

Table 3: Types of attack in KDD datasets.

Types of attack	Quantity	Proportion (%)
Denial of service (DOS)	229853	73.90
Remote to local (R2L)	16189	5.2
User to root (U2R)	228	0.07
PROBING (probe)	4166	1.34
Normal	60593	19.48

feature's relations with other features within the dataset as depicted in Figure 2. It can be seen that the features within the WSN-DS datasets do not have high correlations. We have removed only one feature, and then, we have calculated the p values for the rest 17 features.

As the attack type will be classified by machine learning and deep learning models, Table 7 presents the p values of the 17 features to choose the high correlated features for machine learning and deep learning models. Consequently, the features which have the correlation with attack type above 0.005 have been selected to be fitted in machine learning and deep learning models. In particular, 6 features have been selected based on their high correlations such as *Time, Dist_To_CH, JOIN_R, Rank, DATA_S, and send_code* and their p values are $7.00E-93$, $1.47E-24$, $5.93E-06$, 0.009842, $1.31E-32$, and $2.44E-125$, respectively.

3.3.2. KDD Dataset. For KDD dataset, we have also used a correlation matrix for feature analysis to calculate each feature's relations with other features within the dataset. We found that the features within the WSN-DS datasets do not have high correlations. We have removed 12 features, and then, we have calculated the p values for the rest 30 features.

As the attack type will be classified by machine learning and deep learning models, Table 8 presents the $p$values of the 30 features to choose the high correlated features for machine learning and deep learning models. Consequently, the features which have the correlation with attack type above 0.005 have been selected to be fitted in machine learning and deep learning models. In particular, 14 features have been selected based on their high correlations such as *duration, service, src_ bytes, land, urgent, hot, num_compromised, su_attempted, num_file_creations, num_shells, num_access_files, num outbound_cmds, is_host_login, and srv_diff_host_rate* and their p values are $2.33E-60$, 0.604708, $1.34E-68$, $9.51E-128$, 0.402631, $5.01E-230$, $2.69E-43$, $2.45E-21$, $3.48E-11$, $9.85E-43$, $6.50E-26$, 0.043069, 0.039867, and $1.56E-70$, respectively.

3.4. Machine Learning and Deep Learning Models. Regular machine learning models used in this paper are decision tree (DT), K nearest neighbour (KNN), random forest (RF), and naïve Bayes (NB). Moreover, among deep learning algorithms, we analyze the performance of LSTM (one and two layers) and GRU (one and two layers).

3.5. Optimization and Training Models. In this section, two categories of optimization and training models will be presented, including machine learning and deep learning.

3.5.1. Regular Machine Learning Models

(1) K-Fold Cross-Validation. The dataset is divided into k equal size of the sections in which the k-1 group is used to train the classifiers, and the remaining part is used to test the performance in each stage. The validation process is repeated k times. The output of the classifier is estimated based on the k tests. Various k values are selected for CV. In our analysis, we used $k = 10$, the 10-fold CV process, 70% of the data for training, and 30% of the data for testing purposes.

(2) Hyperparameter Tuning. It is used to pass various parameters to the model. Grid search is the most widely used method for hyperparameter tuning. Initially, the user defines a set of values for each hyperparameter. The model then tests

Table 4: Statistical analysis of KDD dataset.

Feature No	Feature name	Count	Mean	Std	Min	25%	50%	75%	Max
1	Duration	439880	21.55854	538.9139	0	0	0	0	58329
2	protocol_type	439880	0.414052	0.543742	0	0	0	1	2
3	Service	439880	22.25702	12.97022	0	14	14	22	65
4	Flag	439880	8.095365	1.803868	0	9	9	9	10
5	src_bytes	439880	3322.342	1047247	0	179	1032	1032	$6.93E+08$
6	dst_bytes	439880	878.5772	34494.07	0	0	0	0	5155468
7	Land	439880	$4.32E{-}05$	0.006572	0	0	0	0	1
8	wrong_fragment	439880	0.006502	0.135425	0	0	0	0	3
9	Urgent	439880	$1.59E{-}05$	0.00584	0	0	0	0	3
10	hot	439880	0.036876	0.814437	0	0	0	0	30
11	num_failed_logins	439880	0.000168	0.016378	0	0	0	0	5
12	logged_in	439880	0.146685	0.353792	0	0	0	0	1
13	num_compromised	439880	0.010919	1.899432	0	0	0	0	884
14	root_shell	439880	0.000102	0.010114	0	0	0	0	1
15	su_attempted	439880	$3.41E{-}05$	0.007539	0	0	0	0	2
16	num_root	439880	0.01186	2.124426	0	0	0	0	993
17	num_file_creations	439880	0.001075	0.097066	0	0	0	0	28
18	num_shells	439880	0.000109	0.011079	0	0	0	0	2
19	num_access_files	439880	0.000921	0.035509	0	0	0	0	8
20	num_outbound_cmds	439880	0	0	0	0	0	0	0
21	is_host_login	439880	0	0	0	0	0	0	0
22	is_guest_login	439880	0.001434	0.037847	0	0	0	0	1
23	Count	439880	348.345	209.268	0	132	510	511	511
24	srv_count	439880	312.4045	242.5829	0	12	510	511	511
25	serror_rate	439880	0.196742	0.396846	0	0	0	0	1
26	srv_serror_rate	439880	0.196653	0.397185	0	0	0	0	1
27	rerror_rate	439880	0.015626	0.121898	0	0	0	0	1
28	srv_rerror_rate	439880	0.01592	0.123188	0	0	0	0	1
29	same_srv_rate	439880	0.812805	0.373561	0	1	1	1	1
30	diff_srv_rate	439880	0.018327	0.078025	0	0	0	0	1
31	srv_diff_host_rate	439880	0.028292	0.140183	0	0	0	0	1
32	dst_host_count	439880	232.8175	64.40408	0	255	255	255	255
33	dst_host_srv_count	439880	196.4258	101.7786	0	178	255	255	255
34	dst_host_same_srv_rate	439880	0.783795	0.392965	0	0.97	1	1	1
35	dst_host_diff_srv_rate	439880	0.02371	0.089418	0	0	0	0.01	1
36	dst_host_same_src_port_rate	439880	0.633463	0.473977	0	0	1	1	1
37	dst_host_srv_diff_host_rate	439880	0.006677	0.043043	0	0	0	0	1
38	dst_host_serror_rate	439880	0.196806	0.396729	0	0	0	0	1
39	dst_host_srv_serror_rate	439880	0.196491	0.397116	0	0	0	0	1
40	dst_host_rerror_rate	439880	0.016285	0.120639	0	0	0	0	1
41	dst_host_srv_rerror_rate	439880	0.015624	0.120453	0	0	0	0	1
42	Attacktype	439880	0.20354	0.442414	0	0	0	0	4

Table 5: The WSN-DS dataset separated 70% training set and 30% testing set.

Attack type	Training set (70%)	Testing set (30%)
Normal	255049	85017
Grayhole	10947	3649
Blackhole	7537	2512
Scheduling	4978	1660
Flooding	2484	828
Sum	**280995**	**93666**

Table 6: The KDD dataset separated 70% training set and 30% testing set.

Attack type	Training set (70%)	Testing set (30%)
Dos	249609	106896
Normal	54826	23590
Probe	2672	1125
R2l	784	341
U2r	25	12
Sum	**307916**	**131964**

all values for each hyperparameter and selects the best value to achieve the best performance result.

3.5.2. Deep Learning Models. For hyperparameters optimization, we have used a Keras Tuner library to pick the optimal set of hyperparameters in hidden layers (LSTM or GRU) and dropout layers. We set different values for different parameters, which are the number of neurons, reg_rate for l2 regularization technique [22], and the dropout rate for the dropout layers [23]. For this, we have

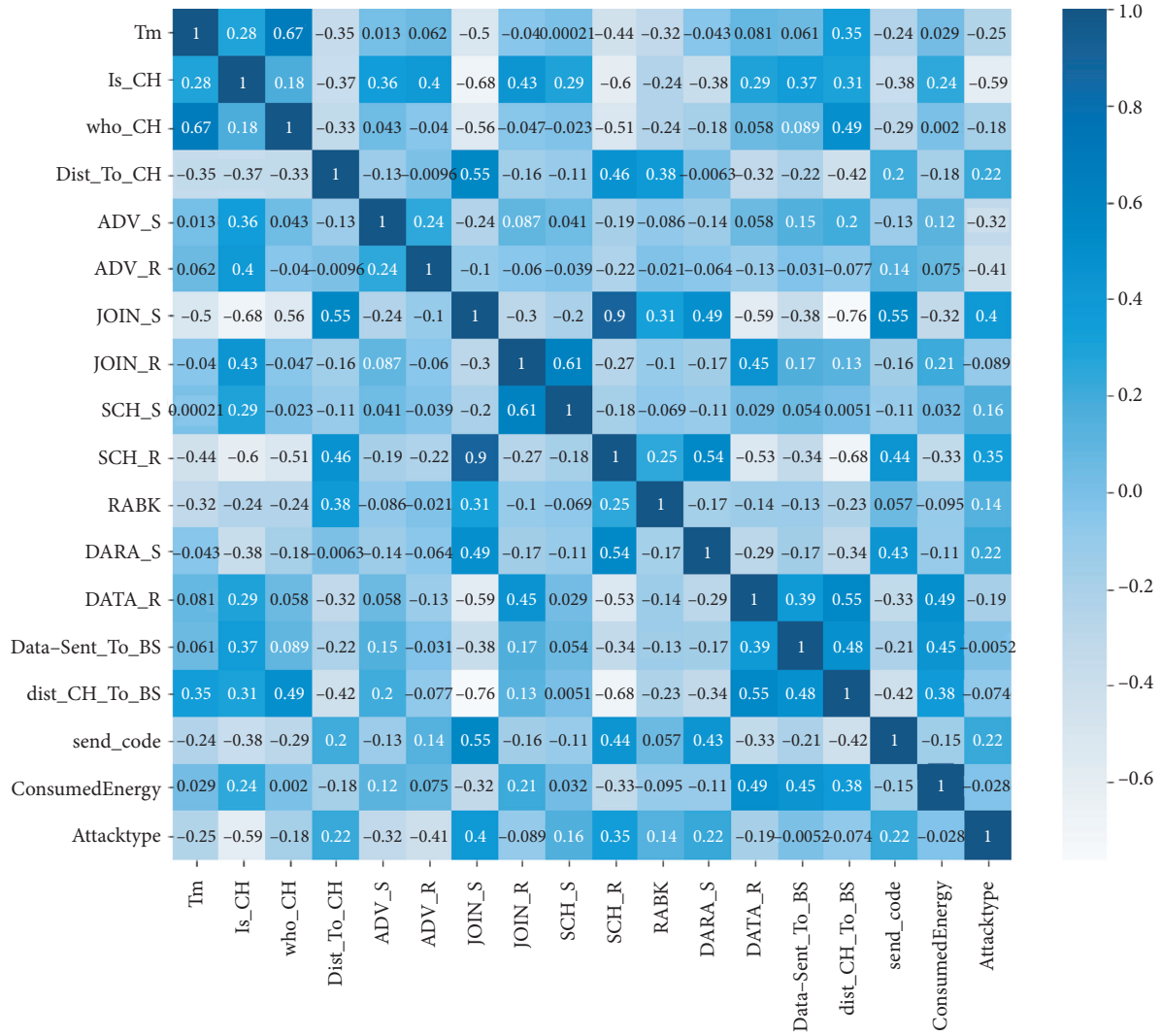

Figure 2: Correlation matrix between features for WSN dataset.

Table 7: Feature selection of WSN dataset using p values.

Feature No	Feature name	p value	Statistically significant
1	Time	$7.00E-93$	Yes
2	Is_CH	0	No
3	who_CH	0	No
4	Dist_To_CH	$1.47E-24$	Yes
5	ADV_S	0	No
6	ADV_R	0	No
7	JOIN_S	0	No
8	JOIN_R	$5.93E-06$	Yes
9	SCH_S	0	No
10	SCH_R	0	No
11	Rank	0.009842	Yes
12	DATA_S	$1.31E-32$	Yes
13	DATA_R	0	No
14	Data_Sent_To_BS	0	No
15	dist_CH_To_BS	0	No
16	send_code	$2.44E-125$	Yes
17	Consumed energy	0	No

Table 8: Feature selection of KDD dataset using p values.

Feature No	Feature name	p value	Statistically significant
1	Duration	$2.33E-60$	Yes
2	Service	0.604708	Yes
3	src_bytes	$1.34E-68$	Yes
4	Land	$9.51E-128$	Yes
5	Urgent	0.402631	Yes
6	hot	$5.01E-230$	Yes
7	num_compromised	$2.69E-43$	Yes
8	su_attempted	$2.45E-21$	Yes
9	num_file_creations	$3.48E-11$	Yes
10	num_shells	$9.85E-43$	Yes
11	num_access_files	$6.50E-26$	Yes
12	num_outbound_cmds	0.043069	Yes
13	is_host_login	0.039867	Yes
14	srv_diff_host_rate	$1.56E-70$	Yes
15	protocol_type	0	No
16	Flag	0	No
17	dst_bytes	0	No
18	wrong_fragment	0	No
19	num_failed_logins	0	No
20	logged_in	0	No
21	root_shell	0	No
22	is_guest_login	0	No
23	Count	0	No
24	serror_rate	0	No
25	rerror_rate	0	No
26	same_srv_rate	0	No
27	diff_srv_rate	0	No
28	dst_host_count	0	No
29	dst_host_diff_srv_rate	0	No
30	dst_host_srv_diff_host_rate	0	No

applied the Keras Tuner on the training dataset to select the best parameters, as shown in Table 9.

3.6. Evaluating Models. Seven standard metrics were utilized to evaluate the models' accuracy, precision, recall, and F1-score. TP is true positive, TN is true negative, FP is false positive, and FN is a false negative. We will consider four metrics for our experimental results, including accuracy, precision, recall, and F1-score.

4. Experiments and Results

This section describes the results of applying machine learning models (DT, KNN, RF, and NB) and four deep learning models (LSTM and GRU, using one and two layers), including cross-validation results and testing results. Each model performance is discussed using two datasets, including WSN-DS and KKD.

4.1. Experiment Setup. The machine learning models and deep learning models which are applied on the collected datasets has been developed in Python3 using Anaconda Python 3. The experiments have been conducted using a laptop with a specification of 20 GB of RAM, 7 cores, and 100 GB disk. The machine learning models and deep learning models have been trained using 70% dataset, while the rest of 30% of dataset has been used for testing. The machine learning models have been implemented using Sklearn library, while the deep learning models have been implemented using Tensorflow and Keras packages.

Table 9: Hyperparameters' configurations selected by Keras Tuner.

Parameter	The value
Dropout rate	Within the range of 0.1 rate to 0.5 rate
The number of neurons	Within the range of 10 neuron to 200 neurons
Regularization l2	0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5

4.2. WSN-DS Dataset. This section presents the results of applying machine learning models (DT, KNN, RF, and NB) and four deep learning models (LSTM and GRU, using one and two layers), and cross-validation results and testing results are described. Each machine learning model and deep learning model performance is discussed using full features and selected features to classify five classes of attack types, including *Normal, Grayhole, Blackhole, Scheduling,* and *Flooding*. All the positive and negative rates results of the cross-validation and testing performances used to compute accuracy, precision, recall, and F-score matrices are presented in Tables 10–13.

Table 10: The performance of machine learning for WSN dataset using all features.

Model	Evaluation metric	Cross-validation performance					Testing performance				
		Normal	Grayhole	Blackhole	Scheduling	Flooding	Normal	Grayhole	Blackhole	Scheduling	Flooding
DT	TN	99.97	99.96	99.92	97.74	99.89	99.96	99.98	99.9	98.23	99.89
	FP	0.03	0.04	0.08	2.26	0.11	0.04	0.02	0.1	1.77	0.11
	FN	0.94	4.73	1.8	0.21	7.1	1.18	3.11	1.44	0.23	6.07
	Accuracy	99.95	99.91	99.86	99.6	99.77	99.93	99.95	99.85	99.63	99.78
	Precision	98.93	95.24	98.12	99.77	94.11	98.56	97.32	97.6	99.82	93.74
	Recall	99.06	95.27	98.2	99.79	92.9	98.82	96.89	98.56	99.77	93.93
	F-score	98.99	95.24	98.16	99.78	93.5	98.69	97.1	98.08	99.79	93.83
KNN	TN	99.71	99.85	99.56	84.25	99.89	99.72	99.9	99.57	84.14	99.89
	FP	0.29	0.15	0.44	15.75	0.11	0.28	0.1	0.43	15.86	0.11
	FN	10.35	25.36	20.58	0.53	35.25	9.82	22.67	21.01	0.49	36.23
	Accuracy	99.44	99.63	98.78	98.06	99.27	99.46	99.72	98.75	98.09	99.26
	Precision	89.46	81.57	88.02	98.41	91.68	89.95	85.71	88.35	98.4	91.57
	Recall	89.65	74.64	79.42	99.47	64.75	90.18	77.33	78.99	99.51	63.77
	F-score	89.55	77.94	83.48	98.94	75.9	90.07	81.31	83.41	98.96	75.18
RF	TN	99.96	99.94	99.95	98.26	99.99	99.96	99.96	99.93	98.42	100
	FP	0.04	0.06	0.05	1.74	0.01	0.04	0.04	0.07	1.58	0
	FN	0.46	1.36	1.42	0.12	7.23	0.52	0.89	1.53	0.12	6.07
	Accuracy	99.95	99.93	99.89	99.73	99.86	99.94	99.95	99.87	99.74	99.89
	Precision	98.66	94	98.67	99.82	99.52	98.45	95.3	98.3	99.84	99.78
	Recall	99.54	98.64	98.58	99.88	92.77	99.48	99.11	98.47	99.88	93.93
	F-score	99.09	96.26	98.63	99.85	96.02	98.96	97.17	98.39	99.86	96.77
NB	TN	99.97	93.06	96.4	98.81	90.56	99.97	93.05	96.32	99.08	90.26
	FP	0.03	6.94	3.6	1.19	9.44	0.03	6.95	3.68	0.92	9.74
	FN	65.69	18.57	53.21	16.99	18.31	65.31	20.00	52.24	17.42	15.99
	Accuracy	98.21	92.96	94.47	84.47	90.4	98.20	92.95	94.40	84.10	90.15
	Precision	97.14	9.48	34.53	99.85	13.5	97.07	8.51	34.91	99.89	13.37
	Recall	34.31	81.43	46.79	83.01	81.69	34.69	80.00	47.76	82.58	84.01
	F-score	50.67	16.98	39.72	90.65	23.17	51.11	15.38	40.33	90.41	23.07

4.2.1. Regular Machine Learning Using All Features

(1) Cross-Validation Results. Table 10 shows the machine learning models' performance using the unseen testing WSN-DS dataset. For the normal class, RF is the highest performance model (accuracy of 99.94%, precision of 98.45%, recall of 99.48%, and F-score 98.96%). DT, KNN, and NB models have achieved the second, third, and fourth ranks on the average of accuracy over unseen data by 99.93%, 99.46%, and 98.20%, respectively. For the Grayhole class, DT is the highest performance model (accuracy of 99.95%, precision of 97.32%, recall of 96.89%, and F-score 97.1%). At the same time, NB is the worst performing model (accuracy of 92.95%, precision of 8.51%, recall of 80.00%, and F-score 15.38%). Similar to Blackhole class, RF is the highest performance model (accuracy of 99.87%, precision of 98.3%, recall of 98.47%, and F-score 98.39%). Regarding scheduling and flooding classes, RF obtained the highest performance by accuracy of 99.74% and 99.89%, respectively. However, NB is the worst performing model for both classes, including accuracy of 84.10% for scheduling and 90.15% for flooding. Yet, the NB model using Scheduling classes has the lowest accuracy performance among all models and classes for unseen data in terms of accuracy, 84.65%.

(2) Testing Results. As shown in the results, for the normal class, the RF model has achieved the highest performances among other models (accuracy of 99.94%, precision of 98.45%, recall of 99.48%, and F-score 98.96%). However, the NB model has recorded the worst performances among other models (accuracy of 98.20%, precision of 97.07%, recall of 34.69%, and F-score 51.11%). For the grayhole class, DT and RF have achieved the highest performances among other models (accuracy of 99.95%). However, the NB model has recorded the worst performances among other models (accuracy of 92.95%, precision of 8.51%, recall of 80.0015.38%, and F-score 15.38%). Like the blackhole class, the RF model has achieved the highest performances among other models (accuracy of 99.87%, precision of 98.3%, a recall of 98.47%, and F-score of 98.39%). However, the NB model has recorded the worst performances among other models (accuracy of 94.40%, precision of 34.91%, recall of 47.76%, and F-score of 40.33%). Regarding scheduling class, the RF model has achieved the highest performances among other models (accuracy of 99.74%, precision of 99.84%, recall of 99.88%, and F-score of 99.86%). However, the NB model has recorded the worst performances among other models (accuracy of 84.10%, precision of 99.89%, recall of 82.58%, and F-score of 90.41%). Flooding class like the other classes, the RF model, has achieved the highest performances among other models (accuracy of 99.89%, precision of 99.78%, recall of 93.93%, and F-score of 96.77%). However, the NB model has recorded the worst performances among other models (accuracy of 90.15%, precision of 13.37%, recall

Table 11: The performance of machine learning for WSN dataset using selected features.

Model	Evaluation metric	Cross-validation performance					Testing performance				
		Normal	Grayhole	Blackhole	Scheduling	Flooding	Normal	Grayhole	Blackhole	Scheduling	Flooding
DT	TN	99.95	99.95	99.9	97.72	99.91	99.96	99.96	99.91	97.41	99.91
	FP	0.05	0.05	0.1	2.28	0.09	0.04	0.04	0.09	2.59	0.09
	FN	1.32	5.18	2.32	0.22	6.87	1.19	6.64	2.55	0.21	7.17
	Accuracy	99.92	99.9	99.82	99.59	99.79	99.92	99.90	99.81	99.57	99.78
	Precision	98.29	94.16	97.65	99.77	94.76	98.37	94.96	97.75	99.74	94.77
	Recall	98.68	94.82	97.68	99.78	93.13	98.81	93.36	97.45	99.79	92.83
	F-score	98.48	94.48	97.66	99.77	93.93	98.59	94.15	97.60	99.76	93.79
KNN	TN	99.59	99.81	99.38	80.16	99.98	99.63	99.81	99.37	80.60	99.98
	FP	0.41	0.19	0.62	19.84	0.02	0.37	0.19	0.63	19.40	0.02
	FN	16.13	32.11	27.18	0.6	40.55	16.24	28.74	27.27	0.59	38.31
	Accuracy	99.17	99.53	98.34	97.62	99.26	99.21	99.55	98.33	97.67	99.30
	Precision	84.84	76.46	82.55	98.01	97.7	86.26	76.52	82.32	98.05	98.27
	Recall	83.87	67.89	72.82	99.4	59.45	83.76	71.26	72.73	99.41	61.69
	F-score	84.34	71.88	77.36	98.7	73.91	84.99	73.80	77.23	98.72	75.80
RF	TN	99.95	99.93	99.93	98.28	99.96	99.95	99.94	99.95	98.28	99.96
	FP	0.05	0.07	0.07	1.72	0.04	0.05	0.06	0.05	1.72	0.04
	FN	0.91	0.96	1.89	0.17	7.04	0.60	0.72	2.03	0.15	6.99
	Accuracy	99.93	99.92	99.86	99.69	99.84	99.93	99.94	99.87	99.71	99.84
	Precision	98.16	92.81	98.24	99.82	97.84	98.11	93.94	98.68	99.82	97.66
	Recall	99.09	99.04	98.11	99.83	92.96	99.40	99.28	97.97	99.85	93.01
	F-score	98.62	95.82	98.17	99.83	95.33	98.75	96.54	98.32	99.84	95.28
NB	TN	99.48	91.62	96.04	56.01	99.95	99.50	91.49	95.95	57.42	99.96
	FP	0.52	8.38	3.96	43.99	0.05	0.50	8.51	4.05	42.58	0.04
	FN	63.86	25.09	70.4	12.48	40.93	61.82	26.45	70.87	12.58	40.60
	Accuracy	97.78	91.47	93.45	84.61	99.23	97.85	91.33	93.34	84.65	99.24
	Precision	65.47	7.39	23.26	95.14	95.83	67.68	7.16	22.55	95.28	96.20
	Recall	36.14	74.91	29.6	87.52	59.07	38.18	73.55	29.13	87.42	59.40
	F-score	46.54	13.44	26.04	91.17	73.08	48.82	13.05	25.42	91.18	73.45

of 84.01%, and F-score of 23.07%). Based on these results, RF and DT models for the grayhole class are the best performing models with respect to other models, while the NB model for scheduling class is the worst performing model.

4.2.2. Regular Machine Learning Using Selected Features. In this section, the machine learning performance results of applying feature selection using WSN-DS dataset are presented.

(1) Cross-Validation Results. This section discusses the 10-fold CV results of four machine learning models (DT, KNN, RF, and NB) over the WSN-DS dataset with selected features, as shown in Table 11. As shown in the normal class results, RF has achieved the highest performances among other models and other classes (accuracy of 99.93%, precision of 98.16%, recall of 99.09%, and F-score of 98.62%). The DT model has achieved the second one, KNN is the third one, and the fourth one is NB. For the grayhole class, RF has achieved the highest performances among other models (accuracy of 99.92%, precision of 92.81%, recall of 99.04%, and F-score of 95.82%). In contrast, NB has done the worst performances (accuracy of 91.47%, precision of 7.39%, recall of 74.91%, and F-score of 13.44%). Like blackhole class, RF has achieved the highest performances among other models (accuracy of 99.86%, precision of 98.24%, recall of 98.11%, and F-score of 98.17%). In contrast, NB has done the worst performances (accuracy of 93.45%, precision of 23.26%, recall of 29.6%, and F-score of 26.04%). Regarding scheduling class, RF has achieved the highest performances among other models (accuracy of 99.69%, precision of 99.82%, recall of 99.83%, and F-score of 99.83%). In contrast, NB has done the worst performances (accuracy of 84.61%, precision of 95.14%, recall of 87.52%, and F-score of 91.17%). Flooding class like the other classes, RF has achieved the highest performances among other models (accuracy of 99.84%, precision of 97.84%, recall of 92.96%, and F-score of 95.33%). In contrast, NB has done the worst performances (accuracy of 99.23%, precision of 95.83%, recall of 59.07%, and F-score of 73.08%). Based on these results, the RF model using normal class is the best performing model concerning other models, while NB using Scheduling class is the worst performing model.

(2) Testing Results. Table 10 shows the performance of the machine learning models using the unseen testing WSN-DS dataset. For the normal class, RF is the highest performance model (accuracy of 99.94%, precision of 98.45%, recall of 99.48%, and F-score of 98.96%). DT, KNN, and NB models have achieved the second, third, and fourth ranks on the average of accuracy over unseen data by 99.93%, 99.46%, and 98.20%, respectively. For the grayhole class, DT is the highest performance model (accuracy of 99.95%, precision of

Table 12: The performance of deep learning for WSN dataset using LSTM and GRU with all features.

Model	Evaluation metric	Cross-validation performance					Testing performance				
		Normal	Grayhole	Blackhole	Scheduling	Flooding	Normal	Grayhole	Blackhole	Scheduling	Flooding
LSTM with one layer	TN	99.99	99.98	100	99.99	100	98.71	99.90	99.35	92.90	99.92
	FP	0.01	0.02	0	0.01	0	1.29	0.10	0.65	7.10	0.08
	FN	0	0.05	0.26	3.19	23.33	7.05	6.64	43.77	0.63	10.66
	Accuracy	99.99	99.98	100	99.98	100	98.55	99.84	97.67	98.77	99.73
	Precision	100	99.92	100	96.76	95	66.47	88.95	77.93	99.28	95.25
	Recall	100	99.95	99.74	96.81	76.67	92.95	93.36	56.23	99.37	89.34
	F-score	100	99.93	99.87	96.75	83.67	77.51	91.10	65.33	99.32	92.20
LSTM with two layer	TN	98.65	99.86	99.41	91.73	99.89	98.66	99.86	99.50	93.73	99.89
	FP	1.35	0.14	0.59	8.27	0.11	1.34	0.14	0.50	6.27	0.11
	FN	8.52	7.36	47.46	0.59	12.21	8.32	3.50	43.66	0.50	11.20
	Accuracy	98.46	99.79	97.58	98.7	99.67	98.47	99.83	97.82	98.96	99.69
	Precision	65.15	85.19	78.39	99.16	93.3	65.31	86.19	82.01	99.36	93.53
	Recall	91.48	92.64	52.54	99.41	87.79	91.68	96.50	56.34	99.50	88.80
	F-score	76.07	88.74	62.87	99.29	90.44	76.28	91.05	66.80	99.43	91.10
GRU with one layer	TN	98.67	99.87	99.3	91.82	99.88	98.78	99.89	99.24	93.29	99.93
	FP	1.33	0.13	0.7	8.18	0.12	1.22	0.11	0.76	6.71	0.07
	FN	9.61	9.42	46.84	0.65	11.89	12.86	6.76	40.94	0.57	10.66
	Accuracy	98.45	99.79	97.5	98.66	99.68	98.47	99.83	97.68	98.86	99.75
	Precision	65.32	86.5	75.67	99.17	93.17	66.27	88.43	75.93	99.32	96.11
	Recall	90.39	90.58	53.16	99.35	88.11	87.14	93.24	59.06	99.43	89.34
	F-score	75.7	88.47	62.23	99.26	90.56	75.29	90.77	66.44	99.37	92.60
GRU with two layer	TN	98.63	99.85	99.46	94.05	99.9	98.79	99.87	99.31	96.45	99.91
	FP	1.37	0.15	0.54	5.95	0.1	1.21	0.13	0.69	3.55	0.09
	FN	8.08	3.75	43.34	0.56	11.31	15.21	2.78	34.06	0.48	10.36
	Accuracy	98.45	99.82	97.79	98.95	99.7	98.41	99.85	98.01	99.23	99.73
	Precision	65.02	85.52	80.9	99.4	94.16	65.88	86.93	79.48	99.64	94.96
	Recall	91.92	96.25	56.66	99.44	88.69	84.79	97.22	65.94	99.52	89.64
	F-score	76.11	90.54	66.43	99.42	91.34	74.15	91.79	72.08	99.58	92.22

97.32%, recall of 96.89%, and F-score of 97.1%). At the same time, NB is the worst performing model (accuracy of 92.95%, precision of 8.51%, recall of 80.00%, and F-score of 15.38%). Similar to blackhole class, RF is the highest performance model (accuracy of 99.87%, precision of 98.3%, recall of 98.47%, and F-score of 98.39%). Regarding scheduling and flooding classes, RF obtained the highest performance by accuracy of 99.74% and 99.89%, respectively. However, NB is the worst performing model for both classes, including accuracy of 84.10% for scheduling and 90.15% for flooding. Yet, the NB model using scheduling classes has the lowest accuracy performance among all models and classes for unseen data in terms of accuracy, 84.65%.

4.2.3. Deep Learning Using All Features

(1) Cross-Validation Results. This section discusses the 10-fold CV results of four deep learning models (LSTM and GRU, using one and two layers) over the WSN-DS dataset with all features, as shown in Table 12. As shown in the results, for the normal class, LSTM using one layer model has achieved the highest performances among other models and other classes (accuracy of 99.99%, precision of 100%, recall of 100%, and F-score of 100%). LSTM using two layers' model has achieved the second one, and GRU with one layer and two layers' models have almost similar performances which achieved the third one. For the grayhole class, LSTM using one layer model has achieved the highest performances among other models (accuracy of 99.98%, precision of 99.92%, recall of 99.95%, and F-score of 99.93%). In contrast, GRU with two layers has done the second one, and LSTM using one layer and GRU with two layers has done similar performances ranked as a third one. Like blackhole class, LSTM using one layer model has achieved the highest performances among other models and other classes (accuracy of 100%, precision of 100%, recall of 99.74%, and F-score of 99.87%). In contrast, GRU with one layer has done the worst performances (accuracy of 97.5%, precision of 75.67%, recall of 53.16%, and F-score of 62.23%). Regarding scheduling class, LSTM using one layer has achieved the highest performances among other models (accuracy of 99.98%, precision of 96.76%, recall of 96.81%, and F-score of 96.75%). Flooding class like the other classes, LSTM using one layer has achieved the highest performances among other models (accuracy of 100% and precision of 95%), while other models include LSTM using two layers, and GRU using one layer and two layers has achieved approximated performances in terms of accuracy such as 99.67%, 99.68%, and 99.7%, respectively. Based on these results, LSTM using one layer model for blackhole and flooding classes is the best performing model with respect to other models, while GRU using one and two layers for normal class is the worst performing model.

(2) Testing Results. Table 12 shows the performance of the deep learning models using the unseen testing WSN-DS dataset. As shown in the results, for the normal class, LSTM

Table 13: The performance of deep learning for WSN dataset using LSTM and GRU with selected features.

Model	Evaluation metric	Cross-validation performance					Testing performance				
		Normal	Grayhole	Blackhole	Scheduling	Flooding	Normal	Grayhole	Blackhole	Scheduling	Flooding
LSTM with one layer	TN	98.59	99.91	99.48	96	99.93	98.50	99.92	99.73	96.61	99.99
	FP	1.41	0.09	0.52	4	0.07	1.50	0.08	0.27	3.39	0.01
	FN	6.82	8.5	39.19	0.47	9.9	1.63	5.19	40.12	0.33	9.22
	Accuracy	98.45	99.84	97.98	99.21	99.76	98.50	99.88	98.18	99.39	99.83
	Precision	64.84	90.6	83.37	99.59	96.08	64.42	91.49	89.92	99.66	99.47
	Recall	93.18	91.5	60.81	99.53	90.1	98.37	94.81	59.88	99.67	90.78
	F-score	76.2	91.02	69.98	99.56	92.98	77.85	93.12	71.89	99.66	94.93
LSTM with two layer	TN	98.41	99.9	99.17	94.62	99.92	98.48	99.91	99.25	94.51	99.88
	FP	1.59	0.1	0.83	5.38	0.08	1.52	0.09	0.75	5.49	0.12
	FN	3.3	13.3	42.91	0.83	21.43	5.10	4.11	42.50	0.80	19.88
	Accuracy	98.37	99.79	97.53	98.75	99.54	98.38	99.87	97.62	98.76	99.53
	Precision	62.68	89.59	73.95	99.45	94.85	63.25	90.43	75.66	99.44	92.62
	Recall	96.7	86.7	57.09	99.17	78.57	94.90	95.89	57.50	99.20	80.12
	F-score	76.04	87.68	64.37	99.31	85.85	75.91	93.08	65.34	99.32	85.92
GRU with one layer	TN	99.08	99.91	99.3	97.85	99.99	98.62	99.93	99.78	97.91	99.99
	FP	0.92	0.09	0.7	2.15	0.01	1.38	0.07	0.22	2.09	0.01
	FN	21.71	3.26	23.88	0.21	7.73	3.42	6.28	34.23	0.22	7.89
	Accuracy	98.52	99.88	98.4	99.61	99.85	98.57	99.87	98.46	99.60	99.85
	Precision	72.69	90.36	83.72	99.78	99.27	65.87	91.94	92.49	99.79	99.61
	Recall	78.29	96.74	76.12	99.79	92.27	96.58	93.72	65.77	99.78	92.11
	F-score	72.62	93.43	78.58	99.78	95.64	78.32	92.82	76.87	99.78	95.7
GRU with two layer	TN	98.53	99.93	99.25	89.1	99.81	98.65	99.91	99.49	94.11	99.82
	FP	1.47	0.07	0.75	10.9	0.19	1.35	0.09	0.51	5.89	0.18
	FN	5.85	39.26	53.04	0.54	19.99	7.44	9.78	42.09	0.49	13.73
	Accuracy	98.41	99.59	97.21	98.5	99.46	98.49	99.83	97.87	99.01	99.58
	Precision	63.87	88.58	72.01	98.9	88.16	65.44	90.33	82.06	99.40	89.44
	Recall	94.15	60.74	46.96	99.46	80.01	92.56	90.22	57.91	99.51	86.27
	F-score	76.09	79.73	56.78	99.18	83.85	76.67	90.27	67.90	99.46	87.83

using one layer model has achieved the highest performances among other models and other classes (accuracy of 98.55%, precision of 66.47%, recall of 92.95%, and F-score of 77.51%). LSTM using two layers and GRU has one layer models have achieved the second one, and GRU with two layers has achieved the third one. For the grayhole class, GRU using two layers' model has achieved the highest performances among other models (accuracy of 99.85%, precision of 86.93%, recall of 97.22%, and F-score of 91.79%). LSTM using one layer has performed the second one, LSTM using two layers and GRU with one layer have done the third one, and LSTM using one layer and GRU with two layers are ranked as a third one. Similar to blackhole class, GRU using two layers model has achieved the highest performances among other models and other classes (accuracy of 98.01%, precision of 79.48%, recall of 65.94%, and F-score 72.08%), while LSTM with one layer has done the worst performances (accuracy of 97.67%, precision of 77.93%, recall of 56.23%, and F-score 65.33%). Regarding scheduling class, GRU using two layers has achieved the highest performances among other models (accuracy of 99.23%, precision of 99.64%, recall of 99.52%, and F-score of 99.58%). For flooding class, GRU using one layer has achieved the highest performances among other models (accuracy of 99.75%, precision of 96.11%, recall of 89.34%, and F-score of 92.60%). Based on these results, GRU using two layers' model for grayhole is the best performing model with respect to other models, while GRU using one layer for blackhole class is the worst performing model.

4.2.4. Deep Learning Using Selected Features

(1) Cross-Validation Results. As shown in the result in Table 13, four deep learning models were over the WSN-DS dataset with selected features, and for the normal class, GRU using one layer model has achieved the highest performances based on its accuracy among other models and other classes (accuracy of 98.52%, precision of 72.69%, recall of 78.29%, and F-score of 72.62%). LSTM using one layer, GRU with two layers, and LSTM using two layers have been ranked as the second, third, and fourth models, respectively, based on their accuracy. For the grayhole class, GRU using one layer model has achieved the highest performances among other models (accuracy of 99.88%, precision of 90.36%, recall of 96.74%, and F-score of 93.43%). In contrast, LSTM with one layer, GRU with one layer, and GRU with two layers have recorded the second, third, and fourth models based on their accuracy. Like the blackhole class, GRU using one layer model has achieved the highest performances among other models and other classes (accuracy of 98.4%, precision of 83.72%, recall of 76.12%, and F-score of 78.58%). In contrast, GRU with two-layer has done the worst performances (accuracy of 97.21%, precision of 72.01%, recall of 46.96%, and F-score of 56.78%).

Regarding scheduling class, GRU using one layer has achieved the highest performances among other models (accuracy of 99.61%, precision of 99.78%, recall of 99.79%, and F-score of 99.78%). Flooding class like the other classes, GRU using one layer has achieved the highest performances

among other models (accuracy of 99.85%, precision of 99.27%, recall of 92.27%, and F-score of 95.64%). Based on these results, GRU using one layer model for grayhole is the best performing model with respect to other models, while GRU using two layers for the blackhole class is the worst performing model.

(2) Testing Results. Table 13 shows the performance of the machine learning models using the unseen testing WSN-DS dataset. As shown in the results, for the normal class, GRU using one layer model has achieved the highest performances among other models and other classes (accuracy of 98.57%, precision of 65.87%, recall of 96.58%, and F-score of 78.32%). However, LSTM using two layers' model has recorded the worst performances among other models and other classes (accuracy of 98.38%, precision of 63.25%, recall of 94.90%, and F-score of 75.91%). For the grayhole class, LSTM using one layer model has achieved the highest performances among other models (accuracy of 99.88%, precision of 91.49%, recall of 94.81%, and F-score of 93.12%). GRU using two layers' model has recorded the worst performances among other models and other classes (accuracy of 99.83%, precision of 90.33%, recall of 90.22%, and F-score of 90.27%). Like the blackhole class, GRU using one layer model has achieved the highest performances among other models and other classes (accuracy of 98.46%, precision of 92.49%, recall of 65.77%, and F-score of 76.87%). In contrast, LSTM with two layers has the worst performances (accuracy of 97.62%, precision of 75.66%, recall of 57.50%, and F-score of 65.34%). Regarding the scheduling class, GRU using one layer model has achieved the highest performances among other models and other classes (accuracy of 99.60%, precision of 99.79%, recall of 99.78%, and F-score 99.78%). In contrast, LSTM with two layers has done the worst performances (accuracy of 98.76%, precision of 99.44%, recall of 99.20%, and F-score 99.32%). Flooding class like the other classes, GRU using one layer model has achieved the highest performances among different models and other classes (accuracy of 99.85%, precision of 99.61%, recall of 92.11%, and F-score of 95.7%). In contrast, LSTM with two layers has the worst performances (accuracy of 99.53%, precision of 92.62%, recall of 80.12%, and F-score of 85.92%). Based on these results, LSTM using one layer model for grayhole is the best performing model with respect to other models, while LSTM using one layer for the blackhole class is the worst performing model.

4.3. KDD Dataset. In this section, the results of applying four machine learning models (DT, KNN, RF, and NB) and four deep learning models (LSTM and GRU, using one and two layers), including cross-validation results and testing results, are described. Each machine learning model and deep learning model performance is discussed using full features and selected features to classify five classes of attack types including *DOS, R2L, U2R, Probe,* and *Normal*. All the positive and negative rates results of the cross-validation and testing performances which are used to compute accuracy, precision, recall, and F-score matrices are presented in Tables 14–17.

4.3.1. Regular Machine Learning Using All Features

(1) Cross-Validation Results. This section discusses the 10-fold CV results of four machine learning models (DT, KNN, RF, and NB) over the KDD dataset with all features, as shown in Table 14. As shown in the results, for the Dos class, RF has achieved the highest performances among other models and other classes (accuracy of 100%, precision of 100%, recall of 100%, and F-score of 100%). The DT model has achieved the second one and KNN is the third one. NB has reached the lowest performances (accuracy of 92.44%, precision of 93.93%, recall of 96.93%, and F-score 95.26%). For the normal class, RF has achieved the highest performances among other models (accuracy of 99.98%, precision of 92.9%, recall of 99.98%, and F-score of 95.94%). In contrast, NB has the worst performances (accuracy of 92.72%, precision of 99%, recall of 59.7%, and F-score 74.48%). Like the probe class, RF has achieved the highest performances among other models (accuracy of 100%, precision of 99.97%, recall of 99.59%, and F-score of 99.78%). In contrast, NB has the worst performances (accuracy of 96.74%, precision of 0.3%, recall of 6.96%, and F-score of 5.7%). Regarding R2l class, RF has achieved the highest performances among other models (accuracy of 99.99%, precision of 99.36%, recall of 96.77%, and F-score of 98.04%). In contrast, NB has the worst performances (accuracy of 98.77%, precision of 0.25%, recall of 1.35%, and F-score of 4.22%). U2r class classes, RF has achieved the highest performance (accuracy of 100%, precision of 93.75%, and F-score of 77.8%), while DT has achieved the highest recall of 96.99%. KNN and NB have the worst performances. Based on these results, the RF model using the DoS class is the best performing model with respect to other models.

(2) Testing Results. Table 14 shows the performance of the machine learning models using the unseen testing KDD dataset. For the RF model, DOS class has achieved the highest accuracy among other models and classes (accuracy of 100%, precision of 100%, recall of 100%, and F-score of 100%). However, NB has the DOS class's worst performances (accuracy of 94.17%, precision of 93.82%, recall of 99.37%, and F-score of 96.52%). For the normal class, RF is the highest performance model (accuracy of 99.98%, precision of 99.89%, recall of 99.99%, and F-score of 98.94%). DT, KNN, and NB models have achieved the second, third, and fourth ranks based on accuracy over unseen data by 99.97%, 99.85%, and 92.65%, respectively. Similar to the probe class, DT is the highest performance model (accuracy of 99.99%, precision of 100%, recall of 99.28%, and F-score of 96.64%). RF, KNN, and NB models have achieved the second, third, and fourth ranks on the average of accuracy over unseen data by 99.99%, 99.85%, and 98.65%, respectively. Regarding R2l and U2r classes, RF obtained the highest performance, while NB is the worst performing model for both classes. However, the NB model using

Table 14: The performance of machine learning for KDD dataset using all features.

Models	Evaluation metric	Cross-validation performance					Testing performance				
		DOS	Normal	Probe	R2L	U2R	DOS	Normal	Probe	R2L	U2R
DT	TNR	99.97	99.98	99.99	100	100	99.95	99.98	100	99.99	100
	FPR	0.03	0.02	0.01	0	0	0.05	0.02	0	0.01	0
	FNR	0	0.08	0.68	3.01	25.83	0	0.07	0.72	5.63	50
	Accuracy	99.99	99.97	99.99	99.99	99.99	99.99	99.97	99.99	99.98	99.99
	Precision	99.99	99.93	99.27	98.1	74.17	99.99	99.89	100	97.1	66.67
	Recall	100	99.92	99.32	96.99	74.17	100	99.93	99.28	94.37	50
	F-score	100	99.92	99.29	97.53	70.6	99.99	99.91	99.64	95.71	57.14
KNN	TNR	99.61	99.94	99.98	99.99	100	99.55	99.89	99.96	99.99	100.00
	FPR	0.39	0.06	0.02	0.01	0	0.45	0.11	0.04	0.01	0.00
	FNR	0.01	0.18	9.76	5.65	100	0.04	0.29	12.27	8.45	100.00
	Accuracy	99.91	99.92	99.9	99.98	99.99	99.89	99.85	99.85	99.97	99.99
	Precision	99.91	99.71	97.82	96.31	0	99.90	99.47	95.67	95.59	0
	Recall	99.99	99.82	90.24	94.35	0	99.96	99.71	87.73	91.55	0.00
	F-score	99.95	99.76	93.87	95.28	0	99.93	99.59	91.53	93.53	0
RF	TNR	99.99	99.98	100	100	100	100.0	99.98	100.00	100.00	100.00
	FPR	0.01	0.02	0	0	0	0.0	0.02	0.00	0.00	0.00
	FNR	0	0.02	0.41	3.23	40	0.0	0.01	0.80	3.52	41.67
	Accuracy	100	99.98	100	99.99	100	100.0	99.98	99.99	99.99	100.00
	Precision	100	99.9	99.97	99.36	93.75	100.0	99.89	99.82	99.70	87.50
	Recall	100	99.98	99.59	96.77	60	100.0	99.99	99.20	96.48	58.33
	F-score	100	99.94	99.78	98.04	77.8	100.0	99.94	99.51	98.06	70.00
NB	TNR	73.23	99.87	97.52	99.02	97.88	71.71	99.86	99.54	99.10	97.89
	FPR	26.77	0.13	2.48	0.98	2.12	28.29	0.14	0.46	0.90	2.11
	FNR	3.07	40.3	93.04	98.65	45	0.63	41.05	100.00	100.00	75.00
	Accuracy	92.44	92.72	96.74	98.77	97.88	94.17	92.65	98.65	98.87	97.88
	Precision	93.93	99	0.3	0.25	0.24	93.82	98.92	0.00	0.00	0.15
	Recall	96.93	59.7	6.96	1.35	55	99.37	58.95	0.00	0.00	25.00
	F-score	95.26	74.48	5.7	4.22	0.6	96.52	73.87	0	0	0.31

normal classes has the lowest accuracy performance among all models and classes for unseen data in terms of accuracy, 92.65%.

4.3.2. Regular Machine Learning Using Selected Features

(1) Cross-Validation Results. This section discusses the 10-fold CV results of four machine learning models (DT, KNN, RF, and NB) over the KDD dataset with all features, as shown in Table 15. As shown in the results, for the DOS class, RF and DT have achieved the highest performances among other models and other classes (accuracy of 99.72%, precision of 99.71%, recall of 99.94%, and F-score of 99.83%). The KNN model has achieved the second one. NB has achieved the lowest performances (accuracy of 65.13%, precision of 80.44%, recall of 75.3%, and F-score of 77.78%). For the normal class, RF and DT have achieved the highest performances among other models (accuracy of 99.93%, precision of 99.76%, recall of 99.86%, and F-score of 99.8%), while NB has the worst performances (accuracy of 82.46%, precision of 66.47%, recall of 2.65%, and F-score of 5.08%). The probe class, RF has achieved the highest performances among other models (accuracy of 99.99%, precision of 99.74%, and F-score of 98.3%), while DT has achieved the highest recall of 97.6%. NB has the worst performances (accuracy of 96.87%, precision of 0.29%, recall of 7.03%, and F-score of 5.61%). Regarding R2l class, RF has achieved the highest performances among other models (accuracy of 100%, precision of 87.04%, recall of 60%, and F-score of 71.85%), while NB has the worst performances (accuracy of 99.24%, precision of 0.48%, recall of 1.36%, and F-score 3.56%). U2R classes, RF and DT, have achieved similar the highest performance. NB has the worst performances (accuracy of 79.47%, precision of 0.09%, recall of 79.17, and F-score of 0.19). Based on these results, RF and DT models for each class have the best performing model with respect to other models, while NB has the worst performances.

(2) Testing Results. Table 15 shows the performance of the machine learning models using the unseen testing KDD dataset. For the RF and DT model, DOS class has achieved the highest accuracy among other models and classes (accuracy of 99.74%, precision of 99.74%, recall of 99.94%, and F-score of 99.84%). However, NB has the DOS class's worst performances (accuracy of 64.87%, precision of 80.27%, recall of 75.10%, and F-score of 77.60%). For the normal class, DT and RF are the highest performance model (accuracy of 99.93%, precision of 99.77%, recall of 99.82%, and F-score of 99.80%). KNN and NB models have achieved the second and third ranks based on accuracy over unseen data by 99.83% and 82.48%, respectively. Similar to the probe class, RF is the highest performance model (accuracy of 99.78%, precision of 96.15%, recall of 77.69%, and F-score of 85.94%).

Table 15: The performance of machine learning for KDD dataset using selected features.

Models	Evaluation metric	Cross-validation performance					Testing performance				
		Dos	Normal	Probe	R2L	U2R	DOS	Normal	Probe	R2L	U2R
DT	TNR	98.75	99.95	99.98	100	100	98.91	99.95	99.97	99.99	100.00
	FPR	1.25	0.05	0.02	0	0	1.09	0.05	0.03	0.01	0.00
	FNR	0.06	0.17	25.37	2.4	50.83	0.06	0.18	22.76	2.05	75.00
	Accuracy	99.72	99.93	99.76	99.99	99.99	99.74	99.93	99.78	99.99	99.99
	Precision	99.71	99.76	96.75	98.43	75.93	99.74	99.77	95.60	97.95	42.86
	Recall	99.94	99.86	74.63	97.6	49.17	99.94	99.82	77.24	97.95	25.00
	F-score	99.83	99.8	84.25	98	57.78	99.84	99.80	85.45	97.95	31.58
KNN	TNR	98.55	99.87	99.97	99.99	100	98.72	99.86	99.98	99.99	100.00
	FPR	1.45	0.13	0.03	0.01	0	1.28	0.14	0.02	0.01	0.00
	FNR	0.07	0.38	27.65	16.11	100	0.07	0.33	25.07	19.94	100.00
	Accuracy	99.67	99.83	99.73	99.95	99.99	99.70	99.83	99.76	99.94	99.99
	Precision	99.66	99.41	95.89	96.8	0.0	99.70	99.37	96.45	94.79	0.0
	Recall	99.93	99.62	72.35	83.89	0	99.93	99.67	74.93	80.06	0.00
	F-score	99.8	99.51	82.46	89.81	0.0	99.81	99.52	84.34	86.80	0.0
RF	TNR	98.76	99.94	99.98	100	100	98.91	99.95	99.97	100.00	100.00
	FPR	1.24	0.06	0.02	0	0	1.09	0.05	0.03	0.00	0.00
	FNR	0.06	0.14	25.29	3.08	40	0.06	0.13	22.31	3.81	58.33
	Accuracy	99.72	99.93	99.76	99.99	100	99.74	99.94	99.78	99.99	99.99
	Precision	99.71	99.76	96.66	99.74	87.04	99.74	99.77	96.15	99.39	71.43
	Recall	99.94	99.86	74.71	96.92	60	99.94	99.87	77.69	96.19	41.67
	F-score	99.83	99.8	84.26	98.3	71.85	99.84	99.82	85.94	97.76	52.63
NB	TNR	21.6	99.75	97.65	99.49	79.47	21.27	99.72	99.72	99.52	77.36
	FPR	78.4	0.25	2.35	0.51	20.53	78.73	0.28	0.28	0.48	22.64
	FNR	24.7	97.35	92.97	98.64	20.83	24.90	96.72	100.00	100.00	8.33
	Accuracy	65.13	82.46	96.87	99.24	79.47	64.87	82.48	98.87	99.27	77.36
	Precision	80.44	66.47	0.29	0.48	0.09	80.27	71.77	0.00	0.00	0.04
	Recall	75.3	2.65	7.03	1.36	79.17	75.10	3.28	0.00	0.00	91.67
	F-score	77.78	5.08	5.61	3.56	0.19	77.60	6.27	0	0	0.07

DT, KNN, and NB models have achieved the second, third, and fourth ranks on the average of accuracy over unseen data by 99.78%, 99.76%, and 98.87%, respectively. Regarding R2L and U2R classes, RF obtained the highest performance, while NB is the worst performing model for both classes.

4.3.3. Deep Learning Using All Features

(1) Cross-Validation Results. As shown in the result in Table 16, four deep learning models were over the KDD dataset with selected features; for the DOS class, LSTM using two layers' model has achieved the highest performances based on its accuracy among other models and other classes (accuracy of 99.83%, precision of 99.97%, recall of 99.82%, and F-score of 99.9%). LSTM using one layer, GRU with one layer and two layers, and LSTM using one layer has been ranked as the second, third, and fourth models, respectively, based on their accuracy. For the normal class, LSTM using two layers' model has achieved the highest performances among other models (accuracy of 99.73%, precision of 98.85%, recall of 99.65%, and F-score of 99.25%). In contrast, LSTM with one layer has achieved the lowest performance (accuracy of 99.79%, precision of 86.16%, recall of 90.36%, and F-score of 88.2%). GRU with one layer and GRU with two layers have recorded the second, third, and accuracy. Like the probe class, LSTM using two layers' model has achieved the highest performances among other models and other classes (accuracy of 99.97%, precision of 99.58%, recall of 97.34%, and F-score of 98.44%). In contrast, LSTM with one layer has the worst performances (accuracy of 97.47%, precision of 74.48%, recall of 53.84%, and F-score of 62.28%). Regarding R2l class, GRU using one layer has achieved the highest performances among other models (accuracy of 99.92%, precision of 84.57%, recall of 85.72%, and F-score of 85.05%). U2R class like the other classes, GRU using one layer has achieved the highest performances among other models accuracy of 99.68%, precision of 93.07%, recall of 88.33%, and F-score of 90.62%). Based on these results, LSTM using two layers' model for DOS is the best performing model with respect to other models.

(2) Testing Results. As shown in the result in Table 16, four deep learning models were over the KDD dataset with selected features; for the DOS class, LSTM using two layers' model has achieved the highest performances based on its accuracy among other models and other classes (accuracy of 99.83%, precision of 99.97%, recall of 99.82%, and F-score of 99.9%). LSTM using one layer, GRU with one layer and two layers, and LSTM using one layer has been ranked as the second, third, and fourth models, respectively, based on their accuracy. For the normal class, LSTM using two layers' model has achieved the highest performances among other models (accuracy of 99.73%, precision of 98.85%, recall of

Table 16: The performance of deep learning for KDD dataset using LSTM and GRU with all features.

Model	Evaluation metric	Cross-validation performance					Testing performance				
		DOS	Normal	Probe	R2L	U2R	DOS	Normal	Probe	R2L	U2R
LSTM with one layer	TNR	98.7	99.87	99.24	91.83	99.88	99.98	99.96	100.00	99.99	100.00
	FPR	1.3	0.13	0.76	8.17	0.12	0.02	0.04	0.00	0.01	0.00
	FNR	10.42	9.64	46.16	0.68	11.67	0.01	0.06	1.24	7.62	66.67
	Accuracy	98.46	99.79	97.47	98.63	99.68	99.99	99.95	99.99	99.97	99.99
	Precision	65.61	86.16	74.48	99.17	93.07	100.00	99.80	99.55	97.22	66.67
	Recall	89.58	90.36	53.84	99.32	88.33	99.99	99.94	98.76	92.38	33.33
	F-score	75.6	88.2	62.28	99.24	90.62	100.00	99.87	99.15	94.74	44.44
LSTM with two layer	TNR	99.89	99.75	100	99.95	100	99.92	99.93	99.99	99.96	100.00
	FPR	0.11	0.25	0	0.05	0	0.08	0.07	0.01	0.04	0.00
	FNR	0.18	0.35	2.66	15.57	100	0.01	0.29	2.22	10.85	100.00
	Accuracy	99.83	99.73	99.97	99.91	99.99	99.98	99.89	99.98	99.93	99.99
	Precision	99.97	98.85	99.58	82.2	0	99.98	99.70	99.37	84.92	0
	Recall	99.82	99.65	97.34	84.43	0	99.99	99.71	97.78	89.15	0.00
	F-score	99.9	99.25	98.44	83.27	0	99.99	99.70	98.57	86.98	0
GRU with one layer	TNR	99.88	99.56	100	99.96	100	99.92	99.90	100.00	99.97	100.00
	FPR	0.12	0.44	0	0.04	0	0.08	0.10	0.00	0.03	0.00
	FNR	0.37	0.32	3.07	14.28	100	0.04	0.25	2.93	11.73	100.00
	Accuracy	99.67	99.58	99.97	99.92	99.99	99.95	99.87	99.97	99.93	99.99
	Precision	99.97	98	99.46	84.57	0	99.98	99.53	99.54	86.74	0
	Recall	99.63	99.68	96.93	85.72	0	99.96	99.75	97.07	88.27	0.00
	F-score	99.8	98.83	98.17	85.05	0	99.97	99.64	98.29	87.50	0
GRU with two layer	TNR	99.89	99.53	99.99	99.96	100	99.96	99.87	100.00	99.97	100.00
	FPR	0.11	0.47	0.01	0.04	0	0.04	0.13	0.00	0.03	0.00
	FNR	0.38	0.28	2.62	23.67	100	0.05	0.19	2.67	14.96	100.00
	Accuracy	99.67	99.56	99.97	99.9	99.99	99.95	99.86	99.97	99.93	99.99
	Precision	99.98	97.87	99.2	86.37	0	99.99	99.43	99.64	87.35	0
	Recall	99.62	99.72	97.38	76.33	0	99.95	99.81	97.33	85.04	0.00
	F-score	99.8	98.78	98.28	76.73	0	99.97	99.62	98.47	86.18	0

99.65%, and F-score of 99.25%). In contrast, LSTM with one layer has achieved the lowest performance accuracy of 99.79%, precision of 86.16%, recall of 90.36% and F-score of 88.2%). GRU with one layer and GRU with two layers have recorded the second and third and accuracy. Like the probe class, LSTM using two layers' model has achieved the highest performances among other models and other classes (accuracy of 99.97%, precision of 99.58%, recall of 97.34%, and F-score of 98.44%). Table 16 shows the performance of the deep learning models using the unseen testing KDD dataset. As shown in the results, for the DOS class, LSTM using one layer model has achieved the highest performances among other models and other classes (accuracy of 99.99%, precision of 100%, recall of 99.99%, and F-score of 100%). LSTM using two layers' model and GRU models have similar performance. For the normal class, LSTM using one layer model has achieved the highest performances among other models (accuracy of 99.95%, precision of 99.80%, recall of 99.94%, and F-score of 99.87%). LSTM with two layers and GRU using one layer and two layers model have recorded the similar performances. Like the probe class, LSTM using one layer model has achieved the highest performances among other models and other classes (accuracy of 99.99%, precision of 99.55%, recall of 98.76%, and F-score of 99.15%). Regarding the scheduling class, GRU using one layer model has achieved the highest performances among other models and other classes (accuracy of 99.60%, precision of 99.79%, recall of 99.78% and F-score99.78%). In contrast, LSTM with two layers has done the worst performances (accuracy of 98.76%, precision of 99.44%, recall of 99.20%, and F-score of 99.32%). Flooding class like the other classes, GRU using one layer model has achieved the highest performances among different models and other classes (accuracy of 99.85%, precision of 99.61%, recall of 92.11%, and F-score of 95.7%). In contrast, LSTM with two layers has the worst performances (accuracy of 99.53%, precision of 92.62%, recall of 80.12%, and F-score of 85.92%). Based on these results, LSTM using one layer model for grayhole is the best performing model for other models, while LSTM using one layer for the blackhole class is the worst performing model.

4.3.4. Deep Learning Using Selected Features

(1) Cross-Validation Results. As shown in the result, in Table 17, four deep learning models were over the KDD dataset with selected features; for the DOS class, GRU using two layers, model has achieved the highest performances based on its accuracy among other models and other classes (accuracy of 96.14%, precision of 96.13%, recall of 99.23%, and F-score of 97.65%). LSTM using one layer model has achieved the lowest performances based on its accuracy among other models and other classes (accuracy of 92.98%, precision of 92.91%, recall of 99.05%, and F-score of 95.84%).

Table 17: The performance of deep learning for KDD dataset using LSTM and GRU with selected features.

Model	Evaluation metric	Cross-validation performance					Testing performance				
		DOS	Normal	Probe	R2L	U2R	DOS	Normal	Probe	R2L	U2R
LSTM with one layer	TNR	66.96	98.75	100	99.99	100	78.06	98.50	100.00	99.99	100.00
	FPR	33.04	1.25	0	0.01	0	21.94	1.50	0.00	0.01	0.00
	FNR	0.95	30.63	100	79.18	100	1.06	19.57	100.00	77.13	100.00
	Accuracy	92.98	93.51	99.13	99.79	99.99	94.98	95.27	99.15	99.79	99.99
	Precision	92.91	92.35	0	84.04	0.0	95.06	92.09	0.0	82.11	0.0
	Recall	99.05	69.37	0	20.82	0	98.94	80.43	0.00	22.87	0.00
	F-score	95.84	77.59	0.0	39.48	0.0	96.96	85.87	0.0	35.78	0.0
LSTM with two layer	TNR	81.53	98.37	99.99	100	100	85.83	98.37	100.00	99.99	100.00
	FPR	18.47	1.63	0.01	0	0	14.17	1.63	0.00	0.01	0.00
	FNR	1.24	15.93	87.69	94.94	100	1.40	11.50	76.09	76.83	100.00
	Accuracy	95.49	95.82	99.22	99.76	99.99	96.17	96.60	99.35	99.79	99.99
	Precision	95.82	91.78	84.58	91.37	0.0	96.74	92.19	98.18	86.81	0.0
	Recall	98.76	84.07	12.31	5.06	0	98.60	88.50	23.91	23.17	0.00
	F-score	97.26	87.71	33.94	39.52	0.0	97.66	90.31	38.46	36.57	0.0
GRU with one layer	TNR	78.73	99.22	100	100	100	79.00	99.19	100.00	100.00	100.00
	FPR	21.27	0.78	0	0	0	21.00	0.81	0.00	0.00	0.00
	FN	0.36	18.26	100	100	100	0.36	18.15	100.00	100.00	100.00
	Accuracy	95.68	96.11	99.13	99.75	99.99	95.72	96.09	99.15	99.74	99.99
	Precision	95.25	95.77	0.0	0.0	0.0	95.29	95.66	0.0	0.0	0.0
	Recall	99.64	81.74	0	0	0	99.64	81.85	0.00	0.00	0.00
	F-score	97.4	88.2	0.0	0.0	0.0	97.42	88.22	0.0	0.0	0.0
GRU with two layer	TNR	82.9	98.94	99.99	99.98	100	83.25	99.34	99.98	99.99	100.00
	FPR	17.1	1.06	0.01	0.02	0	16.75	0.66	0.02	0.01	0.00
	FNR	0.77	14.4	86.83	73.34	100	0.46	14.11	76.09	76.54	100.00
	Accuracy	96.14	96.56	99.24	99.8	99.99	96.45	96.93	99.34	99.79	99.99
	Precision	96.13	94.62	91.67	80.4	0.0	96.20	96.59	93.08	83.33	0.0
	Recall	99.23	85.6	13.17	26.66	0	99.54	85.89	23.91	23.46	0.00
	F-score	97.65	89.88	35.28	39.92	0.0	97.84	90.92	38.05	36.61	0.0

For the normal class, GRU using two layers' model has achieved the highest performances among other models (accuracy of 96.56%, precision of 94.62%, recall of 85.6%, and F-score 89.88%). In contrast, LSTM with one layer has reached the lowest performance accuracy of 93.51%, precision of 92.35%, recall of 69.37%, and F-score of 77.59%). Similar to the probe class, GRU using two layers' model has achieved the highest performances among other models and other classes (accuracy of 99.24%, precision of 99.58%, recall of 13.17%, and F-score of 35.28%). In comparison, LSTM with one and two layers and GRU with one layer have registered similar performance. Regarding R2l class, GRU using two layers has achieved the highest performances among other models (accuracy of 99.8%, precision of 80.4%, recall of 26.66%, and F-score of 39.92%). U2R class is like the other classes; all deep learning models have recorded the same performance (accuracy of 99.99%, precision of 0%, recall of 0%, and F-score 0%). Based on these results, all models for U2R are the worst performance concerning other models.

(2) Testing Results. Table 17 shows the performance of the deep learning models using the unseen testing KDD dataset. As shown in the results, for the DOS class, GRU using two layers' model has achieved the highest performances among other models and other classes (accuracy of 96.45%, precision of 96.20%, recall of 99.54%, and F-score of 97.84%). In contrast, LSTM with one layer has registered the lowest performance (accuracy of 94.98%, precision of 95.06%, recall of 98.94%, and F-score of 96.96%). For the normal class, GRU using two layers' model has achieved the highest performances among other models (accuracy of 96.93%, precision of 96.59%, recall of 85.89%, and F-score of 90.92%). LSTM with one layer has recorded the worst performances (accuracy of 95.27%, precision of 92.09%, recall of 80.43%, and F-score of 85.87%). Like the probe class, GRU using two layers' model has achieved the highest performances among other models and other classes (accuracy of 99.24%, precision of 91.67%, recall of 23.91%, and F-score of 38.05%). Regarding R2L class, GRU using two layers' model has achieved the highest performances among other models and other classes (accuracy of 99.79%, precision of 83.33%, recall of 23.46%, and F-score of 36.61%). In comparison, LSTM with two layers has the worst performances (accuracy of 98.76%, precision of 99.44%, recall of 99.20%, and F-score of 99.32%). For U2R class, all models have achieved the worst performance. Based on these results, GRU using two layers' model have registered the best performance.

5. Discussion

We examine the four machine learning models (DT, KNN, RF, and NB) and four deep learning models (LSTM and GRU, using one and two layers), including cross-validation results and testing using WSN-DS and KKD datasets.

Table 18 describes the summary of the used datasets including the number of all samples, the number of trained samples, the number of testing samples, the number of all features, the number of selected features, and the number of classified classes.

5.1. Regular Machine Learning. The results of machine learning cross-validation performance based on the accuracy for WSN dataset using all features and selected features are depicted in Figure 3. Considering all feature results for cross-validation performance, the normal class using RF has the best performance (accuracy of 99.95%), while the NB model using the scheduling class has achieved the worst performance among all models and classes (accuracy of 84.47%) (see Figure 3(a)). Similar to the selected feature results, the normal class using RF has the best performance (accuracy of 99.93%), while the scheduling class, using the NM model, has achieved the worst performance among all models and classes (accuracy of 84.61%) (see Figure 3(b)). The results of machine learning testing performance based on the accuracy for WSN dataset using all features and selected features are depicted in Figure 4. Considering all features' results for unseen dataset, the grayhole class using RF has the best performance (accuracy of 99.95%), while the scheduling class using the NB model has achieved the worst performance among all models and classes (accuracy of 84.1%) (see Figure 4(a)). Similar to the selected features' results, the grayhole class using RF has the best performance (accuracy of 99.94%), while the scheduling class using the NB model has achieved the worst performance among all models and classes (accuracy of 84.65%) (see Figure 4(b)).

The results of machine learning cross-validation performance based on the accuracy for KDD dataset using all features and selected features are depicted in Figure 5. Considering all features results for cross-validation performance, DOS and U2R classes using RF have the best performance (accuracy of 100%), while the NB model using the DOS class has achieved the worst performance among all models and classes (accuracy of 92.44%) (see Figure 5(a)). Similar to the selected features' results, U2R class using RF has the best performance (accuracy of 100%), while the DOS class using the NB model has achieved the worst performance among all models and classes (accuracy of 65.13%) (see Figure 3(b)). The results of machine learning testing performance based on the accuracy for the WSN dataset using all features and selected features are depicted in Figure 6. Considering all features' results for the unseen dataset, DOS, normal, and U2R classes using RF have the best performance (accuracy of 100%), while the normal class using the NB model has achieved the worst performance among all models and classes (accuracy of 92.65%) (see Figure 6(a)). Similar to the selected features' results, R2L and U2R classes, using RF, DT, and KNN, have the best performance (accuracy of 99.99%), while the DOS class, using the NB model, has achieved the worst performance among all models and classes (accuracy of 64.87%) (see Figure 6(b)).

5.2. Deep Learning. The results of deep learning cross-validation performance based on the accuracy for WSN dataset using all features and selected features are depicted in Figure 7. Considering all features results for cross-validation performance, the flooding class, using LSTM with one layer, has the best performance (accuracy of 100%), while the blackhole class, using GRU with the one layer model, has achieved the worst performance among all models and classes (accuracy of 97.5%) (see Figure 7(a)). Similar to the selected features' results, the grayhole class, using LSTM with one layer, has the best performance (accuracy of 99.84%), while the blackhole class, using GRU with the two layers' model, has achieved the worst performance among all models and classes (accuracy of 97.21%) (see Figure 7(b)). The results of deep learning testing performance based on the accuracy for WSN dataset using all features and selected features are depicted in Figure 8. Considering all features' results for unseen dataset, the grayhole class, using GRU with the two layers' model, has the best performance (accuracy of 99.85%), while the blackhole class, using GRU with the two layers' model, has achieved the worst performance among all models and classes (accuracy of 98.01%) (see Figure 8(a)). Similar to the selected features results, the grayhole class, using LSTM with the one layer model, has the best performance (accuracy of 99.88%), while the blackhole class, using LSTM with the two layers' model, has achieved the worst performance among all models and classes (accuracy of 97.62%) (see Figure 8(b)).

The results of deep learning cross-validation performance based on the accuracy for KDD dataset using all features and selected features are depicted in Figure 9. Considering all features results for cross-validation performance, U2R class using LSTM with two layers and GRU using one and two layers have the best performance (accuracy of 99.99%), while the DOS class using LSTM with one layer model has achieved the worst performance among all models and classes (accuracy of 98.46%) (see Figure 9(a)). Similar to the selected features results, U2R class using LSTM with one and two layers and GRU using one and two layers have the best performance (accuracy of 99.99%), while the normal class using LSTM with the one layer model has achieved the worst performance among all models and classes (accuracy of 93.51%) (see Figure 9(b)). The results of deep learning testing performance based on the accuracy for KDD dataset using all features and selected features is depicted in Figure 10. Considering all features' results for unseen dataset, U2R and Dos classes using LSTM with one and two layers and GRU using one and two layers have the best performance (accuracy of 99.99%), while the normal class using GRU with the two layers' model has achieved the worst performance among all models and classes (accuracy of 99.86%) (see Figure 10(a)). Similar to the selected features' results, the U2R class using LSTM with one and two layers and GRU using one and two layers have the best performance (accuracy of 99.99%), while the DOS class using GRU with the one layer model has achieved the worst performance among all models and classes (accuracy of 95.72%) (see Figure 10(b)).

Table 18: The summary of the used datasets.

Dataset	WSN-DS	KDD
Number of all samples	374662	311029
Number of trained samples	280995	307916
Number of testing samples	93666	131964
Number of all features	19	42
Number of selected features	6	14
Number of classified classes	5	5

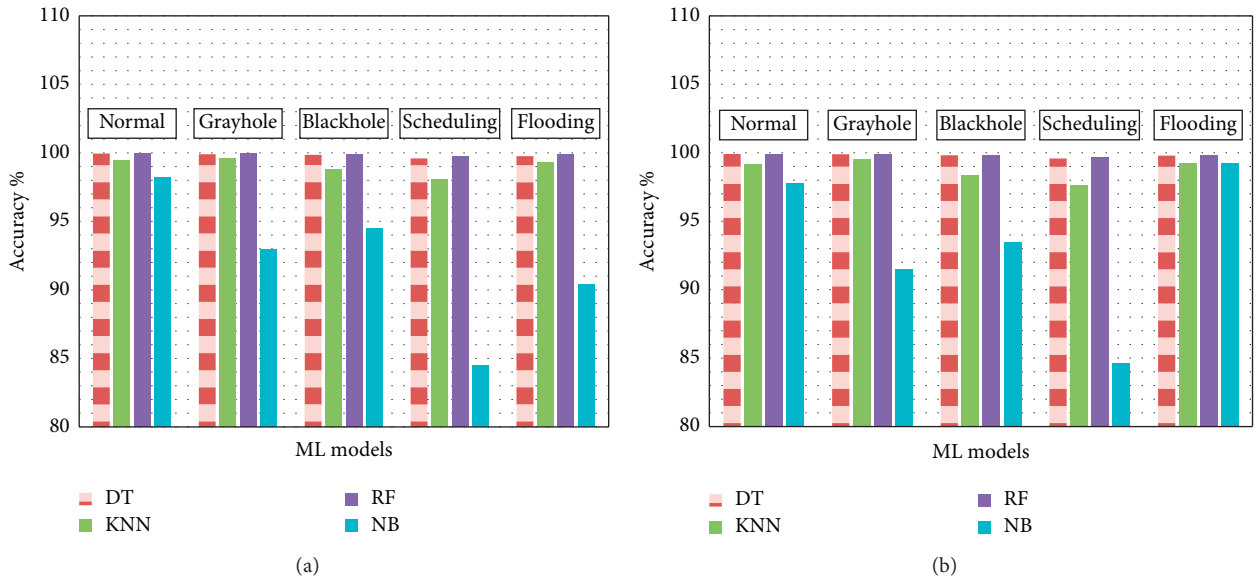

Figure 3: The results of machine learning cross-validation performance for the WSN dataset: (a) accuracy using all features and (b) accuracy using selected features.

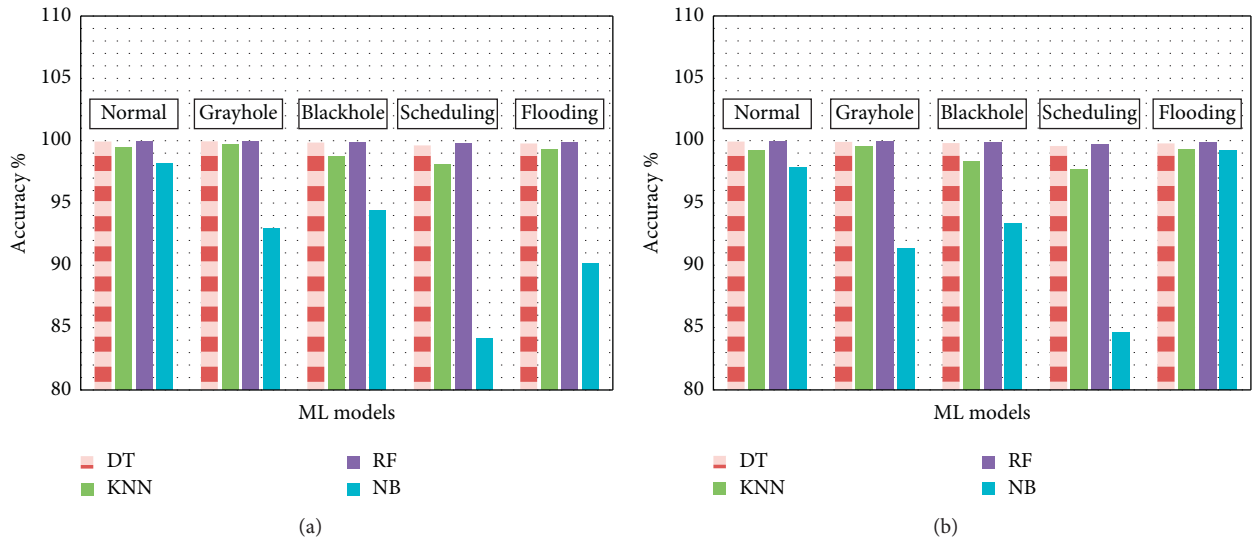

Figure 4: The results of machine learning testing performance for WSN dataset: (a) accuracy using all features and (b) accuracy using selected features.

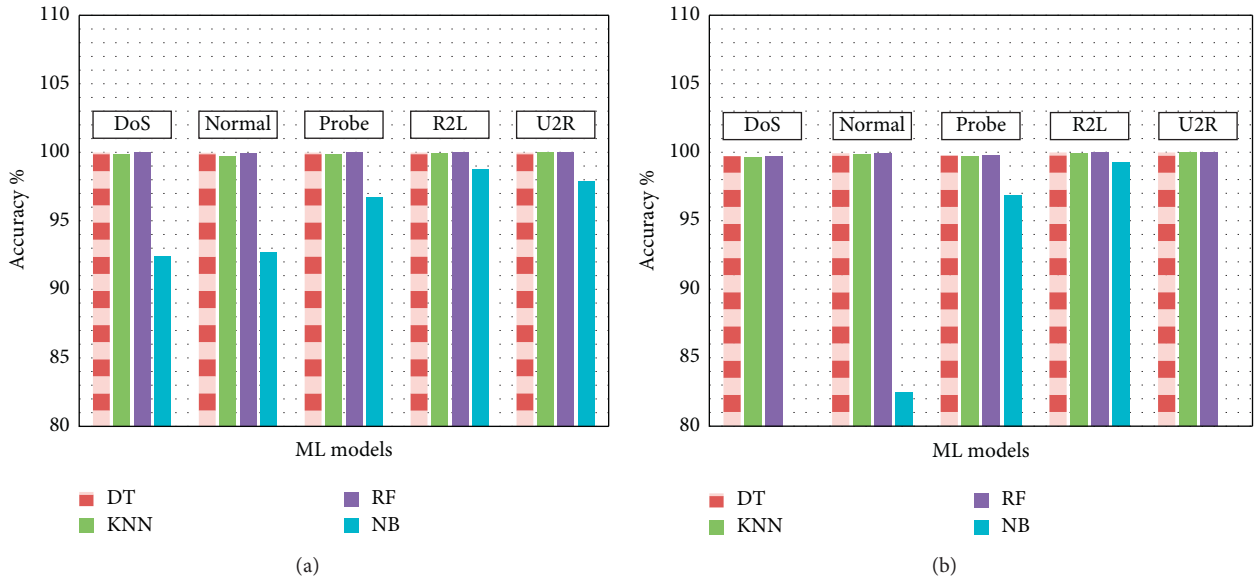

Figure 5: The results of machine learning cross-validation performance for KDD dataset: (a) accuracy using all features and (b) accuracy using selected features.

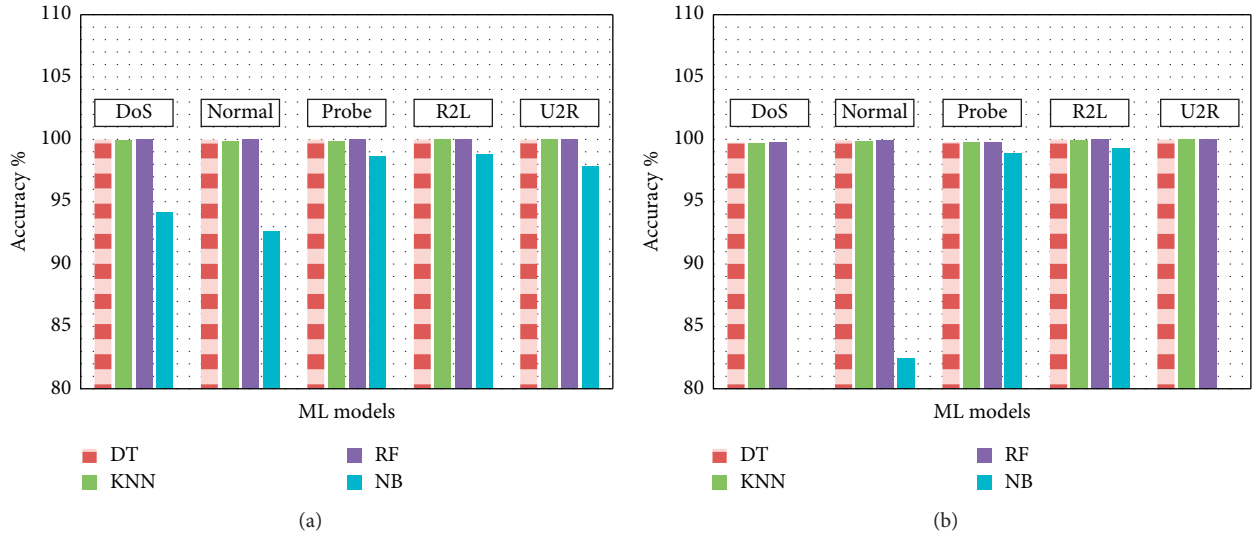

Figure 6: The results of machine learning testing performance for KDD dataset: (a) accuracy using all features and (b) accuracy using selected features.

5.3. Summary. Many research studies in AI-based IDS area have used machine learning and deep learning models. Each of these models possesses its strengths and weaknesses, making them suitable for a particular attack type. Regarding this work, not only do we use machine learning and deep learning models, but an in-depth investigation of performance analysis has been carried out based on the chosen datasets. The performance analysis and comparison of these models on IDS datasets show no superiority of one model among the chosen datasets using all features and selected features. Furthermore, these findings lead to better knowledge and understand the interpretability for choosing the right model to enhance IDS trust. In particular, the authors in [24] have addressed the explainable artificial intelligence (XAI) concept to improve the trust management by exploring the decision tree model in the area of IDS. Compared to our work, we have provided a performance-based comparison using machine learning and deep learning models to investigate the trust impact based on the accuracy of the trusted AI-based systems regarding IDs' malicious data.

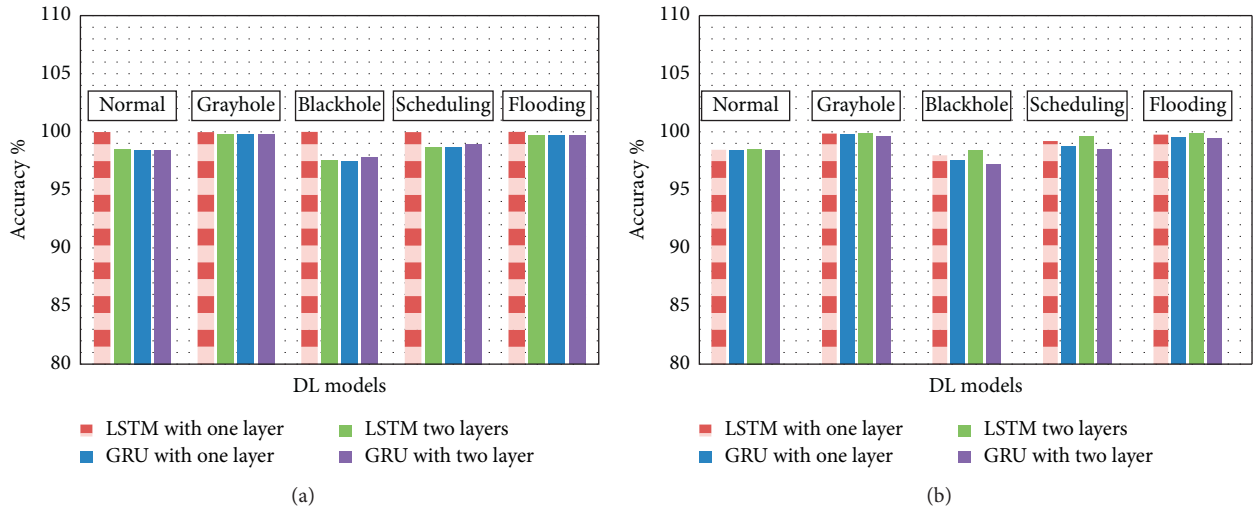

Figure 7: The results of deep learning cross-validation performance for WSN dataset: (a) accuracy using all features and (b) accuracy using selected features.

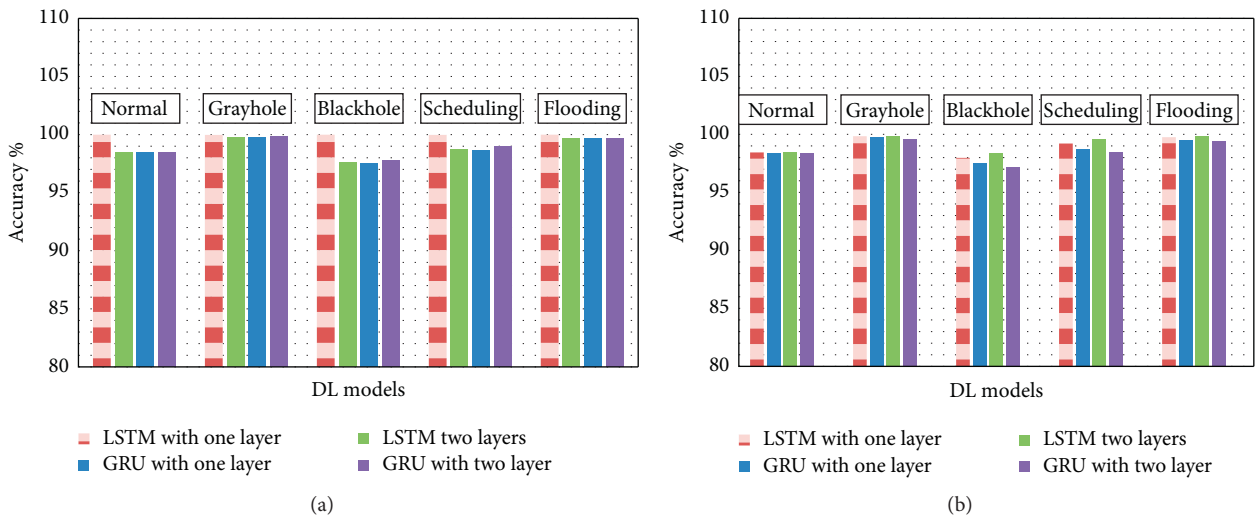

Figure 8: The results of deep learning testing performance for WSN dataset: (a) accuracy using all features and (b) accuracy using selected features.

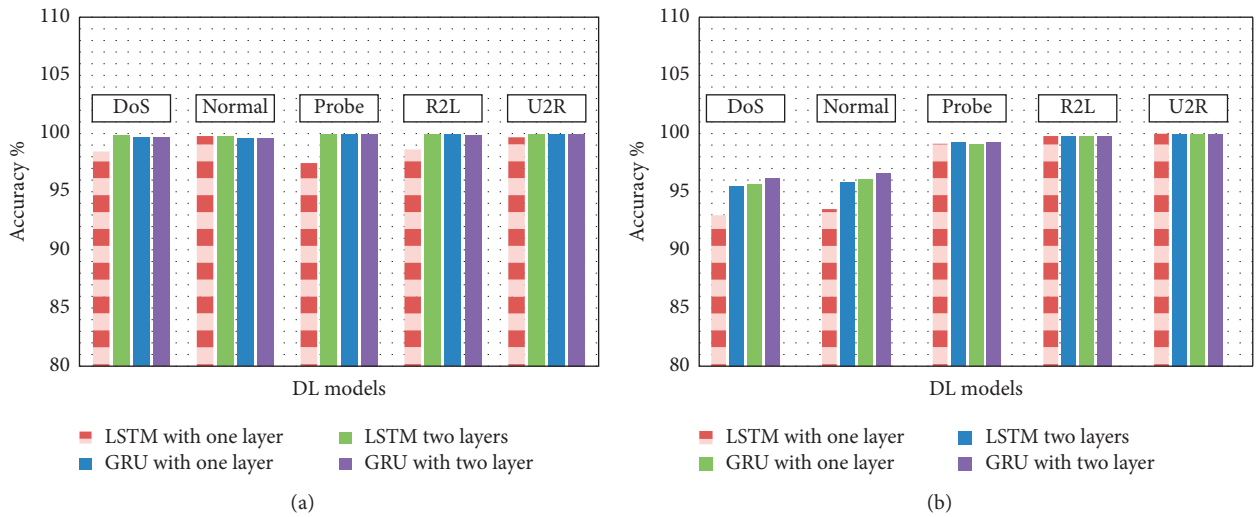

Figure 9: The results of deep learning cross-validation performance for the KDD dataset: (a) accuracy using all features and (b) accuracy using selected features.

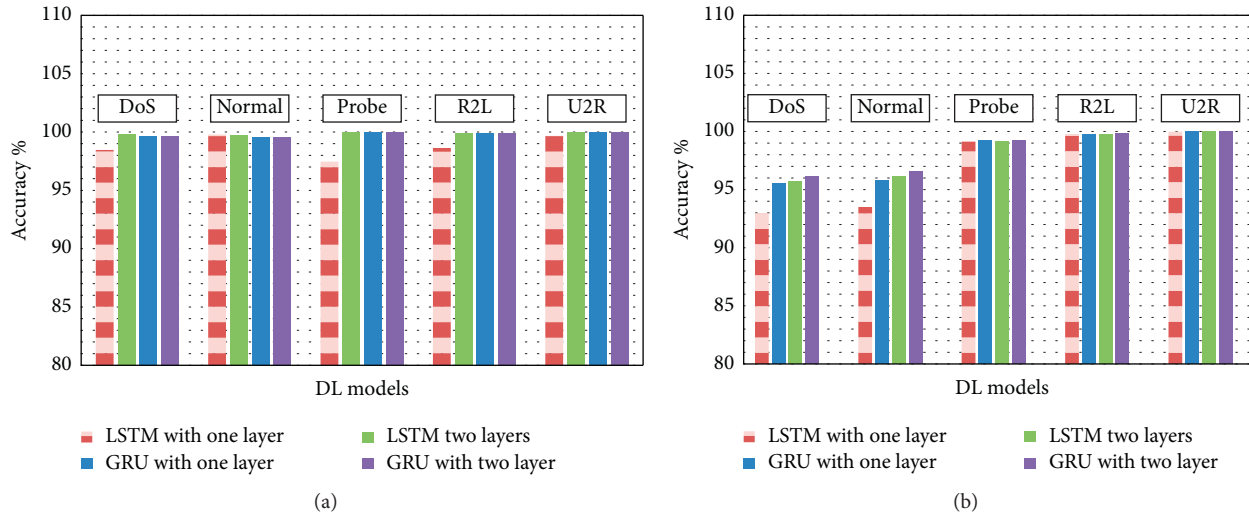

Figure 10: The results of deep learning testing performance for the KDD dataset: (a) accuracy using all features and (b) accuracy using selected features.

6. Conclusion and Future Work

In this paper, a comparison study is introduced to investigate intrusion detection's trust impact, including the data, methodology, and expert accountability by analyzing machine learning and deep learning models' performance. The developed phases of the comparison study have two-folds. The first comparison is made using four regular machine learning models, including DT, KNN, RF, and NB. The second comparison is made using four traditional deep learning models, including LSTM (one and two layers) and GRU (one and two layers). Two datasets are used to classify the attack type in IDS, including WSN-DS and KDD. The experimental results are significantly demonstrated, considering the data, methodology, and expert accountability causes misleading predictions, making the system vulnerable to attacks, and leading to zero-trust security for critical systems. Therefore, for future work, we plan to use XAI concept to enhance trust management by exploring machine learning models and deep learning in IDS.

Data Availability

The KDD dataset used to support the study is available at http://kdd.ics.uci.edu/databases/kddcup99/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by Taif University researchers, supporting Project no. TURSP-2020/254, Taif University, Taif, Saudi Arabia and Science Foundation Ireland SFI.

References

[1] D. Pienta, S. Tams, and J. Thatcher, "Can trust be trusted in cybersecurity?" in *Proceedings of the 53rd Hawaii International Conference on System Sciences*, Maui, HI, USA, January 2020.

[2] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on sdn based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493–501, 2019.

[3] P. Svenmarck, L. Luotsinen, M. Nilsson, and J. Schubert, "Possibilities and challenges for artificial intelligence in military applications," in *Proceedings of the NATO Big Data and Artificial Intelligence for Military Decision Making Specialists' Meeting*, Bordeaux, France, May 2018.

[4] M. Stampar and K. Fertalj, "Artificial intelligence in network intrusion detection," in *Proceedings of the 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1318–1323, IEEE, Opatija, Croatia, May 2015.

[5] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in *International Workshop on Recent Advances in Intrusion Detection*, pp. 161–180, Springer, Berlin, Germany, 2011.

[6] 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.

[7] I. Almomani, B. Al-Kasasbeh, and M. Al-Akhras, "Wsn-ds: a dataset for intrusion detection systems in wireless sensor networks," *Journal of Sensors*, vol. 2016, Article ID 4731953, 16 pages, 2016.

[8] M. Hachimi, G. Kaddoum, G. Gagnon, and P. Illy, "Multi-stage jamming attacks detection using deep learning combined with kernelized support vector machine in 5g cloud radio access networks," in *Proceedings of the 2020 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–5, IEEE, Montreal, Canada, June 2020.

[9] A. B. Abhale and S. Manivannan, "Supervised machine learning classification algorithmic approach for finding anomaly type of intrusion detection in wireless sensor network," *Optical Memory and Neural Networks*, vol. 29, no. 3, pp. 244–256, 2020.

[10] M. Alqahtani, A. Gumaei, H. Mathkour, and M. Maher Ben Ismail, "A genetic-based extreme gradient boosting model for

detecting intrusions in wireless sensor networks," *Sensors*, vol. 19, no. 20, p. 4383, 2019.

[11] M. E. Haque and T. M. Alkharobi, "Adaptive hybrid model for network intrusion detection and comparison among machine learning algorithms," *International Journal of Machine Learning and Computing*, vol. 5, no. 1, p. 17, 2015.

[12] S. V. Farrahi and M. Ahmadzadeh, "Kcmc: a hybrid learning approach for network intrusion detection using k-means clustering and multiple classifiers," *International Journal of Computer Applications*, vol. 124, no. 9, 2015.

[13] S. Paliwal and R. Gupta, "Denial-of-service, probing & remote to user (r2l) attack detection using genetic algorithm," *International Journal of Computer Applications*, vol. 60, no. 19, pp. 57–62, 2012.

[14] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class svm," *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 4, pp. 462–472, 2017.

[15] S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *Computers & Security*, vol. 24, no. 4, pp. 295–307, 2005.

[16] S. Zaman and F. Karray, "Lightweight ids based on features selection and ids classification scheme," in *Proceedings of the 2009 International Conference on Computational Science and Engineering*, pp. 365–370, IEEE, Vancouver, BC, Canada, August 2009.

[17] K. Vimalkumar and N. Radhika, "A big data framework for intrusion detection in smart grids using Apache spark," in *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 198–204, IEEE, Udupi, India, September 2017.

[18] S. Balakrishnan, K. V. Venkatalakshmi, and A. K. Kannan, "Intrusion detection system using feature selection and classification technique," *International Journal of Computer Science and Application*, vol. 3, no. 4, pp. 145–151, 2014.

[19] M. Alkasassbeh, G. Al-Naymat, A. Hassanat, and M. Almseidin, "Detecting distributed denial of service attacks using data mining techniques," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 1, pp. 436–445, 2016.

[20] K. Peng, V. Leung, L. Zheng, S. Wang, C. Huang, and T. Lin, "Intrusion detection system based on decision tree over big data in fog environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 4680867, 10 pages, 2018.

[21] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, p. 20, 2019.

[22] C. Cortes, M. Mohri, and A. Rostamizadeh, *L2 Regularization for Learning Kernels*, 2012, http://arxiv.org/abs/1205.2653.

[23] P. Baldi and P. J. Sadowski, "Understanding dropout," *Advances in Neural Information Processing Systems*, vol. 26, pp. 2814–2822, 2013.

[24] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, "Explainable artificial intelligence (xai) to enhance trust management in intrusion detection systems using decision tree model," *Complexity*, vol. 2021, Article ID 6634811, 11 pages, 2021.