Detection of IoT Cyberattacks in Smart Cities: A Comparative Analysis of Deep Learning and Ensemble Learning Methods

Zeru Kifle Kebede[1 [0009−0001−4272−2250]] and Petr Hajek[1 [0000−0001−5579−1215]]

Faculty of Economics and Administration, University of Pardubice, Studentska 84, 532 10 Pardubice, Czech Republic
zerukifle.kebede@student.upce.cz, petr.hajek@upce.cz

Abstract. Internet of Things (IoT) application integrates various services and products in smart cities. It is used to share and transfer real-time data and information autonomously via the Internet. However, the smart city environment is vulnerable to various new and intelligent cyberattacks that exploit the vulnerabilities of IoT devices integrating into the system. As a result, security becomes one of the most crucial concerns that needs to be addressed. In this paper, we conducted a comparative analysis of deep learning (DL) and ensemble learning-based methods. The MLP, LSTM, GRU, RF, and AdaBoost methods were analysed for detecting binary and individual attack classes. We have utilized an imbalanced and big datasets (UNSW-NB15 and CICIDS2017) for this task. The experimental results indicate that RF outperforms other models. Among DL models, MLP achieves the highest recall, precision, F1-score, and has the lowest FPR. Specifically, on the UNSW-NB15 dataset, MLP achieves values of 99.17% for recall, precision, and F1-score, and 0.0037 for FPR. On the CICIDS2017 dataset, MLP achieves values of 98.11% to 98.12% for recall, precision, and F1-score, and 0.0106 for FPR. These results prove that the DNN model performs well for high-dimensional data compared to ensemble models. Generally, for binary classification, the proposed model performs well, while the experimental result for individual attack classification provides insights into some types of attacks that are difficult to detect and need more attention for deploying an intrusion detection system in a real-world environment.

Keywords: IoT· Cyberattack· Smart City · Intrusion Detection · Deep Learning · Ensemble Learning.

1 Introduction

IoT is a technology in which physical devices share real-time data or information via the Internet without the intervention of human beings. These physical devices are outfitted with various sensors that are embedded in various systems to exchange sensitive data and interact with one another autonomously to perform the specified tasks [1]. However, these applications are highly vulnerable to cyberattacks. This vulnerability occurred due to the limited resources and the open

communication medium of IoT devices. Thus, it is hard to implement classical security methods on this devices [1]. Attackers use Denial of Service (DoS), Distributed DoS (DDoS), man-in-the-middle, identity theft, data theft, ransomware attacks, and so on, to manipulate IoT devices using their vulnerabilities [1,2,3].

Many intrusion detection systems (IDSs) have been developed by researchers using several Machine Learning (ML) and Deep Learning (DL) based methods, and various datasets to secure IoT in smart cities. The authors in [4] used Random Forest (RF), Support Vector Machine (SVM), and Neural Network (NN) models on a reduced sample size of UNSW-NB15 dataset. Among these models, RF scored the highest accuracy of 98.67% and 97.37% for binary and multi-class attack classification respectively. The authors in [5] also used the balanced CICIDS2017 dataset to evaluate their hybrid DL-based IDS model designed for multi-class attack classification. In [6], the authors used both UNSW-NB15 and CICIDS2017 with other datasets to evaluate their DL method-based benchmark IDS for IoT. Authors in [3], proposed an intelligent IDS using DL algorithms on both old and recent datasets such as KDDCup 99, NSL-KDD, Kyoto, WSN-DS, UNSW-NB15, and CICDS 2017.

However, existing studies [3,4,5,6] have not exploited the full sample of the original datasets to train and evaluate their models. To achieve computational efficiency when dealing with large IoT datasets, developers of IoT IDSs have modified large and unbalanced datasets generated by IoT devices using various techniques to reduce its size and dimension and balance the classes for training and evaluating their models. This, in turn, can lead to the loss of important information and a biased decision of IDSs. Moreover, the detection of cyberattacks in IoT is influenced by other factors, such as types and dynamic behaviors of attacks, requiring methods learning temporal patterns in the data. Finally, it may be challenging to detect some specific types of cyberattacks due to the specific behavior of intruders [2,3], a problem neglected in earlier research.

To overcome these problems, this study presents a comparative analysis of DL and ensemble learning models used for intrusion detection and classification in the context of smart cities. To demonstrate the efficacy of the proposed IDS, we have used recent large and unbalanced datasets consisting of various attacks. The DL and ensemble learning methods used for modeling are Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), RF, and AdaBoost. These models were trained and evaluated with UNSW-NB15 [7,8] and CICIDS 2017 [9] datasets. The main contributions of this work are listed as follows:

- We explore the performance of DL and ensemble learning methods in detecting and classifying binary and individual class attacks using large unbalanced datasets.
- To provide a comparative analysis of DNN and ensemble-based models for detecting and classifying IoT cyberattacks in a smart city environment.
- To examine the effect of using full sample of unbalanced datasets (UNSW-NB15 and CICIDS 2017) to evaluate DL and ensemble-based IDS and com-

pare them to earlier research work that utilized different sample size of these datasets.

The rest of this paper is organized as follows. Section 2 covers the review of related work. Section 3 provides the architectures, algorithms and datasets used for the proposed models. Section 4 presents the experimental results of the proposed model and comparison with existing approaches. Section 5 concludes this work.

2 Literature Review

This section presents a review of IoT security related works that applied ML and DL techniques.

To detect intrusion in IoT, the authors in [4] applied RF, SVM, and NN models using the UNSW-NB15 dataset. The dataset was grouped into four clusters: full features, Flow/MQTT features, TCP features, Top (flow/MQTT and TCP) features. They used undersampling techniques to balance the dataset, reducing the normal class by 50% while keeping the others. The RF model was evaluated on the balanced full feature cluster of the dataset. It achieved 98.67% and 97.37% accuracy for binary and multi-class attack classification respectively. Their models evaluated using a small number of dataset features was slightly degraded, which indicates the loss of some important features.

In [3], the authors explore deep NNs that are used to develop an adaptable and efficient IDS. The system was used to detect and categorize unplanned and unpredictable cyberattacks in the network. They used various freely available cyber community malware datasets. This study aims to identify suitable algorithms for detecting dynamic IoT cyberattacks for the future. They conducted experiments on the DL-based model for attack detection using NSL-KDD, UNSW-NB15, Kyoto, WSN-DS, CICIDS 2017, and KDDCup 99 datasets. The result indicates that the model outperforms the KDDCup 99 dataset. They also proposed a new scalable hybrid DL-IDS framework for detecting malware in the network.

In [6], the authors provide comprehensive DL-based IDS models evaluated by various datasets that are applicable as benchmarks for comparison and the development of suitable IDS for the IoT environment. They develop multiple individual and hybrid DL models and evaluate them using various datasets, which can reduce the biased results due to a single classifier model evaluation with a single dataset and vice versa. Their model focuses only on multi-classification and its performance on the UNSW-NB15 and CICIDS2017 datasets. The model was good for majority class attacks but not for minority classes, such as Heartbleed, Web Attack SQL Injection, and Infiltration in the CICIDS2017 dataset and Fuzzers, Backdoors, Shellcode, and Worms in the UNSW-NB15 dataset.

The authors in [5] also proposed a hybrid DL-based intrusion detection model using the balanced CICIDS 2017 dataset to classify multi-class attacks. A random undersampler for the majority class and K-SMOTE for oversampling the minority class were used to balance the dataset. The proposed model (TBGD)

outperforms other ensemble and DL models without data balancing. However, the accuracy of TBGD is very low for some attack categories, detecting 3%, 6% and 12% of the Web_Attack_Sql_Injection, Web_Attack_XSS and Infiltration attacks, respectively.

In [10], the authors proposed IoT cyberattack detection and mitigation models using ML algorithms for IoT-based smart city applications. They used Decision Tree (DT), RF, Logistic Regression (LR), SVM, K-Nearest Neighbour (KNN), and NN ML algorithms to build their models. Apart from this, the authors employed classical and ensemble algorithms with advanced feature selection techniques for binary and multi-class attack classification. The ensemble-based detection model scored better than a single classifier approach with the performance metrics of accuracy, precision, recall, and F1-Score using the UNSW-NB15 and CICIDS2017 datasets. However, for further improvement, using DL was recommended for further enhancement.

In [11], the authors proposed NN-based cyberattack mitigation techniques for smart city applications. They used a small sample size of the UNSW-NB15 dataset for evaluating their model. Its performance was evaluated using the most commonly used performance metrics, such as accuracy, precision, recall, and F1-score, and its scores were 85.1%, 84%, 85%, and 84%, respectively. However, to improve the performance of the detection model, they again recommended a DL model for future work.

The related works show that IoT devices in smart cities need improvement in security aspects to ensure seamless operation. This research aims to develop an intelligent attack classification model using DL and ensemble learning-based methods to detect IoT cyberattacks on large and unbalanced datasets relevant to smart cities. Moreover, a comparative analysis was conducted to evaluate the efficiency of DL and ensemble learning-based IDS for binary and individual class attack classification with the UNSW-NB15 and CICIDS2017 datasets, as well as the implications of the results and comparison to earlier work are presented.

3 Proposed Architecture for IoT CyberAttack Detection Approach

In this section, the overall architecture of the proposed IDS design with the DL and ensemble learning methods, relevant datasets for model training and evaluation are described. Fig. 1, shows the flow of the proposed system used to model an IoT cyberattack detection system for smart cities.

3.1 Deep Learning and Ensemble Learning Methods

Because the DL-based IDS exhibits an impressive performance in many security applications, many researchers now favor it for modeling IDS [2,3]. The DL approach is appropriate for analyzing high-dimensional features on large datasets [3]. In addition, DL models such as LSTM and GRU are more effective at processing sequential network data for attack detection [6]. In this section, the DL algorithms and ensemble method used for this work are explored as follows.

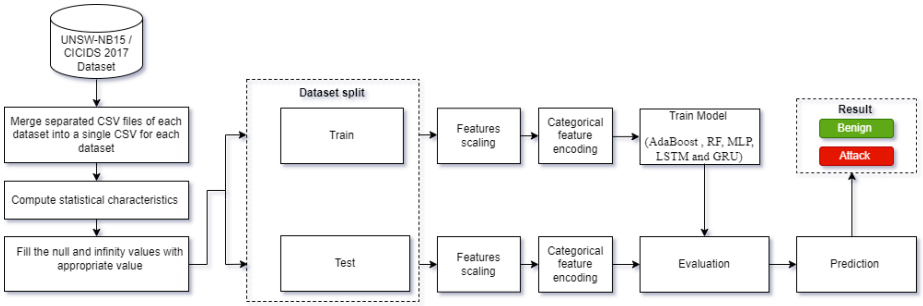

Fig. 1. Proposed architecture for detecting IoT cyberattack in smart cities.

Multi-Layer Perceptron An MLP NN is a family of feed-forward networks, which is built from the input layer, hidden layer, and output layer. The information passes from the input layer to the output layer in a forward direction. The sum of the weighted input signals from the previous layer and a bias are fed to the activation function at each hidden layer node. Then the output of the activation function is sent forward as input to the subsequent hidden layer node or output layer node. The backpropagation is used to update the input signal weight until the model achieve the desired result [6,12,13].

Long Short-Term Memory The LSTM neural network is a family of NNs that is an enhanced version of recurrent NNs. LSTM was originally developed to handle gradient and vanishing gradient problems in recurrent NNs [12,13]. The LSTM architecture is composed of three gates and a memory cell. These gates are an input gate, output gate, and forget gate, which controls the information flow towards or out of a memory cell. The memory cell used to capture and maintain long-term dependencies in sequential data [6,12,13].

Gated Recurrent Units GRU is a Lightweight version of LSTM. Unlike the LSTM, it is made up of two gates. These are the update gate and reset gate. The update gate substitutes the function of the input gate and forget gate used in LSTM, which helps the model's decision-making on what data should be sent to the following state. The activation function at the reset gate stage of GRU takes the combination of the input vector value at time t and the previous hidden state value and then computes the activation function output. This output determines whether the previously hidden state value is partially passed, totally passed, or discarded to pass the update state [13].

Random Forest Classifier An RF classifier is an ensemble machine-learning model that combines different decision trees and is used to solve classification problems such as fraud detection, image classification, and natural language processing. It built from multiple decision trees using random subsets of both

the features of the data set and the data set itself. The algorithm repeats this process to build the forest. The RF classifier combines the predictions from each decision tree by either taking a majority vote or averaging them to produce the final model output. RF classifier is less prone to overfitting problems compared to a single decision tree classifier [14].

Adaptive Boosting (AdaBoost) AdaBoost is an ensemble machine learning method that combines multiple weak classification algorithm models to create a single strong classifier model to improve the classification performance of the model [14]. The initial training data instances used to train the first model have the same weight. To address the misclassified instances in the first model, increase the weight of these instances to give them more weight, and conversely, decrease the weight of the correctly classified instances. Then the second model is trained. The model continues to train by weighting misclassified instances in the dataset until the predetermined criteria are met or the prediction error reaches its expected minimum. The final output of the model is the aggregated value of all the models built during each iteration [14].

3.2 Data Selection and Preprocessing

Dataset Selection Numerous datasets are available for IDS training and evaluation such as UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, BoTNeTIoT-L01, KDD98, DARPA98, KDDCUP99, NSLKDD [15]. However, not all datasets include recent varieties of attacks [7,13]. In this work, we used two recent and publicly available UNSW-NB15 and CICIDS2017 datasets. These datasets comprise various attacks, illustrated in Table 1, as compared to other datasets.

The UNSW-NB15 dataset was developed at the Australian Center for Cyber Security by using the IXIA traffic generator [15]. This dataset contains 2,540,044 records and 49 features, including labels, across four separate CSV files.

The CICIDS2017 dataset was developed by The Canadian Institute of Cyber security [9] contains eight distinct CSV file names with normal and attack variants. In total, 2,830,743 records with 79 attributes, including target features, are included.

Data Preprocessing The first step of data preprocessing was combining the separated CSVs of each dataset into a single CSV file for each dataset. Then, the statistical characteristics of the raw features were explored, such as data types, null values, noise, and so on. These characteristics were used to further analyze the datasets.

Cleaning dataset: Data cleaning began by analysing the records of the datasets. The UNSW-NB15 dataset features such as "ct_flw_http_mthd", "is_ftp_login," and "attack_cat" contain 1348145, 1429879, and 2218764 records of null value, respectively. According to the report presented in [7,15], the null values of the first two attribute have been replaced by 0 and the "attack_cat" has been

replaced by "normal". The features "is_sm_ip_ports", "is_ftp_login", and "ct_ftp_cmd" contained incorrect data types, and we have transformed them to appropriate data types [7,15].

The CICIDS2017 dataset contains 1,358 null instances. The number of instances is very small compared to the whole dataset, so we dropped them. After dropping the null instances, there were 2,827,876 instances left in the dataset. Attacks with similar behaviors and a low frequency have been grouped together and labeled accordingly [16]. See the attacks category after grouping in Table 1. Both datasets used in this work were split into 70% train and 30% test in Table 1.

Feature Scaling: In this research, normalization feature scaling technique [17] was used to transform numeric data values in the dataset to the required scales and ranges of data formats.

Feature Selection and Feature Encoding: All attributes in the dataset do not have the same significance level to influence the model output. Most attributes have higher significance; some do not [17]. The Pearson correlation method was used to reduce less important and highly correlated input features (with $p < 0.05$ significance) from our dataset (13 and 22 features discarded for the UNSW-NB15 and CICIDS2017 dataset, respectively). For categorical data encoding, we used a label encoder for the CICIDS2017 dataset to decode the ('Label') attribute value into 0 for the normal class and 1 for the attack class, and we applied one hot encoder for the USNW-NB15 dataset to convert the (proto, state service) feature value into numerical data.

Table 1. Train and Test subset of the UNSW-NB15 and CICIDS 2017 datasets

UNSW-NB15 Dataset			**CICIDS2017 Dataset**		
Attack category	**Train set**	**Test set**	**Attack category**	**Train set**	**Test set**
Normal	1553058	665706	BENIGN	1590306	681014
Generic	151011	64470	DoS	175867	75845
Exploits	31182	13343	PortScan	110968	47836
Fuzzers	16876	7370	DDoS	89718	38307
DoS	11419	4934	Brute Force	9685	4147
Reconnaissance	9779	4208	Web Attack	1546	634
Analysis	1857	820	Bot	1392	564
Backdoor	1671	658	Infiltration	26	10
Shellcode	1054	457	Heartbleed	5	6
Worms	125	49			

4 Experimental Setup and Results

4.1 Experimental Setup and Model Hyperparameter Configuration

The proposed DL and ensemble learning-based model was simulated with the hardware of the Lenovo IdeaPad 5 14ITL05 laptop, which consists of an 11th Gen Intel (R) CoreTM i5-1135G7 CPU (2.40 GHz), 16 GB of RAM, and runs on 64-bit Microsoft Windows 11 Home, as well as the software's Python 3.11 and various Python libraries (Keras, Tensorflow, Scikit-Learn, and so on).

The hyperparameter configuration of DL models was the result of the grid search procedure. For the LSTM and GRU classifiers, it was set as follows: hidden_layer_sizes = 2 (20,10) neurons, return_sequence = True, activation = sigmoid, optimizer = Adam, loss = binary_crossentropy, batch_size = 1024, epoch = 150, and verbose = 2. The MLP classifier hyperparameter was configured with a hidden_layer_size of 2 (20,10) neurons, activation = sigmoid, optimizer = Adam and batch_size = 1024. The remaining parameters for MLP, Adaboost, and RF classifiers were set to their default values as defined in the Scikit-Learn library for each classifier.

4.2 Result Evaluation Metrics

The performance of models was evaluated with different metrics. The evaluation metrics, accuracy, recall, precision, F1-score, and FPR, used in this research are mathematically defined in [3,17]. Moreover, in this research, True Positive (TP) is the number of non-attack instances classified as a non-attack class, True Negative (TN) is the number of attack instances classified as an attack class, False Positive (FP) is the number of attack instances classified as non-attack class, i.e., assigned in incorrect class, and False Negative (FN) is the number of non-attack instance classified as attack class.

4.3 Results

DL and Ensemble Learning-based IDS for Binary Attack Classification

Table 2 presents the performance evaluation results for binary class attack classification using DL and ensemble learning models on the UNSW-NB15 and CICIDS2017 datasets. On both datasets, RF outperforms the other models, while MLP performs best among other DL models in terms of overall performance metrics. On the other hand, the classification performance of the RF classifier slightly degrades on the UNSW-NB15 dataset as shown in Table 2 compared to the CICIDS2017 dataset. The reduced performance of the RF classifier might be due to the smaller ratio of attack class in the data and higher dimensionality stemming from categorical features in the UNSW-NB15 dataset. In contrast, the DL models show a slight improvement, particularly MLP using this dataset, suggesting that DL models outperform ensemble learning models when dealing with high-dimensional and unbalanced cyberattack data.

Table 2. Results of DL and ensemble learning models for UNSW-NB15 and CICIDS2017 datasets

	UNSW-NB15 Dataset					CICIDS2017 Dataset				
Model	**Accuracy (%)**	**Recall (%)**	**Precision (%)**	**F1-Score (%)**	**FPR (%)**	**Accuracy (%)**	**Recall (%)**	**Precision (%)**	**F1-score (%)**	**FPR (%)**
AdaBoost	99.09	96.49	96.22	96.35	0.55	98.86	98.11	96.17	97.13	0.96
RF	99.68	99.68	99.68	99.68	0.19	99.90	99.90	99.90	99.90	0.09
MLP	99.17	99.17	99.17	99.17	0.37	98.12	98.12	98.11	98.12	1.06
GRU	99.25	95.54	96.00	95.34	0.39	98.73	98.12	95.43	96.46	1.15
LSTM	99.26	95.54	96.08	95.39	0.38	98.56	97.52	95.18	96.00	1.20

DL and Ensemble Learning Model Performance on Individual Attack Categories Although the IoT attack detection model achieved a high detection accuracy rate for binary classification in Table 2, it may not perform well for all attack categories. As shown in Table 3 and in Table 4, the model's detection accuracy was very low for some attack types, despite its overall high detection performance. Table 3 shows that the Fuzzer attack category had a lower accuracy score across all models compared to the other attack categories. In addition, the Shellcode and Analysis attack types were also challenging to detect, following the Fuzzer attack. In the CICIDS2017 dataset, Bot, Infiltration and Web attack types were also the most difficult for the models to detect as shown in Table 4.

Table 3. DL and ensemble learning model performance on individual attack categories for UNSW-NB15 dataset

Models	**Normal (%)**	**Analysis (%)**	**Backdoor (%)**	**DoS (%)**	**Exploits (%)**	**Fuzzers (%)**	**Generic (%)**	**Reconna -issance (%)**	**Shellcode (%)**	**Worms (%)**
AdaBoost	99.45	90.24	93.47	97.08	96.57	66.08	99.89	99.38	86.43	97.96
RF	99.79	93.17	99.85	99.80	99.54	88.32	99.99	99.86	97.16	100
MLP	99.63	81.71	97.11	97.75	96.90	62.43	99.91	95.06	85.12	97.96
LSTM	99.61	82.44	98.02	98.46	97.17	67.80	99.91	98.74	90.81	95.92
GRU	99.62	82.32	97.42	98.64	97.72	66.23	99.93	99.12	91.68	97.96

Table 4. DL and ensemble learning model performance on individual attack categories for CICIDS2017 dataset

Model	**Benign (%)**	**Bot(%)**	**Brute Force(%)**	**DDoS (%)**	**DoS(%)**	**Heartbleed (%)**	**Infiltration (%)**	**PortScan (%)**	**Web-Attack(%)**
AdaBoost	99.04	1.42	97.32	99.80	98.26	83.33	0.00	98.98	5.21
RF	99.91	68.62	99.98	99.96	99.96	83.33	80.00	99.98	97.32
MLP	98.62	36.70	96.17	98.56	99.55	33.33	0.00	95.23	10.73
LSTM	98.73	34.93	99.45	98.97	99.84	0.00	10.00	95.95	34.23
GRU	98.84	37.06	89.53	98.51	99.83	83.33	0.00	96.50	9.15

Comparison with Earlier Research Result Table 5 and Table 6 show the binary attack classification performance of DL and ensemble learning models on different sample sizes of the UNSW-NB15 and CICIDS2017 datasets. The proposed models, including MLP, LSTM, GRU, Adaboost, and RF classifier, achieved better results on the complete records of both datasets. This indicates that the models can learn attack patterns effectively by utilizing the full dataset. This was achieved at the expense of more training and testing time needed. Specifically, for CICIDS2017, training and testing times were 378.7 s (7.0 s) for RF and 285.7 s (0.5 s) for MLP. For UNSW-NB15, it was 1,393.7 s (7.8 s) for RF and 92.1 s (1.7 s) for MLP. This is still acceptable for real-time applications, with more than 120 thous. test records processed per second using RF.

Table 5. Deep Neural Model Performance comparison on UNSW-NB15 dataset

Authors	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FPR (%)	Dataset Sample Size
[13]	AdaBoost	90.51	87.59	96.43	91.8	16.73	257,673
	MLP	87.26	82.02	98.44	89.48	26.43	
	LSTM	88.99	87.59	93.21	90.31	16.17	
	GRU	90.11	86.73	96.84	91.51	18.14	
[11]	NN	85.1	84	85	84	-	82,332
[4]	RF	98.67	-	-	-	-	Benign reduced by 50%(1,430,664)
	SVM	97.69	-	-	-	-	
	NN	94.78	-	-	-	-	
[12]	LSTM	85.08	-	-	90.36	-	257,666
	GRU	88.42	-	-	87.37	-	
	Simple RNN	87.07	-	-	90.03	-	
[3]	NN	78.4	94.4	72.5	82	-	121,981
	AdaBoost	90	98.5	86.6	92.2	-	
	RF	90.3	98.8	86.7	92.4	-	
[20]	MLP	98.33	-	99.97	85	1.75	Full (2,540,044)
	Conv. NN	98.22	-	99.85	84	1.86	
	RNN	98.12	-	99.73	84	1.97	
This study	AdaBoost	99.09	96.49	96.22	96.35	0.55	Full (2,540,044)
	RF	**99.68**	**99.68**	**99.68**	**99.68**	**0.19**	
	MLP	99.17	99.17	99.17	99.17	0.37	
	LSTM	99.26	96.08	95.54	95.39	0.38	
	GRU	99.25	96.00	95.54	95.34	0.39	

5 Conclusion

This paper proposes DL and ensemble-based models for detecting and classifying cyberattacks in IoT systems within smart cities. A comparative analysis of the DL and ensemble models for binary attack classification is performed using two recent and unbalanced big datasets, the CICIDS2017 and UNSW-NB15. The algorithms used as classifiers are MLP, LSTM, GRU, Adaboost, and RF. Furthermore, this study presents the detection accuracy of the models for individual attack categories. Further, it compares the binary classification performances of the models with previous research on these two datasets.

Table 6. Deep Neural Model Performance comparison on CICIDS2017 dataset

Authors	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FPR (%)	Dataset Sample Size
[9]	AdaBoost	-	77	84	77	-	Not specified
	RF	-	98	97	97	-	
	MLP	-	77	83	76	-	
[18]	LSTM	99.01	96.71	98.58	97.64	-	
[19]	Conv. NN-LSTM	93	86.47	76.83	81.36	-	Used only HTTP (29,309 records)
[3]	NN	96.3	90.8	97.3	93.9	-	121,981
	AdaBoost	94.1	88.7	91.8	90.2	-	
	RF	94	84.9	96.9	90.5	-	
This study	AdaBoost	98.86	98.11	96.17	97.13	0.96	Full (2,540,044)
	RF	**99.90**	**99.90**	**99.90**	**99.90**	**0.09**	
	MLP	98.12	98.11	98.12	98.16	1.06	
	LSTM	98.56	95.18	97.52	96.00	1.20	
	GRU	98.73	95.43	98.12	96.46	1.15	

In both datasets, RF outperforms the other models, while MLP performs better among other DNN models in terms of overall performance metrics. Individual attack classification experimental results show that the DL models are struggling to detect undersampled attacks such as a Bot, Heartbleed, and Infiltration, found in the CICIDS2017 dataset, and Fuzzer, Analysis, and Shellcode in the UNSW-NB15 dataset. The result suggests that more data is needed for these attack categories to further improve the performance of the IDSs. Alternatively, approaches using Generative Adversial Networks (GANs) are recommended to generate these undersampled attack categories. In our work, we have observed that the performance of DL models increases as the data dimension increases, while the performance of ensemble learning models decreases. Smart cities intensively use IoT devices that generate huge volumes of high-dimensional data. This result indicates that DL models could be the most suitable IDSs in smart cities with more data and features available. In the future, we propose evaluating the models with real-time smart city traffic datasets.

Acknowledgments. This paper was supported by the grant No. SGS 2024 017 provided by the Faculty of Economics and Administration, University of Pardubice.

Disclosure of Interests. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Singh, S., Fernandes, S.V., Padmanabha, V., Rubini, P.E.: Mcids-multi classifier intrusion detection system for iot cyber attack using deep learning algorithm. In: 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), pp. 354–360. IEEE, Tirunelveli, India (2021)
2. Chen, D., Wawrzynski,P., Lv, Z.: Cyber security in smart cities: a review of deep learning-based applications and case studies. Sustainable Cities and Society **66**, 102655 (2021)

3. Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A. Venkatraman, S.: Deep learning approach for intelligent intrusion detection system. IEEE Access **7**, 41525–41550 (2019)
4. Ahmad, M., Riaz, Q., Zeeshan, M., Tahir, H., Haider, S.A., Khan, M.S.: Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set. EURASIP Journal on Wireless Communications and Networking **2021**, 1–23 (2021)
5. Zhao, Y., Hu, Z., Liu, R.: TBGD: Deep learning methods on network intrusion setection using CICIDS2017 sataset. Journal of Physics: Conference Series **2670**(1), 012025 (2023)
6. Ahmad, R., Alsmadi, I., Alhamdani, W., Tawalbeh, L.: A comprehensive deep learning benchmark for IoT IDS. Computers & Security **114**, 102588 (2022)
7. Moustafa, N., Slay, J.: The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal: A Global Perspective **25**(1-3), 18–31(2016)
8. The UNSW-NB15 Dataset, https://research.unsw.edu.au/projects/unsw-nb15-dataset, last accessed 2023/12/29
9. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proc. of the 4th Int. Conf. on Information Systems Security and Privacy (ICISSp), pp. 108–116 (2018)
10. Rashid, M.M., Kamruzzaman, J., Hassan, M.M., Imam, T., Gordon, S.: Cyberattacks detection in iot-based smart city applications using machine learning techniques. International Journal of Environmental Research and Public Health **17**(24), 9347 (2020)
11. Rashid, Md.M., Kamruzzaman, J., Imam, T.,Kaisar, S., Alam, Md.J.: Cyber attacks detection from smart city applications using artificial neural network. In: 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), pp. 1–6. IEEE, Gold Coast, Australia (2020)
12. Kasongo, S.M.: A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. Computer Communications **199**, 113–125 (2023)
13. Disha, R.A., Waheed, S.: Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique. Cybersecurity **5**, 1 (2022)
14. Das, H., Naik, B., Behera, H.: An experimental analysis of machine learning classification algorithms on biomedical data. In: ICCDC 2019 Proc. of the 2nd Int. Conf. on Communication, Devices and Computing, pp. 525–539. Springer, Singapore (2020)
15. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), pp. 1–6. IEEE, Canberra, ACT, Australia (2015)
16. Panigrahi, R., Borah, S.: A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. International Journal of Engineering & Technology **7**(3.24), 479–482 (2018)
17. Bilal, M., Ali, G., Iqbal, M.W., Anwar, M., Malik, M.S.A., Kadir, R.A.: Auto-prep: efficient and automated data preprocessing pipeline. IEEE Access **10**, 107764–107784 (2022)
18. Figueiredo, J., Serrão, C., de Almeida, A.M.: Deep learning model transposition for network intrusion detection systems. Electronics **12**(2), 293 (2023)

19. Kim, A., Park, M., Lee, D.H.: AI-IDS: Application of deep learning to real-time Web intrusion detection. IEEE Access **8**,70245—70261 (2020)
20. Sarhan, M., Layeghy, S., Moustafa, N., Gallagher, M., Portmann, M.: Feature extraction for machine learning-based intrusion detection in IoT networks. Digital Communications and Networks **10**(1), 205–216 (2022)